Chapter 1

Introduction

1.1Introduction

Topographic sheets (TS) neatly organize various morphological features of a landscape as superimposed thematic layers. It represents the Earth's morphological features, both natural and man-made, on the earth's surface accurately on a two-dimensional plane using distinguishable color codes and standard symbolism. The different features include network and landmarks (roads, railways, landmarks, etc.), hydrography (lakes, rivers, streams, etc.), relief (mountains, depressions, valleys, etc.), and land use. The morphological features represented on the TS may be categorized as points, lines, or polygons, as shown in table 1.1 (page no 2). A point may be used to represent landmarks, etc. Likewise, roads, rivers, etc. are of type line, whereas boundaries, reservoirs, contours, etc. are of type polygon.

The method of extracting these morphological features along with the associated attributes from the map is known as digitization. The process of selecting a feature of interest from a layer is referred to as vectorization, as shown in figure 1.2 (page no 3). These extracted features, along with their associated attributes, are used for various types of morphological analyses.

The vectorization process may be manual, semi-automatic, or fully automatic. The traditional approach, which adopts a manual process for creating vectors for the reference map, calls for increased participation of the digitizer in the digitization process as every aspect of the feature must be manually selected and inducted into the respective vector. This expensive approach not only demands greater time but is also highly susceptible to human-induced errors; such an error-prone scientific basis may adversely hamper the research findings.

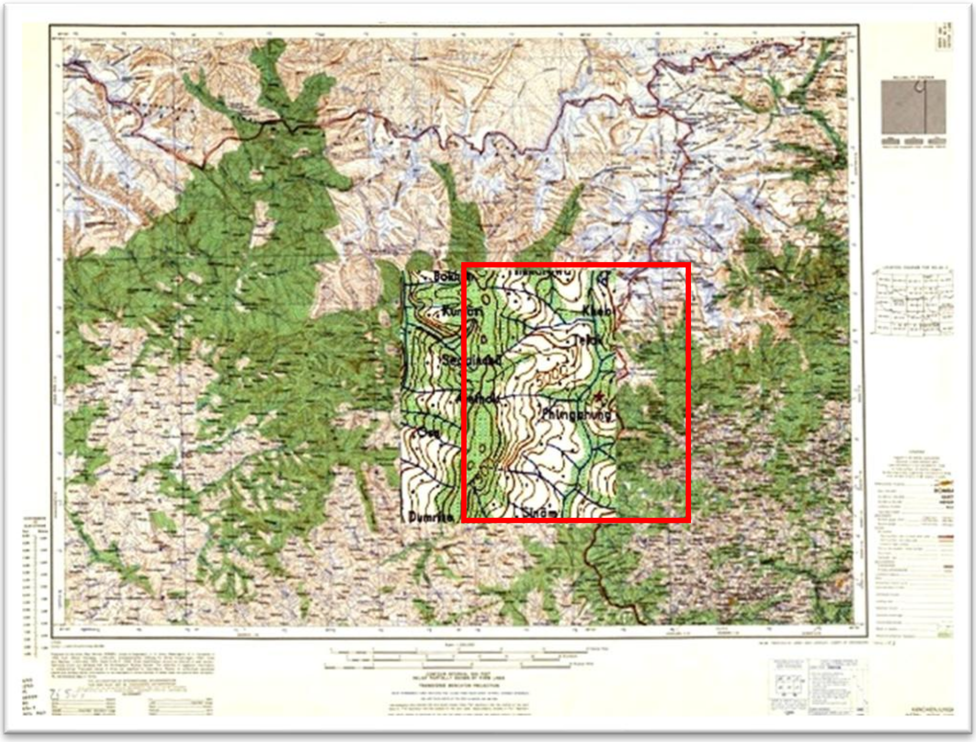
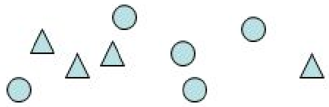
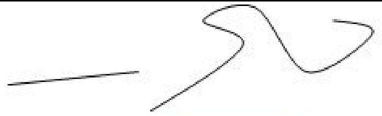
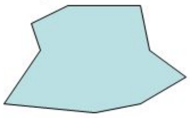


Figure 1.1: Sample Topographic Sheet [1]

Table 1.1: Examples of Point, Line, and Polygon Features in Topographic Sheet [2]

Feature	**Symbolism**	**Examples**
Point		• Landmark • Cities/ Towns • Point of Interest
Line		• Road • Railway

		• River • Contour
Polygon		• Watershed • Contour • Reservoir • Boundary

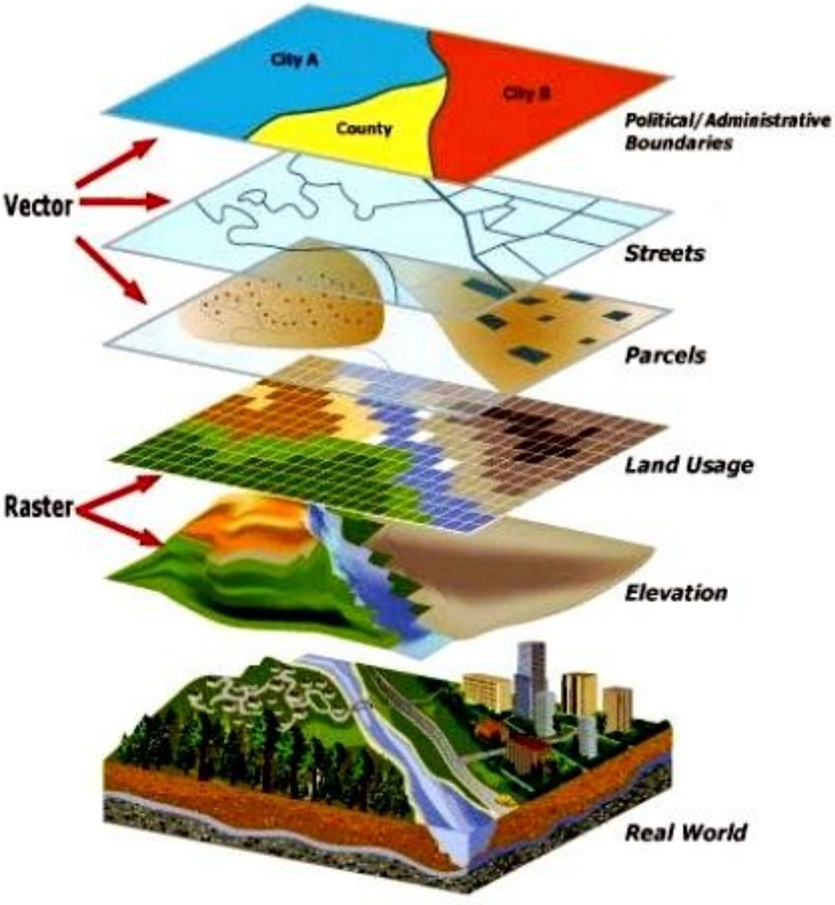


Figure 1.2: Sample Raster and Vector Layers in Geographic Information System (GIS) [3]

Even the semi-automatic computational GIS-based application calls for increased human participation, making it relatively ineffective, and the accuracy of digitization done through such platforms depends on the experience of the digitizer.

The inherent lag of both manual and semi-automatic digitization approaches consequently presents an opportunity to explore the avenue of conceptualizing and realizing a reliable, cost-effective, and self-learning computational system that generates a high-speed, high-quality digitized feature set and can be pursued as a suitable alternative for existing techniques. The creation of such a computational system would, to a great extent, elevate the ability and efficiency of the GIS-based application.

These automatic techniques can be further advanced in order to transform digitized dataset in 2D (Two dimensional) representation to 3D (Three dimensional) by incorporating additional features as the representation of a 2D TS in 3D space eases visualization, understanding, and analysis of any morphological features.

A TS is superimposed representation of various morphological features that can be broadly categorized into points, lines, polygons, and text as shown in figure 1.1 (page no 2). Landmarks is an example of point feature, contour lines, river network, and roadways are examples of line features whereas boundary, and lakes are examples of polygon features.

Contour line is a prominent feature in a TS which is a 2D representation of a 3D isoline. Further, contour lines are the imaginary lines that connect the points at an equal elevation from a reference level. Contour lines are crucial components of a TS, as these lines along with associated elevation values are used for constructing Digital Elevation Model (DEM). There are different 3D representations of the earth's surface: DEM, the Digital Surface Model (DSM), and the Digital Terrain Model (DTM).

A DEM represents an elevated surface as represented in figure 1.3 (page no 5). The construction of DEMs from a TS with the desired accuracy is still a challenge in the studies related to GIS. This calls for conception and realizing of an able computational system that aids the researcher in generating quality and reliable DEM which may be used as scientific basis for GIS-based research initiatives. DEM has applications ranging from urban planning such as transport networks and hydrological management to emergency operations such as mapping landslides and the creation of relief maps, to name a few.

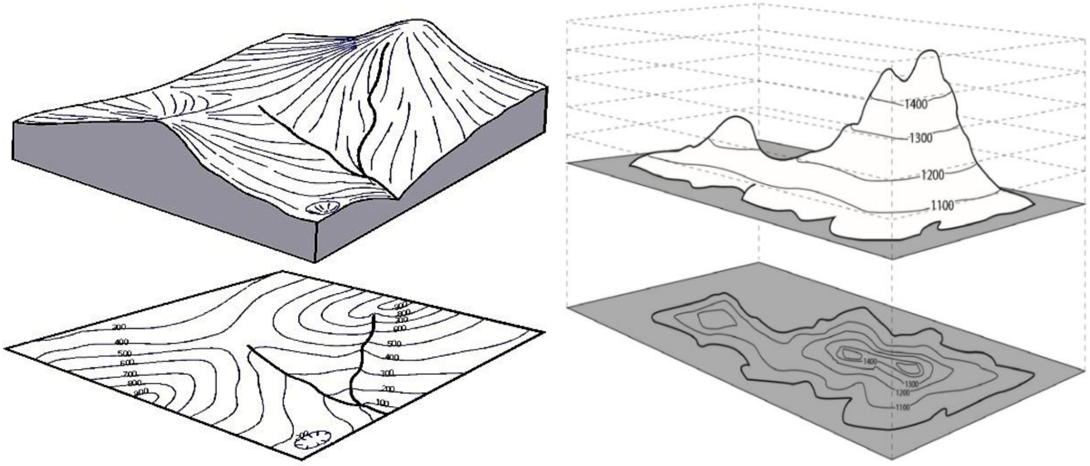


Figure 1.3: Sample Digital Elevation Model from TS [4] [5]

The river network is another important feature represented in the TS, which is extensively used in GIS based research initiatives. It is a networked collection of a wide variety of streams, confluences, and bifurcations as shown in figure 1.4 (page no 6). Computationally, a river network may be perceived as a non-linear organisation of streams. Stream order is a numeric value associated with a stream that reflects its significance to the river network and ranges from 1 to 12. Based on stream order, streams are classified as headstreams, medium streams, and rivers, where headstreams feed the medium streams and the rivers. The shape of the river network may be either dendritic, sub-dendritic, trellied or lattice, radial or concentric, parallel, sub-parallel, rectangular, deranged, centripetal, centrifugal, annular, or violent, depending on the geological and geomorphological characteristics of the landscape.

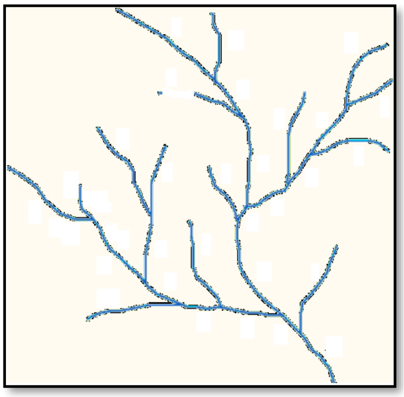


Figure 1.4: Sample River Network

Automatic techniques may be conceived for digitization of river networks and may be advanced further for associating attributes such as stream order, stream number, bifurcation ratio, streams participating in bifurcation ratio, weighted mean bifurcation ratio, stream length, mean stream length, streams participating in length ratio, weighted mean length ratio, and length of the main channel.

With every identifiable morphological feature, contextual annotation is adequately incorporated in the TS for elevating the significance of the feature or for associating attributional values with the features. For example, with contours the elevation values are incorporated either in singular or in intervals. Likewise, names are associated with rivers, transportation networks, and landmarks as shown in figure 1.5 (page no 7). Since the volume of the information content is tremendous, traditional manual approach of extracting the relevant content may be time consuming, expensive, and un-effective. Alternatively, a knowledge enabled (learning based) automatic computational process may be conceived for performing the same. The performance of such automatic processes may be further refined with extensive learning and perfective advancements.

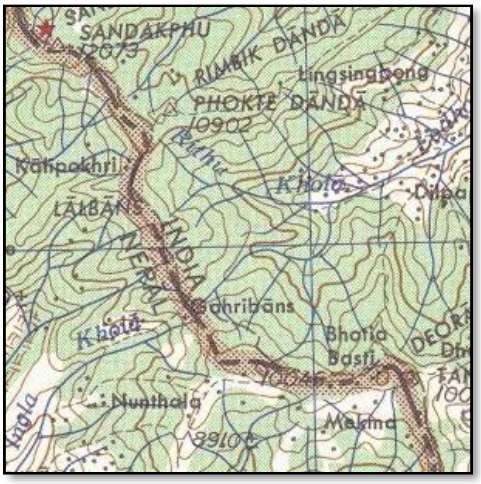


Figure 1.5: Sample Topographic Sheet [1]

1.2 Research Gap

On the basis of a review of related articles, it can be stated that the proposed research initiative should take into account the following crucial aspects while designing a suitable framework for the research objective:

a) The resolution of the TS contributes to a great extent to the complexity of the problem.

b) Feature selection and representation depend greatly on the integrity and effectiveness of computational processes designed for the digitation process.

c) A need-based preprocessing technique deployed in the digitization of various morphological features may lead to a loss of valuable information. Therefore, integrity-preserving restoration mechanisms need to be carefully planned and conceived in such a manner that it does not add significantly to the overall complexity.

d) A TS explicitly portrays some of the important morphological features that are crucial to GIS-based applications. Some of the prominent features are river networks, annotations (including elevation and text), and contours.

e) A river network is a collection of streams with varying significance, confluences, and bifurcations. Computational processes planned and conceived for efficiently characterizing river networks should be capable of identifying various network characteristics as well as generating associated attributes.

f) Annotations in the form of elevation values associated with contours and names associated with other morphological features add necessary semantics. Therefore, its effective localization and recognition through effective computational processes are crucial for associating qualitative and quantitative values with the morphological features.

g) The morphological landscaping of a terrain is expressed using contour lines. Elevation values are associated with contour lines (individual or at intervals), which are used for transforming the contours into an elevation model in 3D space. Computational processes need to be suitably crafted for identifying contours, maintaining their continuity, and regenerating shape while preserving contour lines wherever deemed necessary.

h) A need-based accuracy assessment is deemed necessary for quantitatively justifying the ability of the conceived computational processes.

1.3 Problem Definition

"A traditional approach to digitization incurs greater effort, time, and cost investment with a probability of compromise on quality and integrity, which will definitely have an adverse impact on its reliability."

1.4 Research Objectives

With light on the above-stated issues of concern, the research initiative should be motivated towards the achievement of the following articulated goals:

a) Conceptualize and realize a knowledge-based computational process for digitization of certain morphological features from a TS preserving it structural integrity and its need-based enhancement.

b) Identify morphological features such as river network, annotations, and contours along with their qualitative and quantitative attributes.

c) An integrity preserving, low-cost, and computationally effective interpolation technique for projecting the features in the 3D space for effective visualization.

1.5 Proposed Solution Framework

In order to realize the aforementioned objectives, the research initiative implements the following module, as represented in figure 1.6 (page no 11):

a) Colour segmentation-based feature selection mechanism is deployed for digitizing features of interest from the TS. Further, purpose specific filters are deployed for eradicating insignificant elements as well as for incorporating significant elements omitted as a consequence of imprecise segmentation.

b) Devise supervised learning-based approaches for feature selection and the generation of basic attributes.

c) Devise methods for thresholding, noise removal, thinning, elimination of multiple paths, and traversal mechanism based on the demand of the feature under consideration. In situations where the loss of information is inevitable, a need-based automated process for restoration is mandated.

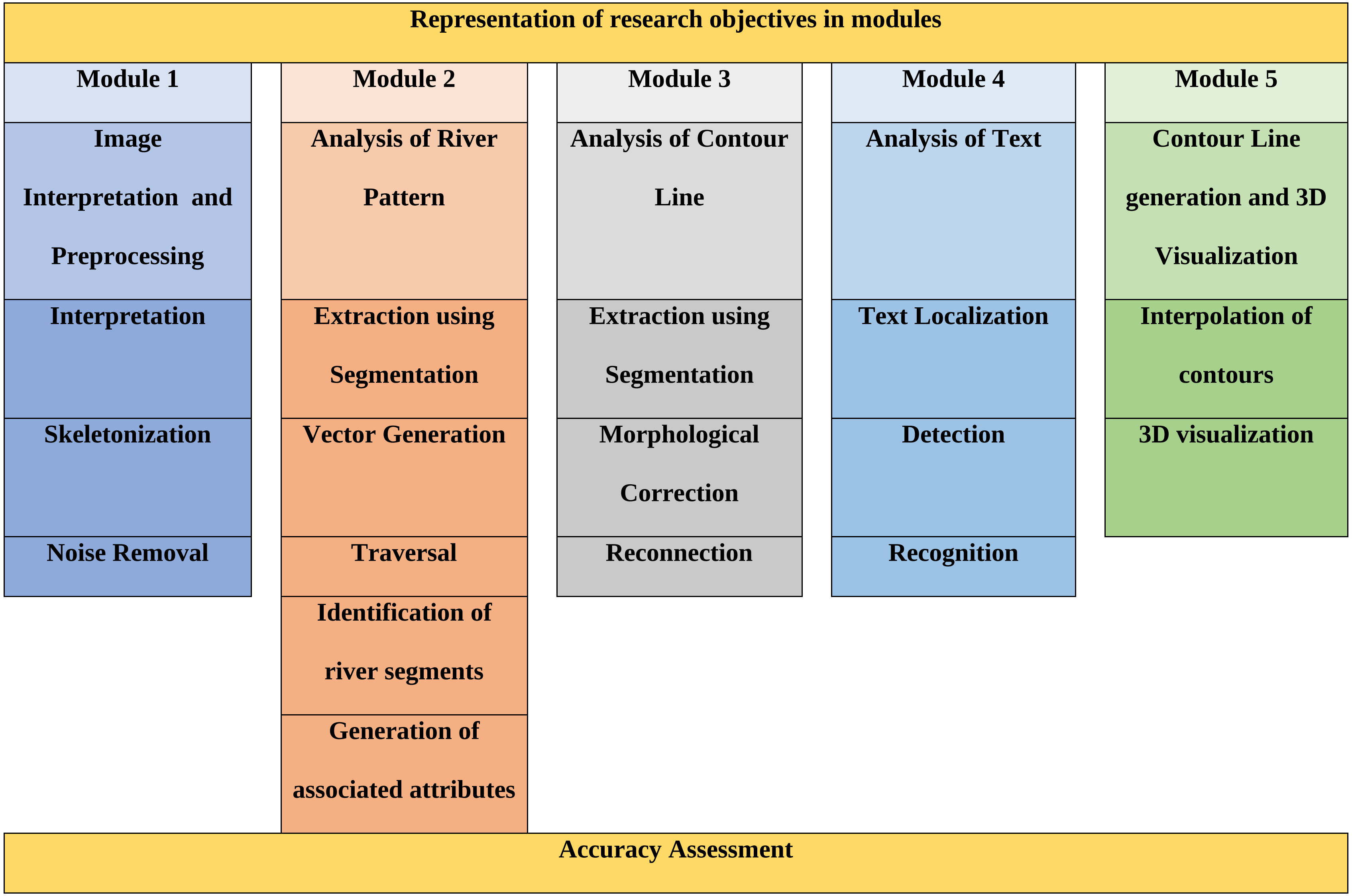
d) This research initiative is motivated towards digitization and processing of geomorphological features, namely, river networks, elevation values, and contours from a TS.

e) Conceive and realize knowledge-based automated process for identifying river patterns and their associated attributes, such as stream order, stream number, bifurcation ratio, streams participating in bifurcation ratio, weighted mean bifurcation ratio, stream length, mean stream length, streams participating in length ratio, weighted mean length ratio, and length of the main channel.

f) Conceive and realize learning-based automated process for localizing annotations in a TS and associating aligned semantics with the same.

g) Deploy a cost-effective and efficient traversal mechanism for the identification and segmentation of contour lines. In addition, conceive methods for restoring missing information as well as for generating additional information necessary for feature refinement. Further, devise a method for transforming digitized contours into 3D space.

h) Correlate the results obtained through the various conceived modules with the existential ground truth reality, considering selective samples.



Representation of research objectives in modules

Module 1	Module 2	Module 3	Module 4	Module 5
Image Interpretation and Preprocessing	Analysis of River Pattern	Analysis of Contour Line	Analysis of Text	Contour Line generation and 3D Visualization
Interpretation	Extraction using Segmentation	Extraction using Segmentation	Text Localization	Interpolation of contours
Skeletonization	Vector Generation	Morphological Correction	Detection	3D visualization
Noise Removal	Traversal	Reconnection	Recognition	
	Identification of river segments			
	Generation of associated attributes			

Accuracy Assessment

Figure 1.6: Block diagram representing different modules for proposed research

1.6 Thesis Contribution

The following are the valuable contributions made by this research work in the design and development of a fully automated computational system for facilitating digitization of features in TS and creation of DEM:

a) Identification, extraction, and representation of morphological features

b) Classification of river pattern and generation of associated attributes

c) Identification of contour lines, its tracing, reconstruction, and representation

d) Localization, detection, and recognition of text from a TS

e) Generation of intermediate contour lines, and its 3D visualization

1.7 Potential for Applications

The objectives and the abilities of the algorithmic procedures proposed in this research initiative has been conceived to address requirements of variety of geo-morphological studies such as:

a) Automatic extraction of contour lines, their refinement, and subsequent generation of shape preserving morphological elements are necessary for preserving the integrity and continuity of the feature of interest. Such procedures would prove effective in studies related to landscaping, land use planning, and the generation of elevation models for terrain analysis.

b) Automatic extraction of river networks, their refinement, and subsequent generation of attributes such as stream order, stream number, bifurcation ratio, streams participating in bifurcation ratio, weighted mean bifurcation ratio, stream length, mean stream length, streams participating in length ratio, weighted mean length ratio, and length of the main channel, which are crucial for hydrological analysis. Such procedures combined with elevation models would help in hydrological modeling, and when combined with satellite data, they would also help in effective flood mapping and its zonation.

c) Automatic extraction of texts and their refinement would prove crucial for associating semantics with morphological features such as elevation with contours and name with other morphological features. If accurately identified, then the elevation values may be put to use for the generation of an elevation model.

d) The product of DEM has wide applications stretching from scientific, commercial, industrial, and operational. Hydrological modeling, morphological analysis, climatic impact behavior, etc. are some examples of scientific applications. In the commercial sector, geological exploration, planning, construction, etc. are a few examples. From the perspective of industrial applications, aviation, telecom, tourism, mining, etc. are a few examples. The DEM can also be used to extract terrain parameters for planning highways, modelling water flow or mass movement (for example, landslides), creating relief maps, etc., and many more. A precisely constructed DEM can be further transformed into a DSM and DTM for better visualization and representation.

1.8 Organization of the Thesis

a) **Chapter 1: Introduction:** It presents different morphological features represented by TS. It highlights the importance of developing a fully automated computational system for the extraction, refinement, and generation of attributes associated with various morphological features represented in the TS. Here, the motivations behind these research initiatives have been categorically expressed. The methodological framework adopted for the proposed research initiative has been highlighted in this section. Further, various scopes for the use of such a process have been discussed.

b) **Chapter 2: Literature Survey:** It highlights various achievements made by researchers globally in the field of automatic digitization. The different techniques and methodologies adopted by researchers and scholars for feature extraction, refinement, and representation of regions of interest (ROI) have been critically reviewed with regards to methodology used, achievements made, advantages, disadvantages, challenges encountered, and scope for future advancements. This cross-sectional analysis greatly helped and assisted in building the research initiatives.

c) **Chapter 3: Extraction of River Pattern, its Refinement and Attribute Generation:** In this section, an effort has been made to understand various types of river patterns and the significance of the attributes related to them. This section aims at digitizing river segments, their refinement, and implementing eight river ordering techniques: Classic Stream Order, Strahler Stream Order, Horton Stream Order, Shreve Stream Order, Scheidegger Stream Order, Order by Path Length, Consistent Stream Order and Cumulative Stream Order. Associated with each of these ordering techniques is the generation of attributes, namely stream order, stream number, bifurcation ratio, streams participating in bifurcation ratio, weighted mean bifurcation ratio, stream length, mean stream length, streams participating in length ratio, weighted mean length ratio, and length of the main channel.

d) **Chapter 4: Extraction of Contour Lines, its Refinement and Attribute Generation:** In this section, an effort has been made to understand the morphological significance of contour lines and their role in the generation of 3D visualizations of a given geographical landscape. Here, the contour lines are digitized and refined to ensure connectivity arising from an inaccurate segmentation process.

e) **Chapter 5: Extraction of Associated Text, its Refinement and Attribute Generation:** The TS contains different types of text, namely, elevation values and names of features. Moreover, these text features intersect or overlap with other geographical features, making their recognition challenging. This section discusses a machine learning model used for recognizing text present in the TS in an efficient way.

f) **Chapter 6: Generation of Digital Elevation Model from Refined and Generated Contour Lines:** This section presents the morphological operation adopted for interpolating shape-preserving points between the existing contour lines. It deploys an unsupervised technique for identifying all possible structural operators present in a set of sizable samples. These structural operators were then closely analyzed to assign an appropriate angular direction for traversal based on some feature-specific reference points. Eventually, the distance values of the directional movements were appropriately portioned to place suitable operators for the generation of the interpolated contour lines. Finally, the contours are plotted in 3D space to generate an elevation model.

g) **Chapter 7: Summary and Conclusion:** This section highlights the achievements of the research initiative, and it’s the scope for seamless integration to GIS based applications. It also presents the pertinent challenges and possible extensions for future research initiatives.

1.9 Conclusion

This chapter of the thesis illustrates the distinct characteristics of different morphological features represented on typical topological sheets and emerging need for the conception and realization of an algorithmic approach for their automatic digitization for use in GIS-based applications. It highlights the pertinent research gap in cognition, to which the problem definition has been articulated. With the problem definition in sight, the achievable research objectives have been suitably framed. This chapter also presents the proposed solution framework adopted for the attainment of the research objectives. Subsequently, some notable contributions achieved through the research initiative are also presented, with potential for applications in the allied domain.

Chapter 2

Literature Survey

2.1 Types of morphological features and the need for extraction, identification, and categorization

A TS is a scale governed representation that highlights various geomorphological land features with the help of distinctive color codes. These maps are produced in a manner that the features are superimposed for ease of representations which can be distinguished through visual inspection. These land features can be categorically partitioned into either of the following shapes: point, line, or polygon. A landmark can be considered as a typical example of a point feature whereas road network, river network, contour lines and boundaries represent line feature. Likewise, lakes, islands and demarcated enclosures are examples of polygon features.

Inferential studies relating to these morphological features mandates extraction of land features. This activity is termed as Digitization. The digitization of a scanned TS can be performed either using manual, semi-automatic or fully automatic techniques. Manual and semi-automatic techniques mandate human intervention. Whereas a fully automatic technique mandates establishment of comprehensive knowledge pertaining to the features before the initiation of digitization process. The integrity of the result from such technique greatly correlates with the integrity of the knowledge basis. In addition, such techniques leverages on computational capabilities of the hardware system generating outcome within limited time in contrast to manual or semi-automatic techniques. Induction of such knowledge based computational approaches into geomorphological studies has opened numerous opportunities for conceptualization and realization of automatic feature extraction processes minimizing effort and time requirement further elevating the quality and accuracy of the research findings.

These geomorphological features can be further explored and analyze to derive crucial attributes for studies. These attributes can be categorized into either basic attributes or derived attributes. Some of the crucial geomorphological features considered for study in this research initiative are river networks, contour lines, and textual contents embedded in the TS.

River Network is an integral part of TS that highlights streams, confluences and bifurcations. Digitization of the same greatly helps in acquiring valuable information related to the basic attributes related to the river network such as network types, stream length, its coordinate orientation, bifurcations and confluences. These basic attributes can be further analyzed to derive attributes such as stream number, stream order, bifurcation ratio, streams participating in bifurcation ratio, weighted mean bifurcation ratio, mean stream length, streams participating in length ratio, weighted mean length ratio, and length of the main channel. These feature aids in understanding and analysis of river patterns, the source and tributaries thus helps in hydrological analysis.

Contour lines are isolines created by connecting points that are located at same elevation from a given reference point. Contour lines are non-intersecting lines that may take any of the following forms based on the morphological orientation of the landscape: lines, enclosures, and island. These contour lines are labeled with elevation values (in intervals), interpolation can be performed for deriving elevation values for contour lines that does not have an explicit elevation value associated with it. These lines are crucial for creating elevation model often referred to as DEM, which forms the basis of many geomorphological studies related to terrain. The construction of a DEM from a topographic sheet is not a trivial task because of various inherent processes involved in it, such as segmentation of features due to false color, reconstruction of lost features due to the segmentation process, elevation value localization associated with a specific contour for mapping into 3D space, complex elevation feature recognition like overlapped elevation values, and interpolation of projected features in a 3D space, to name a few.

With every identifiable feature represented in the TS, textual annotations are incorporated in order to elevate or associate meaningful semantics. This includes names with features such as, landmarks, rivers, and enclosures, elevation values with contours, etc. These annotations are represented with the help of uniform font type and size in a given representation. Detection and recognition of the same is crucial for adding qualitative and quantitative values to the digitized features. Very little research has been initiated towards development of fully automated digitization of river networks and generation of associated attributes, digitization of contours, its refinement and its automatic projection enabled through localization of elevation values and its recognition.

2.2 Extraction of River Pattern, its Refinement, and Attribute Generation

In nature, a river network is a hierarchical, heterogeneous arrangement of streams and confluences [6] which are often found to be anastomosing or wandering [7] [8]. Wandering networks are seen in braided morphological environments, whereas anastomosed networks are seen in flood plains that are divided by islands [9]. River network has in it river segments, confluences and bifurcations.

The river segments in a river network are assigned a positive whole number called the Stream Order [10] that marks the relevance of the segments to the river network. The process of associating order with the segment in the drainage network is referred to as stream ordering [11]. River ordering is greatly influenced by the various mechanisms (stream ordering concepts) adopted for achieving the same. It may be done either by the processing of satellite images, TS, through the analysis of DEM, or through physical observation of the river network [12] [13].

Some of the notable stream ordering techniques used are Classic Stream Order (Hack's Stream Order or Gravelius' Stream Order) [14], Strahler Stream Order [15], Horton Stream Order [16], Shreve Stream Order [17] [18], Scheidegger Stream Order [19] [20],and Order by Path Length [21] [22],

These techniques can be broadly categorized into three categories based on the traversal strategy adopted: bottom-up, top-down, or both. It has been observed that Classic Stream Order and Order by Path Length perform backward traversal, Strahler Stream Order, Shreve Stream Order, and Scheidegger Stream Order perform forward traversal, whereas Horton Stream Order performs both. A set of well-defined ordering criteria makes Strahler Stream Order relatively stable and applicable over other ordering techniques. Computationally, Horton Stream Order is relatively complex and time-consuming compared to the other techniques, but it performs ordering that closely resembles natural occurrences. One of the procedural lags in the cases of Classic Stream Order and Horton Stream Order is uncertainty in reaching the true source, which mandates human intervention to guide the selection of the true source from the mainstream.

Stream order plays a pivotal role in the determination of other network parameters such as Bifurcation Ratio (R_b), Stream Number (N_ω), Stream Length (L_ω), Mean Stream Length ($L_{\omega mean}$), Stream Length Ratio (R_L), Area Ratio(R_L), Weighted Mean Bifurcation Ratio (R_{bwmean}) and Length of Main Channel (Cl) [23] [24] [25].

As per Hortons, the bifurcation ratio varies depending on the landscape, with values ranging between 2 to 3 or 4 on the contrary Strahler stated that the bifurcation ratio is highly stable, with very little variation from region to region with an average value of about 3.5. According to Horton, the number of segments with a given order geometrically decreases with stream order, whereas the mean length of the segments with a given order geometrically increases with stream order. Schumm [26] further added that the drainage basin area geometrically increases with stream order.

The River Continuum Concept (RCC) [27] [28] and process domains [29] in regard to the gradient [30] [6] play a significant role in describing the longitudinal variations in morphological, ecological, physical, biological, and geomorphic characteristics and basin variability [31] [32] that occur along the network. Ecological status is used for tracking the ecological health of rivers and is represented with the help of these categorical values [33] [34] [35]. Buttner et al [21] established relationships between various river parameters such as urban wastewater-river discharge fraction (UDF), agricultural land use fraction (ALF), and ecological status with the stream order.

Some of the most comprehensive river datasets under the regime of the European Union Water Framework Directive (EU-WFD) have been established for accessing the ecological status of various water bodies along with the network with the aid of physicochemical and hydro-morphological characteristics [36] [37].

Natural or anthropological causes greatly influence the shape of the river network and the hydrology [38]. Some of the crucial aspects influencing river networks are ecoregion-dependent susceptibility [39], land use and land cover [40] [41], demographic distribution, sprawling [42] [43] [44], carry-over effects [45], convolution of loadings from upstream to downstream [46], flow splitting [47], bank capacity, and flow capacity [48] [49]. Temporal analysis suggests that geology and tectonics have significantly contributed to the structure and orientation of river networks [50] [51].

With due regard to the above-mentioned facts, it can be inferred that understanding river networks and deriving their associated characteristics is extremely essential for studies related to hydrological and geological analysis. Michael et al [52] have proposed an automated river network processing technique for assuring consistent river network representation in topographic data to address the demand for accurate high-resolution flood hazard mapping for assessing rising river floods and determining suitability for infrastructure development. This work also expresses the disability of the automatic processes, especially in regard to lengthy underground river segments. The methods were applied to a case study in Austria for improving the data quality for flood danger mapping and hydraulic modeling.

Muthusamy et al [53] evaluated the effectiveness of the Digital Elevation Model (DEM) in urban flood modeling using data from a flood occurrence in the region of interest. The proposed technique merged DEMs by combining a higher-resolution DEM for the river channel with coarser-resolution DEMs for surrounding areas for flood modeling.

The crucial process of river capture has an impact on drainage patterns, which in turn influences sediment dispersal and biotic evolution. However, identifying river capture events currently relies on manual observations by geomorphologists, which can be time-consuming and costly. Q. Ma et al [54] introduced an innovative technique for automatic river capture detection based on planform morphology. Huang et al [55] emphasized the importance of topographic data and geomorphological factors in the hydrological response of river basins. It proposes a machine learning approach to forecasting water depth and discharge. The stream network's structure and distribution are taken into account, and a data classification method is used to give the model sufficient performance in predicting hydraulic variables.

Gasnier et al [56] highlighted the importance of river monitoring for both societal and scientific purposes, taking into account the resources rivers provide and the risks that flooding events bring. In order to solve this issue, the author provides a novel river segmentation approach for SAR (Synthetic Aperture Rader) images using a priori datasets like Global River Widths from Landsat (GRWL). The approach combines a conditional random field methodology, historical river centerline data, and a linear structure detector to precisely determine river borders.

River classification is one of the inherent steps involved in understanding a river network. These techniques can be either descriptive or based on a well-defined process. It is often seen that descriptive approaches are quantitative [57] in nature, whereas process-based approaches are qualitative or conceptual [58]. Integration of descriptive approaches with GIS applications aided by aerial photography [59] would greatly aid in the characterization of river networks [60] [6].

Process-based approaches may be combined with the DEM models to facilitate the interpretation of the spatial and temporal patterns of river networks. Classification of the river network has tremendous potential in monitoring the condition of the ecosystem, measuring and monitoring physical and biological parameters, and planning developmental initiatives along with the river network [61] [62] [10] and is largely dictated by the pattern of the river network [29] [63]. It has been observed that most of the research initiatives motivated by the identification of river networks and their characterization either use the DEM or satellite imagery.

Dai, Z., et al [64] proposed a novel automated bottom-up method for stream classification and hydrologic modeling. The suggested method automatically generates the hierarchical structure of dendritic river systems and their related hydrological characteristics by combining digital elevation models, hydrological measurements, and a series of algorithms. The proposed technique efficiently classifies stream segments into different orders and precisely predicts hydrological characteristics like discharge, flow velocity, and travel time. The author also compares the outcomes of the automated system with those of conventional manual approaches and suggests the automated method is quicker, more reliable, and capable of handling large datasets. The suggested automated bottom-up hydrologic coding system can increase the accuracy and efficiency of stream categorization and hydrologic modeling, which has significant implications for environmental management, water resource planning, and assessing the impact of climate change. One of the major limitations is that the system relies heavily on digital elevation models and hydrological observations, which may have errors or inaccuracies. The proposed system is designed for dendritic river systems, which have a hierarchical structure and may not be suitable for all types of river systems, like braided or meandering rivers, which may require different classification approaches. It may also not fully capture the complex physical processes that occur in river systems, such as sediment transport, erosion, and bank stability.

Rieger, W. [65] proposed a two-step algorithm for automatically extracting river networks and catchment areas from the DEM. The algorithm works by analyzing the slope and direction of the terrain, which is represented by the DEM. The river network is identified by tracing flow paths from the highest points in the terrain to the lowest points. The catchment areas are computed by determining the drainage basins for each point on the river network. A comprehensive assessment of the algorithm was performed using DEM from several real-world examples, including a large river basin in China and a mountainous region in the United States, and compared the results with manually delineated river networks and catchment areas, demonstrating that the automated approach achieves high accuracy and efficiency.

The proposed methodology assumes that the DEM data used to model the topography accurately, that the river network is well-connected, and that it can be followed from the highest to the lowest places. It also depends on a particular set of standards and characteristics, which may need to be modified for various terrains and hydrological circumstances. The effects of manmade elements, such as dams, levees, or changes in land use, which can drastically alter the natural river network and catchment areas, are also not considered.

Xue, Y. et al [66] and Ciaburri, C. et al [67] proposed a new technique for extracting information pertaining to mountain river surface and width using high-resolution satellite pictures for managing water resources and hydrological modeling in mountainous regions. The proposed methodology adopts image processing techniques, machine learning algorithms, and river network information to identify and extract river boundaries. It also integrates river network information and digital elevation models to estimate river widths and adjust for terrain slope and relief. The authors admit that seasonal variations in water flow and sediment transport might cause changes in river shape over time and propose that future research could investigate techniques for monitoring these changes.

In the past, several initiatives have been proposed for extracting morphological features of river networks based on automated computer-based extraction procedures [68] [69], DEM [70] [71] [72], reduced scale maps [73] and LiDAR [72].

Most of the research initiatives highlighted above have either used DEM or satellite imagery to extract river networks and associated attributes. These approaches are efficient but greatly rely on the accuracy of the DEM and the resolution of the satellite imagery. Further, such initiatives are realizable only in situations where the researchers have access to corresponding DEM or satellite imagery.

Table 2.1 (page no 28) presented below compares the various river ordering techniques based on approach, number of passes required, advantages and disadvantages, and the need for human intervention.

Table 2.1: Comparison of the various ordering techniques

Sl.	Technique	Approach	Passes Required/ Computation Cost	Advantage	Disadvantage	Human Intervention
a)	Classic Stream Order	Backward/ Bottom-Up	Two/ Low	Simple to use, easy to understand, and intuitive.	• Determination of the true source of the river calls for the assistance of an intelligent decision support system. • Computationally, it requires two passes. 1st pass for determining the order of the mainstream and 2nd pass for assigning the order.	Yes

b)	Strahler Stream Order	Forward/ Top-Down	One/ Low	• It is consistent and has a sound mathematical basis. • Systematic and cost-effective. • Traversal criteria are well-defined, reliable, easy to understand, and easy to use.	• In a situation, if there are streams with different orders arriving at a confluence, it considers the influence of only the highest order stream for deciding on the order of the resulting stream, while it ignores the lower order streams. • It does not allow for distinguishing the mainstream from another stream which in turn will hamper the statistical and analytical processes.	No

c)	Horton Stream Order	Forward/ Top-Down Backward/ Bottom-up	Two/ High	It prioritises the river segments in a manner that closely resembles natural occurrences.	• The network should satisfy “Horton Net”. • An effective decision is to be made at each confluence to determine the true source of the river. • Computationally complex as it is executed over 2 passes. In the 1st pass, Strahler Stream Order is used to order the stream and then the network is backtracked for assigning Horton order.	Yes
d)	Shreve Stream Order	Forward/ Top-Down	One/ Low	• It is intuitively simple. • It considers the influence of all the streams for	• The network should satisfy “Horton Net." An effective decision is to be made at	No

				determining the order of the resulting stream. • Traversal criteria are well-defined, reliable, easy to understand and use	each confluence to determine the true source of the river. • Computationally complex as it is executed over two passes. In the first pass, Strahler Stream Order is used to order the stream, and then the network is backtracked to assign Horton Order. • Identification of the actual source is difficult.	
e)	Scheidegger Stream Order	Forward/ Top-Down	One/ Low	• It is intuitively simple. • It takes into account the influence of all the	Identification of the actual source is difficult.	No

				streams in determining the order of the resulting stream. • Traversal criteria are well-defined, reliable, easy to understand, and easy to use.		
f)	Order by path length method	Backward/Bottom-up	One/Low	Traversal criteria are well-defined, reliable, easy to understand, and easy to use.	Identification of the actual source is difficult.	No
g)	Consistent Stream Order	Forward/Top-Down	One/Low	• It takes into account the influence of all the streams in determining the order of the resulting stream.	• Identification of the actual source is difficult. • Computation intensive.	No

				• Traversal criteria are well-defined, reliable, easy to understand, and easy to use.		
h)	Cumulative Stream Order	Forward/ Top-Down	One/ Low	• It establishes a direct relationship with discharge as it takes into account the influence of all the streams in the watershed when deciding on the order of the resulting stream. • Traversal criteria are well-defined, reliable, easy to understand, and easy to use.	Identification of the actual source is difficult.	No

2.2.1 Inferences drawn from the existing literature

Some of the inferences drawn from the study of associated literature are:

a) The River Network is a hierarchical arrangement of streams of various degrees of influence, represented with the help of an ordinal value called Stream Order.

b) There are identifiably eight stream ordering techniques: classical, Strahler, Horton, Shreve, Scheidegger, Order by Path Length, Consistent, and Cumulative.

c) There are many attributes that may be associated with streams in a river network, some of which are as follows: Stream Order, Stream Number, Bifurcation Ratio, Streams participating in Bifurcation Ratio, Weighted Mean Bifurcation Ratio, Stream Length, Mean Stream Length, Streams participating in Length Ratio, Weighted Mean Length Ratio and Length of Main Channel.

d) These attributes play a crucial role in studies related to river networks such as morphological (dynamic) change analysis, flood mapping, tracking the ecological health of river networks, etc.

e) These river networks can be drawn either through the analysis of DEM or the physical observation of topographic sheets.

f) Traditional research initiatives motivated by the morphological analysis of river networks deployed manual approaches for the digitization of river networks.

g) Manual approaches are extremely time-consuming and expensive; moreover, the quality of the results of such digitization techniques is often found to be highly inaccurate and subjective in nature.

h) In addition, quantifying the values of the associated attributes is also extremely labor-intensive and time-consuming using such manual approaches.

i) Semi-automatic techniques overcome some of the limitations of the manual approach, but increased human interference makes such processes highly ineffective and inefficient.

j) Therefore, it is essential to conceive and realize fully automatic techniques for extraction, pre-processing, morphological refining, traversal, ordering, and generating attributes related to a river network.

k) Computational time and space requirements should be essential considerations while devising such methods for the attainment of the above-highlighted objectives.

2.2.2 Research motivations

The proposed research initiative is motivated by conceiving an efficient generic computational programme capable of sequentially performing the stated activities for addressing the research gaps identified during the review of related works,

a) Color-segment the river network from the topographic sheet.

b) Pre-processing the segmented image to eradicate noise.

c) Skeletonize the pre-processed segmented image to create features with single-pixel width.

d) Resolve m-connectivity to facilitate efficient processing.

e) Determine the terminal streams using an efficient spiral traversal mechanism.

f) Identify all the stream segments by identifying all the confluences in the river network.

g) Order the identified streams as per the principles of Classic, Strahler, Horton, Shreve, Scheidegger, Order by path length, consistent, and cumulative ordering techniques.

h) Associate attributes with river networks such as stream order, stream number, bifurcation ratio, streams participating in bifurcation ratio, weighted mean bifurcation ratio, stream length, mean stream length, streams participating in length ratio, weighted mean length ratio, and length of the main channel based on the principles of the ordering techniques.

2.3 Extraction of Contour Lines, Their Refinement, and Attribute Generation

The design and development of a fully automated system for extracting contour lines and reconnecting broken contour lines over time has been a challenging task. There has been several initiatives executed in the past for the reconstruction of broken contour lines using unique concepts and approaches for establishing a connection between the broken ends. Through the review, it was witnessed that reconnection of broken contours suffers from three major problems: localization of broken contour endpoints, selecting the best matching breakpoint pair, and efficiently reconnecting end pairs to preserve the continuity of the contour and its integrity. Identifiably, there are three ways of connecting broken contour lines: pixel-adjacency-based, continuity-based, and gradient-flow-based [74].

In pixel-adjacency-based approach, the terminal points are determined based on the similarity of intensity values in the adjacent pixels. Further, it deploys line drawing algorithm for establishing continuity. This approach mandates creation an adjacency matrix for determining fitting neighbors. The process incurs greater time and effort in situation if the input image is not adequately pre-processed.

The continuity-based technique makes use of either geometric or topographic features, or both. The same has been used by Chengming Li et al [74], wherein Frechet distance is used for selecting a similarity-based reference line based on which the broken endpoints are joined. However, it proves ineffective in situations where the resemblance between a selected reference line and the broken contour line is considerably low. Another commonly used Continuity-Based approach makes use of the Euclidean distance measure to reconstruct broken endpoints by drawing a straight line in an auxiliary direction [75] [76]. This method works effectively for plain surfaces and smaller breakpoint gaps but does not preserve the integrity of the original contour lines.

Another widely used technique in raster images for contour reconstruction that preserves the continuity of broken contours is the geometric-based approach. Nikita D. [77] has proposed a complex method that uses geometric and relative grouping, maximum likelihood estimation, etc. for connecting broken contours in low-quality maps. E. Hancer, R. Sametet, et al [78] have adopted a geometric-based approach where a hybrid method is proposed to increase the efficiency of reconnecting by using geometric features as a basis for recognition.

Wang Feng et al [79] used multiple incremental back-traced pixels for computing the weighted average directional angle to fill the gaps between broken contours. In addition, an angle control mechanism was used to handle complex topographic surfaces. To identify matching endpoints, the deviation angle of the endpoint and weighted Euclidean distance were used.

One of the important aspects of reconnecting a broken contour is identifying the broken end-point pair through a comprehensive problem-specific search process. Pradhan et al [80] have designed three different search techniques for locating broken end points of contours based on leech movement, water-flow movement, and wiper movement. It was observed that in the leech approach, the search space increased exponentially with the increase in the distance between endpoint pairs, thereby increasing computational complexity. However, by creating variable-sized sectors as search spaces in the water flow and wiper movement techniques, the author was able to efficiently manage the computational complexity. Further, it was also concluded that the technique is efficient in a scenario where the breakage of a contour is small.

Bin Xu et al. proposed a mathematical formulation for calculating the probability of the endpoint being connected [81]. The probability of broken contour re-connection was found to be inversely proportional to the angle between tangent lines and the distance between the breakpoints. In some situations, the Euclidean distance between various candidate end-point pairs was defined, and those points with the minimum distance were joined.

A similar approach was also adopted by Khotanzad et al., using the A* search algorithm [82]. This algorithm works on the principle of selecting a path from the A* adjacency graph based on the minimum distance to fill broken gaps [79] [83] [82]. Sadia Gul et al. proposed a technique for finding matching pairs and connecting them using an incremental search window in which the endpoints are extended using tan-1 (Δy/Δx) for reconnection [75].

Further, the Delaunay triangulation and Voronoi diagram method were used [83] to interpolate the contour end-points, using the concept of the medial axis. Xin et al [84] [81] proposed a gradient model to retrieve contours and their reconnection. But, this method was found to be relatively slow and complex for practical purposes. Edge linking, the Newton Interpolation method, the curve fitting technique based on the regression model, Cubic spline interpolation, etc. [85] [86] [75] may also be used for joining related endpoints. It was stated that [87] Chaikin's algorithm works with control polygons directly, unlike Bezier, is similar to a quadratic B-spline curve method, and provides an effective mechanism for drawing curves [87]. However, the robustness of the algorithm lies in how accurately it interpolates the contour points for complex terrains or terrains with severe breakage in contour lines to be reconstructed [74].

Many researchers have proposed a semi-automatic approach making use of distance and angle as a basis for reconnection, whereas some have used geometric or topographic characteristics for reconnection. But it was found that such methods may demand greater human intervention.

Mansourifar et al [88] [89] [90] have discussed the geometric problem of data fitting for drawing curves in the case of real-time systems, considering constraints like less power usage, a minimum number of segments, a lesser execution time, etc. In the case of a larger dataset, the traditional approach (iterative approach) demands greater usage of time for fitting curve data points. However, the proposed method uses a non-iterative approach based on shoulder point detection and a quadratic rational Bezier curve for fitting data points with minimum execution time and realistic accuracy [89] [90] [91].

Amirkhani et al [88] have proposed an evaluation method based on the subject of interest as perceived by the human eye for estimating vision density characteristics for assessing the quality of the image. The proposed technique uses penalty and compensation metrics to make the assessment procedure more reliable. It was also stated that the assessment of results may be either objective or subjective based on the availability of the reference map. The subjective method is most useful in situations where a reference image is not available for assessment of the results obtained; therefore, estimated vision density characteristics of the human eye may be relied upon.

Table 2.2 (page no 41) presented below compares the various techniques for reconnection of contour lines considering strategy used for detection of terminal points and reconnection mechanism adopted, along with computational complexity, its advantages, and disadvantages.

Table 2.2: Comparison of various techniques for reconnection of contour lines

Sl. No.	Description	Strategy used for detecting terminals	Strategy used for Reconnection	Components of Computational Complexity	Advantage	Disadvantage	Motivation
a)	Pixel-adjacency-based reconnection	Determine terminal points based on the similarity of intensity values in the adjacency matrix.	▪ Line drawing algorithm	Creating an adjacency matrix Determining neighbors' reconnection	▪ Simple and easy to understand. ▪ Pixels in close proximity with similar intensity values may be identified as terminal points.	In this situation, if the input image is not binarized, determining the similarity of intensity values may be difficult and time-consuming.	▪ The determination of terminal points should be proximity-based. ▪ The image should be binarized.

b)	Continuity-based reconnection	Determines terminal edges based on orientation energies, alignments, or locations of the edges, exploiting curvilinear continuity.	Using curves based on the orientation reduces the elastic function.	▪ Edge detection. ▪ Determining the curve with the minimum elastic function. ▪ Placement of the curve between the edges.	Generation of integrity-preserving curves for reconnecting broken contour lines.	▪ A continuity-based approach never provides a closed, connected region. ▪ Hard decisions may lead to improper detection of edges for reconnection. ▪ It does not efficiently handle textured regions.	Hard decisions should be avoided, as intensity values are highly dynamic.

c)	Gradient-flow based reconnection	Here, the Riemannian metric is defined and combined with gradient flow to refine the snake model for determining terminal points for reconnection.	The Euclidean metric is used for reconnection.	▪ Creating a Riemannian metric. Combining gradient information with a Riemannian metric refinement of the snake model.	Determination of the optimal search space for locating terminal points.	▪ Computationally complex and time-consuming.	▪ Backtracking is essential for determining the nature and characteristics of contour lines. ▪ Directional information is essential for framing optimal solutions.

d)	Chengming Li et al [74]	To ensure continuity of broken contour lines, geometric or topographic features or both neighbouring contours are considered for determining terminal points.	Densification of contour points is performed to reduce the proximity between the terminal points. Subsequently , terminal points within proximity are reconnected to ensure continuity.	▪ Determining the geometric or topographic features of neighbouring curves. ▪ Densification of contour points. ▪ Reconnection of terminal points.	▪ Integrity preserving. ▪ Generates a greater number of control points through densification, resulting in the generation of smooth curves.	▪ Computationally intensive. ▪ Generates a greater number of control points. ▪ Heavily relies on the morphological orientation of neighbouring contours. ▪ The creation of a greater number of control points	Morphological information from adjacent contours proves decisive in the design of reconstructed contour lines.

						may lead to incorrect connections.	
e)	Geometric based method	The approach is based on crucial morphological information of contours, such as coordinate and geometric orientation and gradient, for determining terminal points for reconnection.	▪ Cubic Spline Interpolation or Newton Interpolation Method.	▪ Determining geometrical orientation in contour lines. ▪ Determining gradient.	▪ Computationally simple and easy to implement. ▪ Suitable for handling simple breakages with a smaller distance between terminal points.	▪ Inappropriately handles severe breakage along contour lines. ▪ Miss-connections between the terminal points.	The morphological orientation of contour elements can be pursued as the basis for determining terminal points.

f)	Maximum-likelihood based approach	Here, to reduce the size of search space, a probability function is associated along the directional alignment of the contours as it is observed that the probability is not equally likely along	Line drawing algorithm.	▪ Determining the probability distribution along the various angular orientations. ▪ Reconnection of broken contour lines.	Highly accurate and creation of optimal search space for determining terminal points.	▪ Computationally complex and intensive. ▪ The search mechanism should be aided by a quality decision-support system. ▪ Highly subjective and often prone to bias.	The association of probability distribution with other contour features such as gradient and orientation would enable the selection of optimal terminal points.

		with all directions.					
g)	Weightage d-average directional angle	Terminal points are determined based on the average directional movement among the adjacent coordinates.	Line drawing algorithm.	▪ Back-tracking for determining the average of directional movement. ▪ Reconstruction of lines.	▪ It takes into account the cumulative influence of the alignments along with the contour coordinates. ▪ It creates an optimal search space for determining	It requires backward recursion for determining the average of directional movement.	Backtracking is essential for determining the nature and characteristics of contour lines.

					terminal coordinates.		
h)	Leech movement	▪ Terminal points are determined by expanding the search window incrementally around the point of breakage. ▪ Inter-contour distance was set as the	Bresenhems line drawing algorithm.	▪ Creation of a search window for the determination of terminal points. ▪ Reconstruction of lines.	▪ Simple, easy to understand, and easy to implement. All possible adjacent coordinates are searched for in determining the terminal points.	Computationally inefficient, as search space exponentially grows with distance.	The localization of search space is crucial for managing computational complexity.

		criteria for expansion.					
i)	Water-flow movement	Terminal points are determined by creating non-uniform sectors formed by aligning the flow of direction at different angles.	Bresenhem's line drawing algorithm.	▪ Creation of non-uniform sectors for the determination of terminal points. ▪ Reconstruction of lines.	▪ Creation of a variable search space based on the angle of alignment. ▪ Requires considerably less search time compared to the Leech approach.	The creation of non-uniform sectors is computationally complex and intensive.	The direction of alignment should be one of the fundamental basis for determining terminal points for reconstruction.

j)	Wiper movement	Terminal points are determined by creating uniform sectors formed by aligning the wiper at different angles.	Bresenhem's line drawing algorithm.	▪ Creation of uniform sectors for the determination of terminal points. ▪ Reconstruction of lines.	▪ Creation of a uniform search space using a wiper along the angle of alignment. ▪ Requires considerably less search time compared to the leech and water-flow approaches.	The creation of uniform sectors may result in a non-optimal solution.	The direction of alignment should be one of the fundamental basis for determining terminal points for reconstruction.

k)	Bin Xu et al [81]	▪ A probability-based approach is used for determining terminal points for reconnection. Here, the probability of reconnection is inversely related to the distance between the	Reconnection is based on proximity.	▪ Determining breakage. ▪ Determining terminal points and associated probabilities based on distance. ▪ Reconnection.	▪ The nearest terminal points will be selected for reconnection. ▪ Suitable for handling small breakages.	▪ Computationally intense and may often lead to the selection of inappropriate terminal points for reconnection. ▪ Fails to handle severe breakage.	Selection of optimal points based on a distance metric.

		identified terminal points.					
l)	A* adjacency graph based approach	Terminal points are determined based on the shortest distance between the breakage points.	Line drawing algorithm.	• Determining a set of terminal points with a distance vector. • Selecting the best terminal pair based on the A* approach.	• Computationally simple and easy to implement. • Significantly reduces search space based on distance value.	Misconnection due to inappropriate selection of terminal points, as the distance may not only be deemed sufficient for reconnection.	The designed approach should be able to constrict the list of potential terminal points for reconnection.
m)	Sadia Gul et al [75]	Terminal points are determined by growing the search space	The shortest distance is based on Euclidean	• Preprocessing operations such as thinning,	Simple, easy to understand, and easy to implement.	Inappropriately handles severe breakages and often fails to	Preprocessing operations such as colour segmentation,

		along with the breakage points incrementally.	values and tangents.	removal of holes, and bifurcation, and noises. • Determination of terminal points. • Determining the tangent or angle of alignment. • Reconstruction of lines.		maintain contour continuity.	binarization, thinning, removal of holes and bifurcation, and noise reduction are essential for reducing the computational complexity involved.
n)	Delaunay triangulation and	Terminal points are determined based on	Line drawing algorithm	• Triangulation. • Selection of optimal points	Effective as it considerably reduces search	Computationally intense and time-consuming.	Geometrical orientation may be

	Voronoi diagram based approach	criteria for triangulation.		for reconnection. • Reconnection of the line	space for terminal points.		pursued for determining the terminal points.
o)	Xin et al [84] [81]	A gradient is used for determining terminal points for reconnection.	Reconnection is achieved by using the Generalized Gradient Vector Flow (GGVF) Snake model.	• Determination of search space based on gradient. • Reconnection based on the GGVF Snake model.	• Accurate and effective as the determination of search space is based on gradient and reconstruction is based on the snake model.	Computationally intense and time-consuming.	It is essential to preserve the structural integrity of contours while performing reconnection between broken terminal points.

					- Gradient reduces the search space, and GGVF retains the structural characteristics of the contours.		
p)	B-Spline Curve Method	Here, a B-spline (polynomial) is used to appropriately select or approximate	Spline technique	- Designing a suitable function for a spline. - Determination of terminal points.	Requires a limited number of control points and integrity preservation.	Computationally intense.	It is essential to use an optimal number of control points to retain the characteristics

		terminal points for reconnection with the least number of control points and polynomials.		• Reconnection of terminal points based on the spline.			of contours during reconstruction.
q)	Chaikin’s Algorithm	Here, a limited number of control points are used to generate a smooth curve by cutting	Recursive corner-cutting approach.	• determining the fragmentation point. • Line reconstruction. • Smoothening.	Generates integrity-preserving curves.	Computationally intense as a consequence of the recursive approach.	It is essential to retain the geometric structure of the contours during reconnection.

		corners recursively. This approach may be adopted for determining terminal points for reconnection.					

2.3.1 Inferences drawn from the existing literature

Some of the inferences drawn from the study of associated literature are:

a) The efficiency of the reconnection techniques greatly varies with the tone and texture of the input image.

b) To ensure continuity, the input image should be suitably pre-processed for determining similarity essential for establishment of connected components.

c) Hard decisions imposed for selection of end points to establish connection may lead to faulty joins.

d) The complexity and time requirement of the reconnection technique greatly varies depending on the quality of the results from the pre-processing stage.

e) The effectiveness of the shape preserving reconnection technique depends on its ability in determining optimal number of control points and morphological orientation of neighbouring contours.

f) Presence of severe breakage along contour lines, the reconnection mechanism may fail to identify appropriate end points leading to Miss-connections

g) The effectiveness of the reconnection techniques can be further advanced by incorporating suitable search mechanism supported by valued decisions based on directional movement.

h) It is crucial to limit the search space in order to contain the computational overhead which may be achieved through creation of optimal variable sized search space along the direction of the flow.

2.3.2 Research motivations

Through a thorough analysis of relevant literature, the following important derivations were made:

a) Preprocessing operations such as colour segmentation, binarization, thinning, removal of holes and bifurcation, and noise reduction are essential for reducing the computational overhead involved in processing a topographic sheet. While performing image segmentation, hard decisions should be avoided as intensity values are highly dynamic and vary from sample to sample.

b) The strategy for determining terminal points should be proximity-based, i.e., based on inter-terminal point distance.

c) Some of the crucial information that may be pursued for determining terminal points for reconnection may be:

- Morphological orientation of contour elements
- Direction of alignment of contours
- Geometrical orientation of contours
- Morphological information from adjacent contours proves decisive in the design of reconstructed contour lines

d) While performing reconnection, it is essential to preserve:

- Geometric structure of the contours
- Structural integrity of contours

e) While performing reconnection, it is essential to address the following key aspects for reducing computational overhead:

- Use an optimal number of control points to retain the characteristics of contours.

- The design approach should be able to constrict the list of potential terminal points for reconnection.
- The localization of search space is crucial for managing computational complexity.
- Selection of optimal points based on a gradient and distance metric

f) Some of the generic processes involved in similar research initiatives in the past are:

- Backtracking is essential for determining the nature and characteristics of contour lines.
- The association of probability distribution with other contour features such as gradient and orientation would enable the selection of optimal terminal points.

2.4 Extraction of Associated Text, its Refinement, and Attribute Generation

2.4.1 Text Localization

2.4.1.1 Text Localization using image morphology

Localization is the process of automatically detecting the presence of text features in a topographic sheet and presenting them using a bounding box. These features may later be categorized as text features, including elevation values.

The computational process should be able to address critical issues like aliasing effects, pseudo-color, closely placed contour lines, and intersecting features. These demands lead to the incorporation of knowledge into the computational process and its continual refinement to enhance its capabilities. Li et al [92] enlightened the process by providing a framework for localization and recognition of text from a topographic sheet. Also, localization was achieved using complex deep learning techniques; however, the basic principle was not discussed. Also, the major limitation of the proposed method is that it has been tested on a manually prepared topographic sheet using high-end software. Such computer-generated maps are noise-free and less complex as compared to the original topographic sheet. The same accuracy may not be achieved using the same framework if a true topographic sheet is considered for the experiment. The author in [93] highlighted the possibility of having an automated computational process for locating the labels in a topographic sheet using an artificial neural network (ANN). Using this method, 80% accuracy was achieved in an optimal condition. The main limitation of this method is that it is a semi-automatic process.

Many other researchers used the linear features as a basis for extracting text features. Some common linear features, like contours, present in topographic sheets are represented using the same color as elevation values or text features, so it is difficult to separate them. Pezeshk and Tutwiler [94] eliminated such linear features using the modified multi-angled parallelism (MAP) algorithm, retaining only text features. The resultant image was enhanced further for recognition. Such a method is useful only if the map contains straight lines and linear features. However, it may fail if the geographical terrain of the map is complex, like a hilly region with many complex contours (curves), or if the text feature overlaps with other complex morphological features. The author [95] aims at extracting and recognizing symbols; letter matching was used to extract features of interest. The extracted features were further analyzed to recognize them using various shape descriptors.

This method retains symbols as well, in addition to elevation detail or text features that are unwanted. After a thorough survey, it was found that any work pertaining to the identification of (x, y) locations for elevation data or elevation data localization associated with contour has not yet been explored.

This research initiatives aims at adopting a method for designing an efficient, fully automated computational system capable of detecting the location of elevation values or any text features present in a poor quality original topographic sheet using basic geometric and morphological image processing operations.

2.4.1.2 Text Localization using Deep Learning technique

The ability of a computational system to interpret an image, characterize it, and learn its essential features for enabling a decision support system to make quality decisions has to a great extent eased the task of human-machine interaction. Examples include facial recognition [96], biometric identification systems, text recognition [97], autonomous vehicles, mobile robotics, medical image processing, behavior analysis [98], and many more [99]. In order to facilitate effective recognition, it is quite essential to know the object's concept and its position [100]. Informative region selection, feature extraction, and classification are some of the crucial activities required for recognition. Challenges such as diversity in appearances, brightness, contract, sharpness, and varied background greatly influence the accuracy of the recognition process.

The ability of Convolutional Neural Networks (CNN) relies on their multi-layer perceptron, use of local perception, established connections, and variable weights assigned [101]. These attributes of CNN help to a great extent in managing complexity and reducing the number of weights [99]. Features such as color, tone, identifiable shapes, and the image itself can be used to prepare CNN for image classification. In applications that are expected to deal with visual interpretation-based business realization, the incorporation of automatic image classification processes becomes essential. There are various techniques proposed for the same, one of which is based on CNN [102]. It is not only effective, but the results also exhibit high confidence. It is crucial that while attributing the contribution of the various layers, both the top and bottom layers are given priority, as it has been often seen that the lower layers hold in them greater discriminative power than the layers on top [103]. The training can be done layer wise, for effective representation of features [104]. Fusion of all the layers would help elevate the performance [105]. Chauhan et al [106] in their research initiative have used CNN for image recognition on the CIFAR-10 and MNIST datasets. A very high accuracy was achieved on MNIST, and a notably high accuracy was achieved on CIFAR-10. It was further articulated that the accuracy achieved can be elevated by increasing the instances present in the learning set and the instances of hidden layers [107]. CNN can also be used for classifying features in a color image [108] [109]. Jmour et al. have used the "fine tuning technique", wherein the existing refined layers are reused for classifying images. Effective feature extraction is crucial for enhancing the recognition process [110]. This deep learning technique not only automates the process of extracting and learning the features but also has the ability to work with a limited number of training sets [111], thus reducing the time required. CNN has also proven effective in situations where features to be classified are under variable real-world conditions.

Two essential criteria to judge classification mechanisms are their ability to identify features and the time within which the results are generated. Here, two different approaches [112], namely YOLOv1 and YOLOv2, have been proposed. In YOLOv1, for the prediction of multiple bounding boxes, a lone CNN is used, followed by probability estimation for each of the identified boxes. Here, a collection of grids represents an input image. In situations where the object to be classified belongs to a particular grid, it is the responsibility of the grid to detect the object. It is the Individual grid cells responsibility to determine a suitable bounding box with associated confidence and probability. YOLOv2 improves YOLOv1, raising the mAP by using a multiscale learning mechanism.

CNN in combination [113] with image un-sampling techniques and LIDAR data has been deployed for classifying images, and the result of the same is used for enabling an autonomous vehicle to navigate [114]. Such an approach would not only ensure quality but also allow for minimal loss.

Rashid et al [115] have used the CNN model for dehazing an outdoor image, wherein for convolution operations, features of the dehazed image are used in the first few layers of the hidden network. The quality of the hazed image is also influenced by the quality of the features extracted as a reference from the dehazed image, i.e., the tone and texture of the dehazed image [116]. In order to enhance the quality of the result, maps of reduced size were used in the subsequent hidden layers. It has also been observed that the overall time complexity increases with the dimension of the image. Wang et al [117] have improvised CNN by improvising the activation function for extracting the depth feature for 2D images. The images were classified using a combination of the Extreme Learning Machine technique and CNN. It has been articulated that it delivers greater accuracy with reduced execution time. This technique can be used in situations where the features are not obvious.

A high-capacity R-CNN [118] has been proposed to identify feature vectors of known length. These feature vectors are then sent to a class-specific linear SVM for object detection. There are mainly two different aspects to an object detector, namely a feature extractor and a feature classifier. A neural network classifies or makes decisions using the aspects drawn by these two techniques. The ability of CNN, R-CNN, Fast R-CNN, and Faster R-CNN depends on efficient training of the network for enabling object detection, with implications on time, cost, and space [119] [120]. Fast and Faster R-CNN deploys discriminative initialized layers of Convolution for the purpose of extracting features that are independent of region, followed by Multi-layered Perceptron (MLP) based classification. The efficiency of these two techniques lies in the manner in which they optimize the classifier and the bounding box [121]. This research initiative tries to address three important aspects, such as the size of the input, the use of classification networks like ResNet and GoogleNet, and skip connections to speed up the training process [122] [123] [124]. Object detection is definitely computationally complex compared to that of classification, but due to the onset of ConvNets, these two tasks have been considerably improved and eased with increased accuracy [125] [118]. Girshick [126] has proposed a fast version of R-CNN known as Fast R-CNN that aims at increased testing speed, faster detection, and increased accuracy. Compared to the traditional approach of a multi-stage pipelined test model, which is rather slow in context to training and detection with increased computational cost, the proposed research initiative uses a single-stage training model. This model learns the objects to classify as well as refines their respective locations. In contrast to R-CNN, this approach has higher detection quality (mAP), multi-task loss-based single-stage training, rapid update of network layers, and non-usage of storage for feature caching. With the use of a deep network, Fast R-CNN is capable of detecting objects at a very fast rate, nearing real time, excluding the time spent on region proposal [127]. For enhancing the ability of an object detector, it is essential that it have a reliable region proposal technique embedded in it. Currently, the running time of the detector networks has been reduced to a great extent due to the onset of Fast R-CNN. In this approach, a well-trained region proposal network is combined with Fast R-CNN for efficiently determining object boundaries and objectless scores [121]. Cao et al [128] have proven an improvisation on the existing Faster R-CNN for detecting objects in an image. A two stage detection approach has been conceptualized, wherein for the bounding box, an intersection over Union (IoU)-based improved loss function is used, followed by bilinear interpolation for improving the RoI. To enhance the information content present in the feature map, multi-scale convolution has been used, and to avoid loss due to overlapping objects, improved non-maximum suppression (NMS) has been used. Alamsyah et al [129] have devised a mechanism for detecting fingertips using Faster R-CNN and the Inception V2 architecture in CNN to reduce computational time [130]. This is done in order to cut short the preprocessing time required, therefore enhancing efficiency. Various geometric structures, geometric features [131], color, and depth information [132] [133] may also be used for detecting various objects. Faster R-CNN [134] takes a considerable longer time compared to YOLO [135] and Single Shot Multi-Box Detector [136], but Faster R-CNN comparatively performs better. In applications where the quality of the detection process is of the essence, Faster R-CNN proves effective. A faster R-CNN-based [137] improvised model has been conceptualized for detecting vehicles in real time using MobileNet architecture [138], resulting in reduced computational cost and time. Instead, in this work, soft non-max suppression is employed along with context-aware RoI pooling [139] [140] [125].

Although text detection and recognition techniques have achieved significant gains, they lack efficiency when it comes to real-world natural scenes containing textual information [141] [142] [143] [144] S. Shivajirao [141] proposed a novel framework inclusive of a pre-processing step (deblurring and sharpening input samples) for noise reduction to enhance accuracy. Subsequently, a Faster RCNN based on cascading RPN (Region Proposal Network) was used for detection and recognition. However, N. Gandhewar [142] emphasized the tradeoff between speed and precision of results for discovering text in natural images; hence, the Yolo V5 framework was used, in contrast to the RCNN Family, which is a two-phase detector. The model was built with less training time and a limited model size, and its performance was validated on standard datasets like ICDAR 2015, ICDAR 2013, ICDAR 2003, SVT, MSRA-TD500, considering F-measure, Precision and Recall performance metrics. The model detects horizontal text as well as vertically aligned text but fails to support the detection of multilingual text. Text detection from natural scenes is extremely challenging as text present in natural scenes or topographic sheet varies in scaling, color, brightness, type, and orientation, and more often than not, these features intersect or overlap with other features [143] [144]. Joseph Raj [143] highlighted the degradation of algorithm performance in natural scenes containing bilingual text. To detect such types of text, the author used a state-of-the-art technique consisting of three basic steps. Initially, extract probable text regions using Faster RCNN, followed by rearrangement of the text region for extracting shape features from three orthogonal planes, and eventually classifiers are used to predict features as textual or non-textual information. A notable F1 score of 0.7 on the MSRA-TD500 was achieved. Lingqian Yang [144] presented a summary of deep learning-based text detection algorithms for natural scene text detection under two broad categories: Region proposal-based and Semantic Segmentation. The region proposal-based algorithms are Connectionist Text Proposal Network (CTPN), Segment Linking (SegLink), and Attention-Based Cloud net (ABCNet), whereas Semantic Segmentation includes Efficient and Accurate Scene Text (EAST) algorithm, PixelLink, and Progressive Scale Expansion Network (PSENet). CTPN, which is based on basic Faster RCNN, has the ability to detect multi-scaled text features using a single-sized sliding window, however, it fails to detect non-horizontal text. SegLink, based on CTPN, which is rotation-invariant, can detect text aligned at different angles. However, the attainment made by ABCNet by being able to detect curved text using the Bezier Curve Technique is inspiring. Although deep learning has achieved impressive results in the field of text detection and recognition, it requires huge amounts of labeled data for modeling. X. Y. Ding [145] motivates us to adopt an unsupervised technique where huge amounts of labeled data are not available. Also, annotating data for modeling is an extremely time-consuming and expensive process. A framework with five different modules was used to develop a recognition model using Dual adaptation and Clustering (DOC). It initially extracts global features, then deploys a global discriminator for global level visual feature adaptation, followed by the extraction of local features of position. Adaptive Feature Clustering (AFC) is then used for performing local-level feature adaptation, and eventually, decoders with Recurrent Neural Network (RNN) structures are used to decode the local features into character sequence.

2.4.1.3 Text Localization using Faster R-CNN with resnet-50/101-feature pyramid network

Deep Convolutional Neural Networks (DCNN) have been applied on computer vision tasks like detection, segmentation and classification of objects [125] [118]. Overfeat [146] runs sliding windows of different scales on an image and classifies a region using a classifier and regress the bounding box coordinates. This method is computationally expensive and time consuming. Region based Convolutional Neural Network or R-CNN [147] incorporated region-based proposal technique for object detection which further lead to a family of networks like Fast R-CNN and Faster R-CNN [147] [126]. R-CNN [147] applies selective search [148] on an image and proposes 2000 regions that are passed through a Convolutional Neural Network (CNN) for feature extraction and evaluated for classification and bounding box regression. Selective search is not a part of the neural network. Graph based image segmentation method initially hypothesize set of regions which are then hierarchically grouped based on similarity measures like color, texture, size and region-filling. Bottleneck of R-CNN [147] is in computing feature maps of each 2000 region proposals. Fast R-CNN [126] compensates the bottleneck of R-CNN [147] by first generating feature map of the image through a CNN and a separate pipeline that does selective search on the image to produce 2000 region proposals that are mapped to the feature map using RoI pooling [126]. Fast R-CNN [126] reduces test time of an image by R-CNN [118] on a magnitude of 10. Faster R-CNN [121] is an advancement on Fast R-CNN [126] which incorporates RPN that replaces selective search pipeline that was offline i.e., not trainable to reduce computation time while testing. A RPN [121] that gives region proposals is a fully connected neural network that does binary classification of whether a region is object or not and hence assigns confidence or objectness score. The regression end of the fully connected layer predicts offset values of the proposals from the ground truth region. Prior to the fully connected layer the RPN also has CNN as it takes feature maps as input [121]. The network in alternatively trained in a manner that CNN layers of RPN and detector network are shared [121]. Detector networks are not bound to just region proposal-based networks. Dense sampling-based networks [149] [150] [151] [136] are one stage detection network that detect objects over dense sampling of possible locations. Text detection on topographic sheet could be carried out by any above network but the choice of network is crucial. Dense sampling-based detector networks can fall short of accuracy but are much faster than region-based proposal detector networks [152]. Huang et al [152] have considered parameters like object size, input image resolution, number of proposals, memory, Graphics Processing Unit (GPU) speed that contribute towards accuracy and speed. The conclusion relevant to us produced by Huang et al [152] is that Faster R-CNN [121] using Inception Resnet [153] as feature extractor with 300 region proposals attained highest mean average precision (mAP). Therefore, for detecting text on topographic sheet Faster R-CNN architecture is adopted.

2.4.2 Text Recognition

Text by virtue is a sequential data where every character contributes to the context. CNNs predict only a single label to an input and also expect the input of a fixed size. Thus, sequential data will require a network that predicts a series of labels over time and also which can accept inputs of variable lengths. DCNN were heavily used to recognize texts, where [154] [155] used the idea of detecting characters in a text and segmenting each to be classified against a DCNN model trained in on characters. Also, another plausible method would be to recognize texts by classifying the entire text image over a DCNN that was implemented by [156]. In [154] [155] [156] implementations would require a strong classifier that would accurately predict a class of a character. The models would likely fail to generalize on classifying texts belonging to vocabulary of huge number of characters. Jaderberg et al [156] network for predicting text for a language would require training the DCNN on an entire set of vocabulary which is computationally expensive as well would suffer accuracy. Therefore, from analysis it was noted that DCNN are not a viable option for text recognition. Classifications tasks on sequential data are commonly done using RNN. A RNN can accommodate dependence between inputs and variability in size of input. In case of image containing text the RNN expects to accept features of the image sequences. The extraction of features for image feature sequenced inputs had been handcrafted using image processing techniques as adopted by [157] [158]. Convolutional Recurrent Neural Networks (CRNN) [159] replaces the offline image processing task of extracting features for image feature sequenced input with CNNs hence makes the entire network end to end trainable.

The CRNN [159] is adopted for doing text recognition on topographic sheet. The three-component architecture with CNN for feature extraction, RNN for predicting labels per frame of input feature sequence and a top transcription layer to unify the prediction labels into a single label. All three layers are well justified for carrying out image-based sequence recognition. The CNN layer encodes representations of a text image. The input size of CNN is fixed; however, the character representations of variable length input image sequence are not lost since representations are encoded depth wise through collection of feature maps hence making it invariant of the fact of fixed input size. The feature sequence is passed through a deep bi-directional RNN i.e., a stack of bidirectional Long Short-Term Memory (LSTM) [160] [161] [162] which stores contexts of past predictions in both directions thereby making it more suitable for image based text recognition. RNNs suffer from vanishing gradient descent problem and cannot store long range contexts hence LSTMs are used [163].

Table 2.3 (page no 73) presented below highlights recent techniques for localization and recognition of text features in an image.

Table 2.3: Recent techniques identifying text in an image

Sl. No	Description	Strategy used for Detection/ Recognition	Inference	Motivation
a)	Zhao Z Q, Zheng P [100]	• An efficient recognition system relies on the accuracy of object detection. Efficient recognition activities include informative region selection, feature extraction, and classification.	Challenging for a robust system in a situation where the quality of the input image varies in terms of appearance, brightness, contrast, sharpness, and varied background.	Localization of objects is essential for an effective recognition system.
b)	Convolutional Neural Networks (CNN) [99] [101] [102] [103] [104] [105]	• CNN's ability relies on its multi-layer perceptron, use of local perception, established connections,	• The contribution of various layers varies, as it has been often seen that lower layers hold in them greater	The results of CNN in object detection and recognition are not only effective but also hold high confidence.

		and variable weights assigned. • Image features are used to prepare CNN for the automatic classification task.	discriminative power than the layers on top. • For better feature representation, training can be done layer-wise. • The fusion of all the layers would help elevate the performance.	
c)	Chauhan et al [106] and Lin Liu C [107]	CNN was adopted for image recognition on the CIFAR-10 and MNIST datasets.	• High accuracy was achieved. Performance can be further improved by increasing the instances in the learning set and the hidden layers.	The CIFAR-10 and MNIST datasets are similar to text features present in topographic sheets.
d)	Redmon J [110] and Jmour [111]	Effective feature extraction is crucial for enhancing the recognition process.	• used the “fine tuning technique," which reuses	There is no standard dataset for topographic text features, so a self annonated

			existing refined layers for classifying images. • The adopted technique highlights the ability to work well with a limited number of training sets.	dataset using data augmention may be used for a transcription model with a limited dataset.
e)	YOLOv1 and YOLOv2 [112]	Draws multiple bounding boxes on the region of interest (ROI), with the probability estimation of each box indicated by mean average precision (mAP)	• YOLOv1 determine a suitable bounding box from many bounding box. • YOLOv2 improves YOLOv1, raising the mAP by using a multiscale learning mechanism.	The best possible bounding box containing text features can be identified, ensuring the presence of text features in the bounding box.
f)	Regions with CNN (R-CNN) [147]	RCNN has the ability to detect multiple objects present in a single image.	• RCNN identifies feature vectors of known length and use this object detection.	CNN may not be suitable for a topographic sheet as

			Model ability depends on efficient training of the network for object detection, with implications on time, cost, and space [119] [120].	many text features are present in the image.
g)	Fast R-CNN [164]	Fast R-CNN uses single-stage training that is capable of detecting object at a very fast rate nearing real time, excluding the time spend on region proposal [127].	The model has higher detection quality (mAP), multi-task loss based single-stage training, rapid update of network layers and non-usage of storage for feature caching.	The fast RCNN is a reliable region proposal technique that uses single-stage training to efficiently determine objects' boundaries.
h)	Faster R-CNN [128] [134]	It uses a two-stage detection approach.	• For the bounding box, an IoU based improved loss function is used, followed by bilinear interpolation for improving the RoI.	• The accurate detection and recognition of text features plays a vital role as these features are mapped to other

			• Faster R-CNN [134] takes a considerable longer time compared to YOLO [149] and Single Shot Multi-Box Detector [136], but Faster R-CNN comparatively performs better.	associated geographical features, so the quality of the detection process is of the essence. • With faster R-CNN [134] that takes considerably longer, but it has a higher mAP value.
i)	Natural Scene Text Detection and Recognition using Faster RCNN based on Cascade RPN [141]	The proposed novel framework is inclusive of a pre-processing step and a faster RCNN based on cascade RPN for detection and recognition of text in natural scenes.	• The commonly used text detection algorithm lacks efficiency for real world natural scenes containing textual information. • Highlighted the use of pre-processing steps like deblurring and sharpening to	• The natural scenes are analogous to topographic sheet, where the text present varies in size, color, type, and orientation, and more often than not, these features intersect or

			reduce noise for attaining better accuracy.	overlap with other features, thus making it a challenging task. • The use of preprocessing steps deblurring and sharpening.
j)	Natural Scene Text Detection using Yolo V5 framework [142]	Optimizing the tradeoff between speed and precision of results to discover text in natural images using the Yolo V5 framework on the ICDAR 2013 dataset.	• Implemented single phase detector, Yolo V5, for detecting horizontal text as well as vertically aligned text. • F-measure, precision, and recall metrics were used to analyse the performance of the model.	to emphasise the tradeoff between speed and precision of results to discover text in natural images with an ability to build a rotation invariant system. To emphasise the tradeoff between speed and precision of results to discover text in natural

			For the ICDAR 2013 dataset, the model was developed in less time with a limited model size.	images with an ability to build a rotation invariant system.
k)	Bilingual Text Detection in Natural Scene [143]	Presented state-of-art technique for detecting bilingual text in natural scenes using Faster RCNN on MSRA-TD500 dataset.	• Challenges highlighted due to different orientations, scaling, brightness of features of interest, and complex backgrounds. • Moreover, model performance degrades if images contain bilingual text.	The use of faster RCNN for extracting probable text regions embedded in complex environments
l)	Natural Scene Text Detection using ABCNet for detetcing normal as	Summarises deep learning based text detection algorithms for natural scenes, namely, region proposal-	• CTPN detects multi-scaled text features. • SegLink can detect text aligned at different angles.	The topographic sheet contains text aligned at different angles, multi-scaled, and curved as well.The topographic sheet

	well as curved text [144]	based and semantic segmentation.	• ABCNet can detect curved text.	contains text aligned at different angles, multi-scaled, and curved as well.
m)	Text Recognition using unsupervised technique [145]	Proposes an unsupervised technique for text detection using DOC.	Highlighted the need for huge amounts of labelled data for modeling, which is an extremely time-consuming and expensive process.	Use of unsupervised techniques where huge amounts of labelled data are not available.

2.4.2 Inferences drawn from the existing literature

Some of the inferences drawn from the study of associated literature are:

a) It has been observed that most of the text localization techniques fails to deliver the desired results with variation in qualities like tone, texture, attenuation, sharpness, and contrast. Orientation and scaling of the text further intensifies the problem complexity.

b) Pre-processing is mandated in order to elevate the accuracy of the obtained results as it helps in effective handling of embedded noises.

c) Text localization can be achieved either using traditional approach relying on morphological understanding (structural formation) or using the conventional approaches such as multi-layered learning methods.

d) The effectiveness of text localization technique depends on its ability to neatly encapsulate the intended text within a bounded space known as bounding box.

e) The model conceived is expected to exhibit higher detection quality, reduced data loss, and subsequently advance the learning ability of the network.

f) The performance of the model can be validated using various measures such as F-measure, precision, and recall metrics.

g) On successful localization of intended text, recognition is performed to identify the information contained in localized text.

h) The quality of the results obtained from recognition techniques used can be advanced if the input image is dealt with in forms of distinct layers as it has been observed that the lower layers contributes to the greater extent in regards to the discriminative powers. Therefore, it is suggested that the training of the intended model is done layer-wise.

i) It has also been suggested that the performance of the model can be enhanced by creating a fusion of layers with relative priority.

j) The overall efficiency of the model can be amplified by increasing the number of instances in the training set as well as the number of hidden layers. It is advised to have a planned trade-off between accuracy and resource investment as such approach would inherently incur greater computation time and resources.

k) The effectiveness of the model also depends on the variation of instances in the training set.

l) The complexity of recognition manifolds with variation in scale, alignment, structural orientation and lingual formats used.

2.4.3 Research motivations

Topographic sheet contain valuable textual information, such as place names, labels, symbols, and annotations, which provide crucial details about the geographic features represented in the map. The following are the motivations for research initiatives:

a) Manual identification and recognition of text features in topographic sheets can be labor-intensive, time-consuming, and prone to human error. Automated techniques for text detection and recognition with increased efficiency and accuracy aid in map analysis and data extraction. Manual identification and recognition of text features in topographic sheets can be labor-intensive, time-consuming, and prone to human error. Automated techniques for text detection and recognition with increased efficiency and accuracy aid in map analysis and data extraction.

b) Text features present in topographic sheets often intersect with other morphological features; in addition, these features are multi-scaled and multi-aligned. An automated system that can mitigate all issues is challenging.

c) Manual updating and maintenance of topographic sheets can be a time-consuming and costly process. Automated text feature detection and recognition aids in the timely maintenance and updating of topographic sheets.

2.5 Generation of a Digital Elevation Model from Refined and Generated Contour Lines

Elevation models are used by various audiences to attain their purpose-specific objectives. Some of the prominent users include researchers and administrators [165] Elevation models are steadily finding their relevance in different domains such as geomorphology, geology, agriculture and horticulture, infrastructural planning, and hydrological studies [166] [167] [168] and most importantly, for estimating slope and aspect. The elevation model is also prominently used in hazard management such as landslides, creating terrain profiles, delineating surface features [169] [170], and determining potential terrain for tapping solar energy [171] [172] [173].

It is essential for elevation models to express sufficient quality information to meet the emerging requirements of various morphological operations [174]. Subsequently, morphological operations should also be effectively designed to enable the generation of spatially processed features for improving the quality of elevation models [175] [176].

The overall process starts with the digitization of the contours from the reference map. It may be performed manually or automatically. Manual digitization is time-consuming and often prone to errors. As per recent trends, there is a paradigm shift towards automatic approaches. Automatic digitization of features from reference maps can be achieved by deploying color-based segmentation processes that use distinctive pixel intensity values used for representing features on the reference map as a basis [177] [178] [179]. Further, it is essential to skeletonize the image for ease of data processing [180].

It has been observed that the results of the segmentation process deployed for contour line extraction often suffer from breakages, resulting in a loss of contour continuity. This may be due to a large variation in the intensity of pixels representing the contours. Further use of such incomplete reference data would lead to the generation of inaccurate 3D models. Such inconsistencies can be avoided by establishing continuity between the broken contour lines with the help of knowledge-based curve drawing methods. Alternatively, a simple straight-line drawing algorithm may also be adapted based on the severity of breakage to ease complexity.

The basic slope-intercept mathematical formula [181] provides a mechanism that iteratively calculates points between two broken points identified by a pair of coordinates. Since there are many pixels between broken points and the basic slope-intercept formula involves iterative complex calculations like division and multiplication, this may incur complexity. Alternatively, the simple approach for connecting broken points is to use the

Digital Difference Analyzer (DDA), which uses an incremental approach for identifying pixels that connect terminal points [182]. Although it is simple and easy to implement, it involves fractional calculation, resulting in the calculated points being in floating point format, further requiring rounding off of the calculated points as coordinate values are discrete. A naive line-drawing algorithm [183] may also be used, but the algorithm is not found to be versatile. A similar efficient technique for drawing a straight line that involves minimal complex computation is Bresehmen’s Line Drawing Algorithm [184]. This algorithm involves only integer calculations, drastically reducing the complexity of the process.

There are many other algorithms like Xiaolin Wu's line [185] and the Gupta-Sproull algorithm [186], which is also an efficient line drawing technique that generates a straight anti-aliased line with some additional computational complexity. The simple Bezier curve [187] drawing technique can draw a straight line as well as a curved line based on the number of control points. A computationally efficient technique that uses Bresehmen’s Line Drawing technique for generating a straight line in an auxiliary direction [74]. The plotting of pixels stops when two lines intersect with each other.

Table 2.4 (page no 86) presented below compares the various line drawing techniques for reconnecting the contour lines based on strategy used, its advantages, and disadvantages.

Table 2.4: Comparison of the various line drawing algorithms

Sl No	Method	Strategy	Advantage	Disadvantage
a)	Basic slope-intercept formula [181]	uses the slope-intercept formula iteratively to calculate points between two pixels.	Simple and easy to implement.	Time-consuming as it involves multiplication and division operations iteratively.
b)	DDA- Digital Difference Analyzer [182]	Draws straight using an incremental approach between terminal points based on multiplication and division operations.	Simple and easy to implement.	• Involves fractional calculation. • A round-off operation is performed. • Accumulated round-off of coordinate locations leads to a shift of calculated coordinate points away from the true line.
c)	Bresehmen's Line Drawing	Draws straight using an incremental approach between terminal points	• Faster execution due to integer calculation.	• Directional information is not preserved as the algorithm works

	Algorithm [184] [188]	based on addition and subtraction operations.	Round-off operations are not required.	on the principle of slope between the current and previous pixels. • Anti-aliasing is not implemented.
d)	A naive line-drawing algorithm [183]	The technique may be used, if x2 > x2 and points are ordered, for drawing a line from (x1, y1) to (x2, y2).	Efficient for a line if the gradient (slope) is less than or equal to one.	• Inefficient for a line with gradient more than 1, generates gaps between pixels.A less efficient technique for a line with a gradient greater than one. It generates gaps between pixels. • May encounter an exception (divide by zero) situation. • Involves floating point number calculation. • Implementing anti-aliasing is extremely time-consuming.

e)	Xiaolin Wu's line algorithm [185]	Draws a line by straddling the coordinate points with an anti-aliasing effect.	Comparatively faster but slower than Bresenham's.	Implements an anti-aliasing effect.
f)	Gupta-Sproull algorithm [186]	Draws anti-aliased lines by calculating the perpendicular distance from the pixel’s centre to the line’s centre.	Generates an anti-aliased line.	• Involves fractional calculation. • Additional round-off operations are required. • Accumulated round-off of coordinate locations leads to a shift of calculated coordinate points away from the true line.
g)	Simplest Bezier curve [187]	Draws a straight line using three control points.	Has the ability to draw lines as well as curves using flexible control points.	Computationally Complex.
h)	Auxiliary direction based	Starting from both terminal points in the	• Fast and simple to implement.	Aliasing effect or stair-case effect

	Straight Line Drawing technique [74]	auxillary direction, apply Bresenhem’s algorithm until they intersect.	• The time taken to draw a line is reduced by half in comparison with Bresenhem’s algorithm.	

Three important components required for the spatial processing of a dataset are knowledge, rules, and models. The prospect of 3D representation of features calls for effective identification of its notable attributes, and clear feature definitions are to be established, including both structural as well as dimensionally represented information with the help of pixel orientations [189]. It is crucial to construct suitable models by establishing topological as well as contextual relations between various aspects of the features. Further, it is also important to pre-specify a set of well-defined constraints for guiding relationships [190].

One of the popular approaches for the generation of 3D information is using photogrammetric tools, but it is often found that these techniques demand more time, greater investment due to the use of sophisticated equipment and increased human intervention [190] [191].

As per recent trends, the Structure from Motion (SfM) technique is gaining tremendous ground in generating a 3D dataset from a series of images and a set of ground control points. This technique is fast, economically viable, and requires relatively less effort and expertise. As a consequence, SfM is being deployed in various applications. As in the case of photogrammetry, SfM deploys a comprehensive matching process wherein image textures present in different photographs are taken into consideration. SfM is affected by homogeneity, image resolution, and geometry. SfM allows for the generation of topological images with a certain degree of quality and resolution for images of variable size [192] [193] [194] [195] [196].

Both mathematical modeling and spatial approaches, as well as hybrid approaches, may be pursued for generating interpolated points. Spatial approaches mandate the deployment of appropriate mechanisms for extracting the contour segment coordinates and information regarding their morphological orientations. It is also further stated that the generation of contours and maintaining continuity are difficult [197]. Upon successful generation of features, the reference maps may be updated for improved visual representations [198] [199] [200] [201] [202].

Subsequently, elevation models may be generated using Linear prediction, wherein covariance based on distance is used to depict spatial correlations between neighboring points [203] [204] [205]. Further, image analysis-based spatial processing tools relying on contour lines, triangular irregular networks, gridded surfaces, and parametric surfaces can be deployed for topographic modeling of surfaces, such as generating 3D representations from 2D data [206] [207]. Masataka et al articulated that the generation of elevation models from small scale spatial data proves very effective. The same was achieved by deploying a buffering-based approach for generating intermediate contours in-between existing contours. The elevation detail was determined using curve fitting with regards to the profile [208].

Two crucial elements for converting contour maps into DEM are the sampling density of the contours and the quality of the same, which may be elevated by incorporating contour lines generated through interpolation techniques based on a linear or cubical relationship between data points [209], dilation of contour lines [210], structural orientation [211], profile, proportional distance, direction, the weighted moving average combined with geometric relationships [212] and regionalized comprehension [213] [214]. This implies that the quality of the elevation model greatly depends on the capability of the interpolation technique [215] [216]. Hermite splines can be combined with the inverse distance weighting technique to generate intermediate contours in between existing contours [217] [218].

2.5.1 Inferences drawn from the existing literature

Some of the inferences drawn from the study of associated literature are:

a) Elevation model for an area of interest can be generated either using satellite imaginary or by using contour lines specified in the TS.

b) The elevation model derived from satellite imaginary is subjected to relative positioning of pixels based on its intensity i.e., DEM based on satellite imaginary is relative in nature whereas elevation model derived from contours are based on absolute distance.

c) Based on the difference of representation it is always preferred to construct elevation model from contour lines.

d) Generation of elevation model from contour lines mandates the following inherent steps:

    - Digitization of contours Preprocessing for refinement
    - Reconnection for ensuring contour continuity
    - Generation of intermediate contours for enhancing the quality of elevation model
    - Association of elevation values with the contour lines

- Representation of contour lines in 3D space for visualizing elevation model

e) An intensity-based color segmentation is to be deployed for digitizing the contours.

f) Digitized contours may explicitly incorporate uncalled for features. These features termed as noise is to be suitably eradicated using preprocessing tools

g) Due to lack of effective digitization and preprocessing operation, some of the significant content may be lost resulting in broken contour lines

h) The broken contour lines are to be connected using purpose specific reconnection tools for ensuring continuity

i) It is often observed that when the scale of representation is high the quality of elevation model tends to degrade. Therefore, in such situation intermediate contour lines may be generated based on knowledge of the adjacent contours.

j) Thereafter, elevation values is associated or assigned with each contour to facilitate projection in the 3D space.

k) Here it is important to ensure that the data structure should be aptly designed to meet the requirement of the processes.

l) Subsequently, the contours are projected in the 3D space to create a visualization of the elevation model.

2.5.2 Research motivations

A DEM is a 3D representation of the Earth's surface, that aids in detailed analysis, visualization, modelling, and decision-making processes related to terrain, water flow, infrastructure, and natural resources. The following are the motivations for this research initiative:

a) DEMs transform contour lines and elevation points from a topographic sheet into a DEM, making it easier to understand and interpret terrain characteristics.

b) The information about area between any two successive contour is missing leading to inadequate information for analysis and shape-preserving generation of contour is essential to address the issue.

c) DEMs generated with the aid of sophisticated software are either manual or semi-automatic and involve human intervention, making them time consuming and relying on the experience of the digitizer.

d) A common method of generating DEM is using satellite images that consider relative mechanisms to represent earth features. In contrast, generating DEM using topographic sheets is more interesting because it represents earth features more accurately.

e) DEMs have wide applications in GIS, which include geospatial analysis, terrain visualization, 3D modelling and simulation, flood modelling and risk assessment, infrastructure planning and engineering, natural resource management, etc.

Chapter 3

Extraction of River Pattern, its Refinement and Attribute Generation

3.1 Introduction

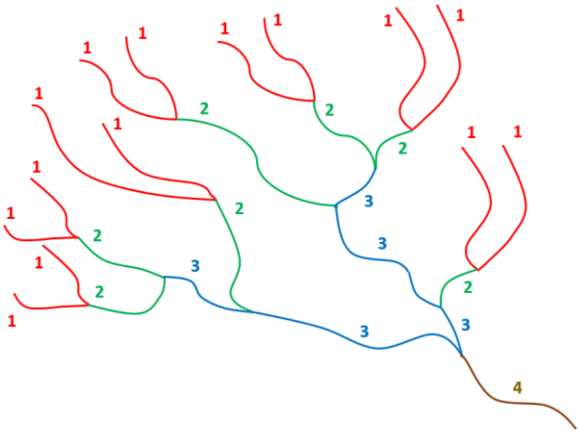
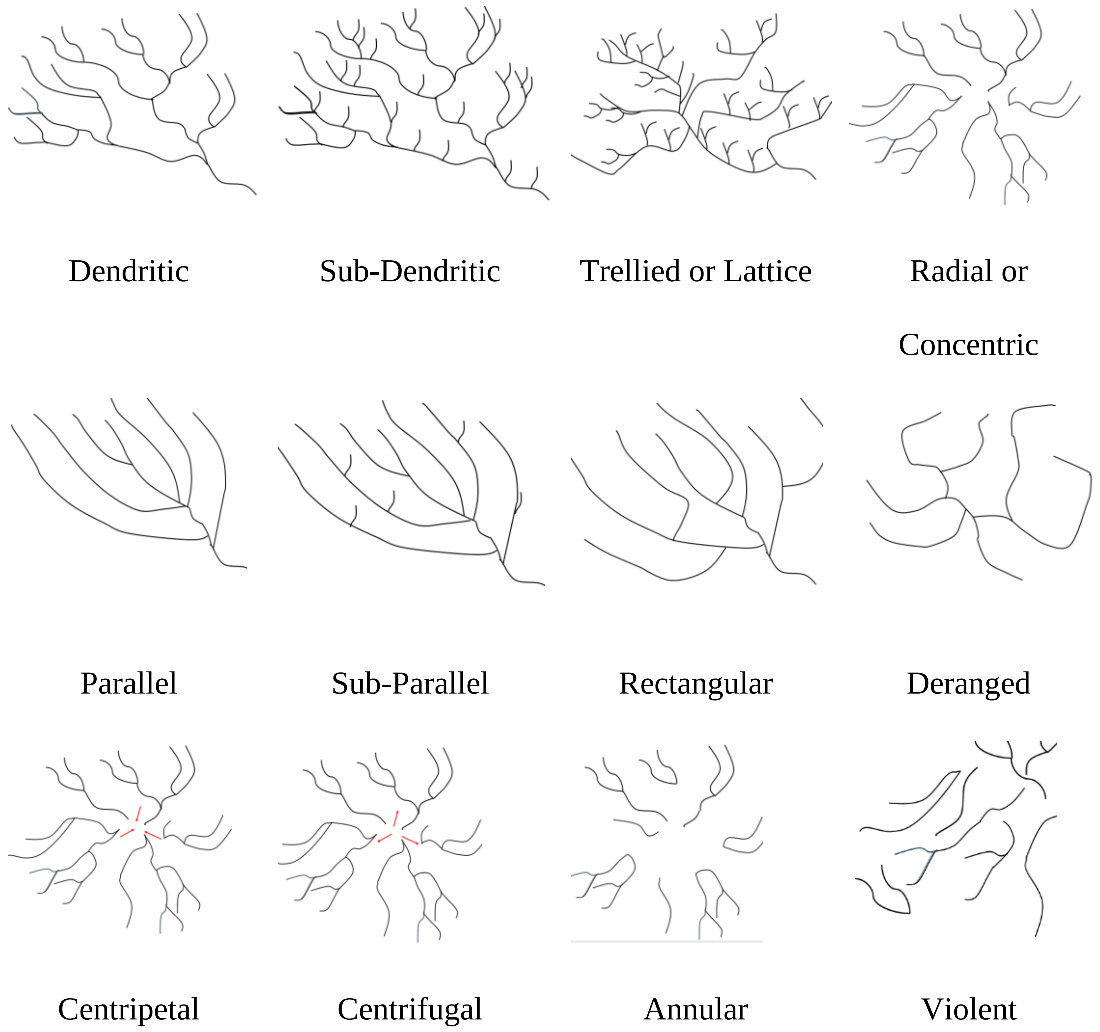


Figure 3.1: Stream Ordering

River networks, as represented in figure 3.1 (page no 95), an integral part of a watershed, have been instrumental in shaping the very essence of human civilization since time immemorial. It is a complex collection of streams, ranging from the smallest of the tributaries to the largest of the streams. Mathematically, a river network may be perceived as an inverted hierarchical tree, with the edges as the river segments, peripheral sources, and confluence as nodes wherein peripheral sources have an in-degree of 0 and confluence has an in-degree greater than 1. In a river network, the water flows are categorised into brooks, creeks, headstreams, medium streams, and rivers.

The orientation or pattern of the river network may be either dendritic, sub-dendritic, trellied or lattice, radial or concentric, parallel, sub-parallel, rectangular, deranged, centripetal, centrifugal, annular, or violent, largely dictated by the geological and geomorphological characteristics as represented in figure 3.2 (page no 96) [60] [219].



Dendritic

Sub-Dendritic

Trellied or Lattice

Radial or Concentric

Parallel

Sub-Parallel

Rectangular

Deranged

Centripetal

Centrifugal

Annular

Violent

Figure 3.2: Different types of river patterns

3.1.1 Understanding Stream Order (S_ω) and various River Ordering Techniques

In hydrological studies, river segments are categorized based on a positive numeric value known as stream order [70]. Ranging from 1 to 12, stream order is an increasing numerical value assigned to the streams in the river network; it is a mechanism for associating relative priorities with the various river segments, i.e., it describes the relative position of a stream in the hierarchy of streams in a river network. Streams with order values ranging from 1 to 3 are referred to as the "headstream." These are fast, steep, and generally located in the peripherals. Streams with order values ranging from 4 to 6 are referred to as "medium streams" and are slow, less steep, and generally located towards the interiors of the watershed [220]. Whereas a stream with order values ranging greater than 7 is referred to as a river, as represented in figure 3.1 (page no 95). Here, it is also to be understood that river segments of the same order are expected to behave differently based on their topographical and physiographical orientations [10]. This implies that river order is more about structural classification [221] and relative sizing of streams and provides very little insight regarding the inherent geomorphological processes.

The headstreams feed the medium streams and the rivers. With the progression from lower-order streams to higher-order streams, the river network gains in size, volume, and strength. Stream order plays a crucial role in studies relating to flora and fauna, planning development initiatives along waterways, and more importantly, in understanding, managing, and containing streams of different magnitudes.

Some of the popular approaches proposed for ordering streams are Classic Stream Order (also known as Hack's or Gravelius' Stream Order), Horton Stream Order, Strahler Stream Order, Shreve Stream Order, Scheidegger Stream Order, and Order by Path Length.

There are two notable ways of classifying these ordering techniques which are based on the traversal strategy adopted and the number of passes required for ordering the segments in the network.

The ordering methods may deploy bottom-up traversal, top-down traversal, or both for identifying the streams in the network. Further, based on the number of passes required these methods may be classified into one pass or two-pass.

The proposed research initiative is motivated by understanding the underlying principles of the river ordering techniques mentioned below and conceiving an automated computational program for the realization of the same.

a) Classic Stream Order

b) Strahler Stream Order

c) Horton Stream Order

d) Shreve Stream Order

e) Scheidegger Stream Order

f) Order by Path Length

g) Consistent Stream Order

h) Cumulative Stream Order

The above-mentioned river ordering techniques have been explained in detail with the help of explanations, algorithms, figures, and results in serials 3.4 (f) i) to 3.4 (f) viii).

3.1.2 Determination of various attributes associated with river segment

The proposed research initiative is also motivated by understanding the significance of the various attributes associated with river segments, as mentioned below, and conceiving an automated computational program for the attainment of the same.

a) Stream Order

b) Stream Number

c) Bifurcation Ratio

d) Streams participating in Bifurcation Ratio

e) Weighted Mean Bifurcation Ratio

f) Stream Length

g) Mean Stream Length

h) Streams participating in Length Ratio

i) Weighted Mean Length Ratio

j) Length of Main Channel

3.1.2.1 Stream Number (N_ω)

The stream number, as represented in equation 3.1, is the number of instances of a segment in a particular order. As per Horton's "law of stream numbers," the instances of segments with a given order geometrically decrease with stream order [222].

As expressed by Horton,

$$\mathbf{N_{\omega}/ N_{\omega+1} = R_b \text{ Or } N_\omega = R_b^{\Omega-\omega} \ldots\ldots\ldots\ldots\ldots\ldots \text{(Equation 3.1)}}$$

where,

R_b is the bifurcation ratio

N_ω is the instances of segment with order ω

Ω is the order of the mainstream

3.1.2.2 Bifurcation Ratio (R_b)

The bifurcation ratio [223], as represented in equation 3.2, is the ratio between instances of segments of a given order and instances of segments of the immediate next higher order. Quantitatively, the bifurcation ratio is expressed as,

$$\mathbf{R_b=N_\omega /N_{\omega+1} \ldots\ldots\ldots\ldots\ldots\ldots \text{(Equation 3.2)}}$$

where, R_b is the bifurcation ratio

N_ω is the instances of segment with order ω

$N_{\omega+1}$ is the instances of segment with order ω +1

As per Horton, the bifurcation ratio will have values in between 2 and 3 or 4 depending upon the type of landscape of the basin, ranging from plane to steep and dissected landscapes of the river basin. Strahler suggested that the bifurcation ratio is subjected to less variation, with a mean around 3 to 3.5.

3.1.2.3 Number of streams participating in Bifurcation Ratio (N_{Rb})

As represented in equation 3.3 it is equivalent to the total number of streams participating in a bifurcation ratio.

$$\mathbf{N_{Rb} = N_\omega + N_{\omega+1}} \text{ ………………. } \textbf{(Equation 3.3)}$$

3.1.2.4 Weighted Mean Bifurcation Ratio (R_{bwmean})

As per Strahler, the weighted mean bifurcation ratio can be determined by dividing the sum of the products of the bifurcation ratio and the total number of segments involved in the ratio by the total number of segments involved in the ratio, as represented in equation 3.4.

As expressed by Strahler,

$$\mathbf{R_{bwmean}} = \left(\sum_{\mathbf{i=0}}^{\mathbf{m}} \mathbf{R_{bi}} * \mathbf{n_{R_{bi}}}\right) / \left(\sum_{\mathbf{i=0}}^{\mathbf{m}} \mathbf{n_{R_{bi}}}\right) \text{ ………………. } \textbf{(Equation 3.4)}$$

where,

R_{bwmean} is weighted mean bifurcation ratio

R_{bi} is the i^{th} instance of bifurcation ratio

n_{Rbi} is the total number of segments involved in the ratio R_{bi}

3.1.2.5 Stream Length (L_ω)

Stream length, as represented in equations 3.5 and 3.6, can be determined through spatial processing of the river network and adding the length of segments of a particular order.

$$\mathbf{L_\omega = \left(\sum_{i=0}^{m} n_{i\omega}\right) * k} \text{ ……………… } \textbf{(Equation 3.5)}$$

(in specified unit as per scale where 'k' is the resolution)

or

$$\mathbf{L_\omega = \left(\sum_{i=0}^{m} n_{i\omega}\right)} \;\; \textit{(in pixel)} \text{ ……………… } \textbf{(Equation 3.6)}$$

where,

L_ω is the total length of segments with order ω

$n_{i\omega}$ is the pixel count in all i segments with order ω

k is the resolution of the reference image

3.1.2.6 Mean Stream Length ($L_{\omega mean}$)

The mean stream length can be determined by dividing stream length (L_ω) by the number of instances of segments of a particular order, as represented in equation 3.7.

$$\mathbf{L_{\omega mean} = L_\omega / N_\omega} \text{ ……………… } \textbf{(Equation 3.7)}$$

where,

L_ω is the total length of segments with order ω

$L_{\omega mean}$ is the mean length of segments with order ω

N_ω is the instances of segments with order ω

3.1.2.7 Stream Length Ratio (R_L)

As per the Horton “law of stream lengths," the mean length of a segment ($L_{\omega mean}$) with a given order geometrically increases with stream order [50], as represented in equation 3.8.

As expressed by Horton,

$$\mathbf{R_L = L_{\omega mean/}\ L_{\omega mean-1}\ Or\ L_{\omega mean} = L_1 R_L^{(\omega-1)}}\ \ldots\ldots\ldots\ldots\ldots\ldots\ \textbf{(Equation 3.8)}$$

where,

R_L is the length ratio

$L_{\omega mean}$ is the mean length of segment with order ω

L_1 is the mean length of segment with order 1

3.1.2.8 Number of streams participating in Bifurcation Ratio (N_{Rb})

As represented in equation 3.9 it is equivalent to the total number of streams participating in a length ratio.

$$\mathbf{N_{Rb} = N_{\omega} + N_{\omega -1}}\ldots\ldots\ldots\ldots\ldots\ldots\ \textbf{(Equation 3.9)}$$

3.1.2.9 Weighted Mean Length Ratio (R_{Lwmean})

The weighted mean length ratio can be determined by dividing the sum of the products of the length ratio and the total number of segments involved in the ratio by the total number of segments involved in the ratio, as represented in equation 3.10.

As expressed by Stralher,

$$\mathbf{R_{Lwmean} = (\sum_{i=0}^{m} R_{Li} * n_{R_{Li}}) / (\sum_{i=0}^{m} n_{R_{Li}})}\ \ \ldots\ldots\ldots\ldots\ldots\ldots\textbf{(Equation 3.10)}$$

where,

R_{Lwmean} is weighted mean length ratio

R_{Li} is the ith instance of length ratio

n_{RLi} is the total number of segments involved in the ratio R_{Li}

3.1.2.10 Area Ratio (R_A)

As per the Schumm "law of stream areas", the drainage basin area (A_ω) geometrically increases with stream order [50], as represented in equation 3.11.

As expressed by Schumm,

$$\mathbf{R_A = A_\omega / A_{\omega-1} \text{ Or } A_\omega = A_1 R_A^{(\omega-1)} \ldots\ldots\ldots\ldots\ldots\ldots \text{ (Equation 3.11)}}$$

where,

R_A is the area ratio

A_ω is the mean area of segment with order ω

A_1 is the mean area of segment with order 1

3.1.2.11 Length of Main Channel (Cl)

The length of the main channel, as represented in equations 3.12 and 3.13, can be determined through spatial processing of the river network by adding the length of all the segments belonging to the mainstream and eventually multiplying the same by the resolution of the reference image.

$$\mathbf{L_{ms} = (\textstyle\sum_{i=0}^{m} n_{ims}) * k \ldots\ldots\ldots\ldots\ldots\ldots \text{ (Equation 3.12)}}$$

(in specified unit as per scale where 'k' is the resolution)

Or

$$\mathbf{L_{ms} = (\textstyle\sum_{i=0}^{m} n_{ims}) \ldots\ldots\ldots\ldots\ldots\ldots \text{ (Equation 3.13)}}$$

(in pixel)

where,

L_{ms} is the length of the mainstream.

n_{ims} is the pixel count in all i segments with order ms

k is the resolution of the reference image

3.1.3 Selection of appropriate data structure for maintaining values of the attributes

The effectiveness of the automated computational process also depends on the efficient organisation of data values in the data structure; rather than relying on traditional linear organization, a nonlinear jagged representation would serve as a better alternative.

3.2 Objectives of the proposed work

The proposed work is motivated by the development of automated computational programs for effortless digitization of river networks, their ordering, and the generation of associated attributes, which may be of great assistance for inferential studies related to the understanding of river networks.

The proposed work also attempts to understand and implement eight river ordering techniques. A comparative assessment is also planned to discuss in detail the ordering technique, inherent approach, number of passes required, computation cost, advantage, disadvantage, and extent of human intervention required.

With the need for adherence to the achievement of the above-mentioned motivations, the objectives of the proposed research initiatives are therefore conceived as:

a) Deploy an appropriate digitization technique for extracting the river network from the reference map.

b) Refine the extracted river network by efficiently removing noise using a suitable filter.

c) Skeletonize the digitized feature by applying an appropriate morphological operator.

d) Conceive and deploy an effective data traversal technique for determining the terminal streams.

e) Understand the working principles of various river ordering techniques.

f) Apply various river ordering techniques to the skeletonized river network to identify streams of different orders.

g) Generate various attributes associated with the river network.

h) Store the attributes in an efficient data structure for facilitating easy analysis.

3.3 Applications of the proposed research initiative

River order can be set as a scientific basis for:

a) Describing the size of a particular river network and the gradient of the landscape

b) Drawing the hydrological characteristics of the streams classification of river segments into headwater streams, medium-sized streams, and large rivers

c) Determining various stream attributes such as stream length, stream count, and many more

d) Describing the stream ecosystem (including nutrients, habitat, organic materials, and the energy cycle) and its variation with the physical gradient

e) Determination of non-point-source pollution problems arising due to natural or human activity

3.4 Proposed Methodology

The proposed research initiative is motivated towards conceiving an efficient generic computational program capable of sequentially performing the stated activities for addressing the research gaps identified during the review of related works:

a) Color segment the river network from the topographic sheet

b) Pre-process the segmented image to eradicate noise

c) Skeletonize the pre-processed segmented image to create features with single-pixel width

d) Resolve m-connectivity to facilitate efficient processing

e) Determine the terminal streams using an efficient spiral traversal mechanism

f) Identify all the stream segments by identifying all the confluences in the river network

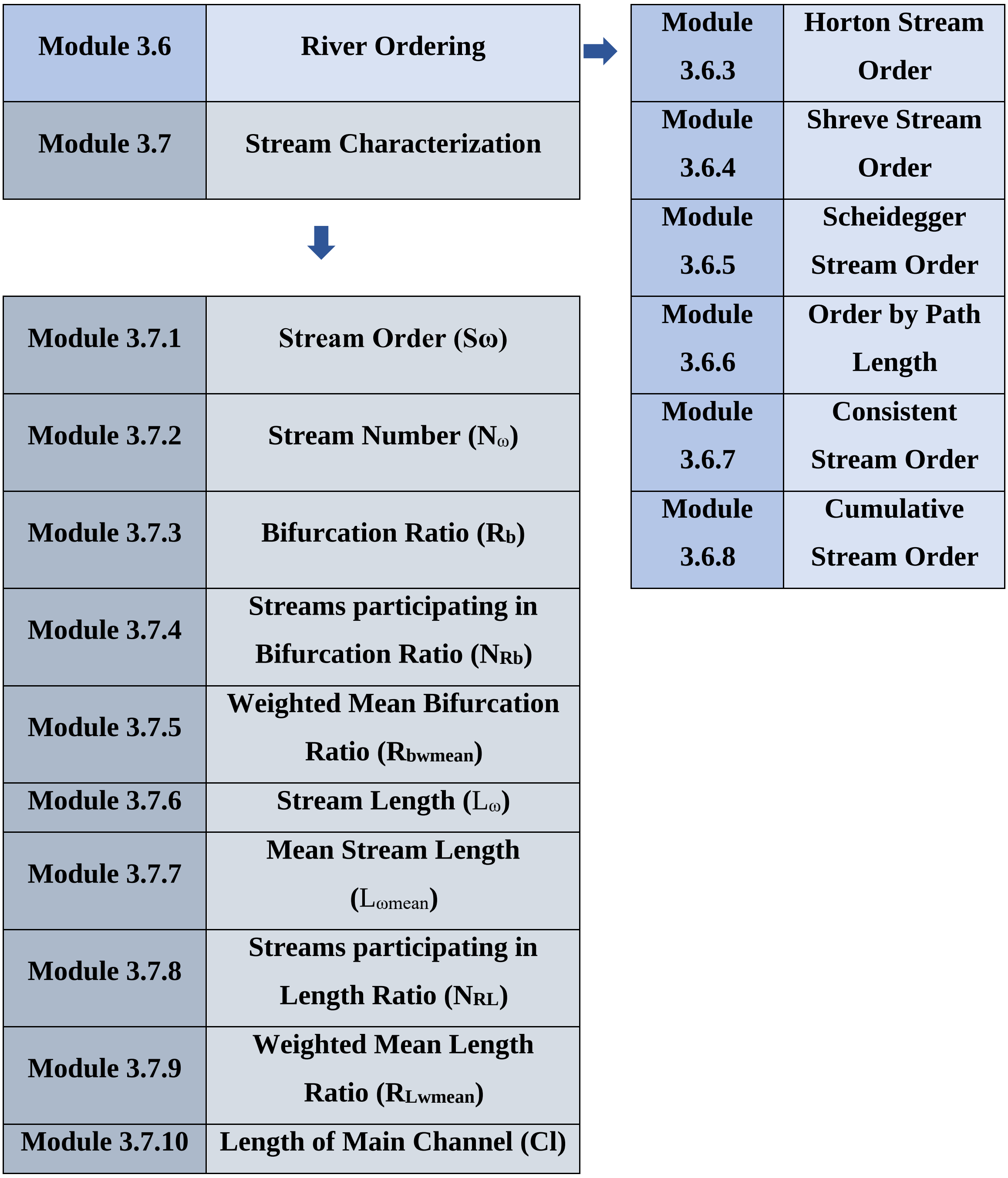
g) Order the identified streams as per the principles of Classic, Strahler, Horton, Shreve, Scheidegger, Order by Path Length, Consistent and Cumulative ordering techniques

h) Associate attributes with river network such as Stream Order, Stream Number, Bifurcation Ratio, Streams participating in Bifurcation Ratio, Weighted Mean Bifurcation Ratio, Stream Length, Mean Stream Length, Streams participating in Length Ratio, Weighted Mean Length Ratio and Length of Main Channel based on the principles of the ordering techniques

The proposed solution framework for this research initiative has in it seven identifiable modules as represented in figure 3.3 (page no 107):

Module 3.1	**River Digitization**
Module 3.2	**Noise Removal**
Module 3.3	**Thinning**
Module 3.4	**Resolve Common Neighbors**
Module 3.5	**Spiral Traversal**

Module 3.6.1	**Classic Stream Order**
Module 3.6.2	**Strahler Stream Order**



Module 3.6	River Ordering
Module 3.7	Stream Characterization

Module 3.6.3	Horton Stream Order
Module 3.6.4	Shreve Stream Order
Module 3.6.5	Scheidegger Stream Order
Module 3.6.6	Order by Path Length
Module 3.6.7	Consistent Stream Order
Module 3.6.8	Cumulative Stream Order

Module 3.7.1	Stream Order (S_ω)
Module 3.7.2	Stream Number (N_ω)
Module 3.7.3	Bifurcation Ratio (R_b)
Module 3.7.4	Streams participating in Bifurcation Ratio (N_{Rb})
Module 3.7.5	Weighted Mean Bifurcation Ratio (R_{bwmean})
Module 3.7.6	Stream Length (L_ω)
Module 3.7.7	Mean Stream Length ($L_{\omega mean}$)
Module 3.7.8	Streams participating in Length Ratio (N_{RL})
Module 3.7.9	Weighted Mean Length Ratio (R_{Lwmean})
Module 3.7.10	Length of Main Channel (Cl)

Figure 3.3: Proposed framework for the research initiative

a) River Digitization

Digitization of a feature of interest may be executed through a traditional manual approach, such as tracing. Manual digitization is extremely time-consuming and expensive, as well as does not ensure quality; moreover, confidence in the results greatly depends on the level of experience of the digitizer [224] [225].

In order to overcome the inherent problems associated with the traditional manual approach, an intensity-based automated computational program may be conceived and realised in order to perform segmentation of the feature of interest from the reference map.

In contrast to the manual approach, automated digitization is fast and efficient, but its effectiveness greatly relies on the ability to correctly appropriate the range of intensity values representing a feature of interest. It may be subjected to the error of commission (inclusion of non-contextual information) or the error of omission (exclusion of contextual information) if the range of intensity values is not suitably fixed. Therefore, digitization is a task critical activity and demands great attention [75] [226].

Rivers are explicitly represented on a reference map with the help of a distinctive color code for ease of visual interpretation. An intensity-based image segmentation module needs to be suitably scripted to digitise the same into a distinctive layer. This process needs to be carefully executed, as the overall capability and confidence of other processes depend on it. The adopted strategy is presented in algorithm 3.1.

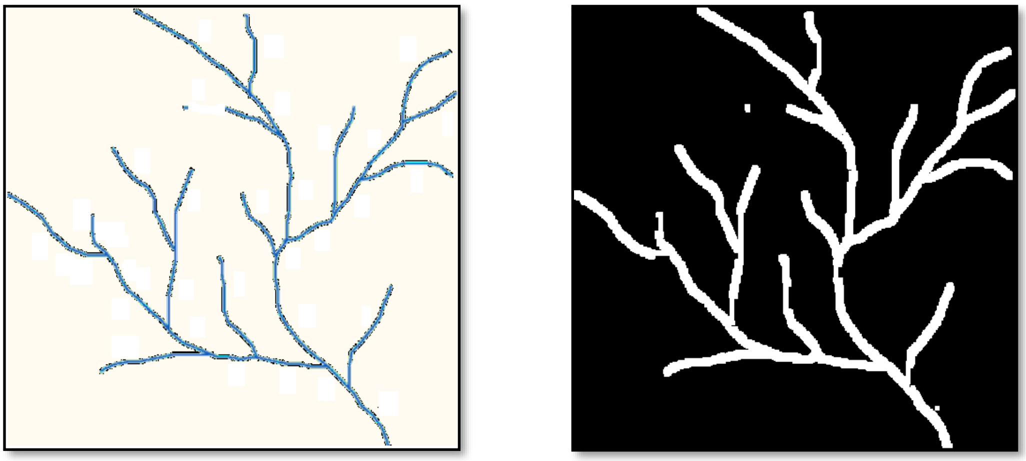
Algorithm 3.1: Algo_ Digitize_Network

```
Consider  img        input image
          row, col   row size and column size
          river      output image
          temp       processing variable
```

```
                 $val_1$ $val_2$   bound for threshold representing
                                   river
Start            temp:=            rgb to gray (img)
                 row,col:=         size(temp)
                                   for i =1 to row do
                                       for j=1 to col do
                                           if temp (i,j) $\geq val_1$ && temp (i,j) $\leq val_2$
                                               river(i,j)=1;
                                           else
                                               river(i,j)=0;
                                           End if
                                       End for
End                                End for
```

Algorithm 3.1: Algorithm for Digitizing River Network

Figure 3.4 a) (page no 110) above represents a sample of river networks taken from the topographic sheet, and figure 3.4 b) (page no 110) represents the result of the digitization process where the river network is digitized into a distinctive vector layer.



(a) (b)

Figure 3.4: Sample input river network and digitized river network

b) Noise Removal

The inclusion of non-contextual information is inevitable during the automatic digitization of features. It may be due to the close resemblance of the non-contextual information with the feature of interest in terms of its intensity value. Such non-contextual information is called "noise." Salt and pepper noises are the most suitable examples [227] incorporated into the dataset as a consequence of improper segmentation, which further has to be removed using purposive spatial processing.

It is extremely crucial to weed out noise from the digitized data, as it would significantly hamper the results of the inferential study. Error-specific spatial filters of appropriate size may be conceived and deployed for the removal of the same, such as a median filter.

Very often, it has been observed that segmented images, in addition to significant datasets, also express noises (e.g., salt and pepper). This can be achieved by using standard, well-accepted filtering techniques or by designing a simple contextual morphological module. Here, a median filter of 3x3 is used for removing the salt and pepper noise, as detailed in algorithm 3.2.

Algorithm 3.2: Algo_ Remove_Noise

```
Consider  river           Dataset
          row, col        row size and column size
          wind            processing variable
          win_w           window width
          win_h           window height
          i,j,k,l,ex,ey   Variables
----------------------------------------------------------
Start          row,col:=  size(river)
                          ex:= floor(win_w/2)
                          ey:= floor(win_h/2)
                            for i:=ex to row-ex do
                                for j:=ey to col-ey  do
                                    k: = 0
                                    for l:=0 to 3 do
                                        for m:= 0 to 3 do
                                            wind[k] := riv[i + l - ex][j + m - ey]
                                            k:= i + 1
                                        End for
                                    End for
                                    sort entries in wind[]
```

```
                                   river[i][j] := wind[wind_w * wind_h / 2]
End                            End for
                          End for
```

Algorithm 3.2: Algorithm for Noise Removal

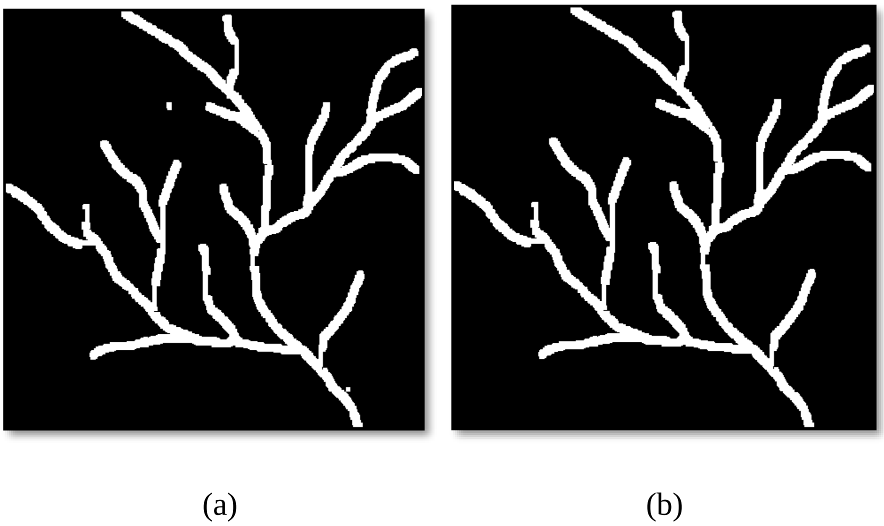

(a) (b)

Figure 3.5: Digitized river network and pre-processed river network

Figure 3.5 a) (page no 112) above represents digitized river networks that need to be filtered for the removal of noise or errors caused by improper digitization, and figure 3.5 b) (page no 112) represents the result of the preprocessing performed using the median filter.

c) Thinning

One of the most effective strategies to reduce the extent of computation required during the assessment of digitized features is to create a single pixel-width integrity-preserving representation of the feature through the application of morphological operators [228] [229] [76]. The creation of such a skeletonized representation of morphological features eases quick assessment as well as facilitates the generation of values for various attributes associated with the feature.

Morphological processing can be further simplified by condensing the dimensional information in the digitized image. In this initiative, Zhang et al.'s thinning algorithm [76] is deployed to create a skeletonized representation of the river network.

It is a two-pass algorithm that selects the pixel to be deleted while maintaining the structural appearance of an object and converting the same into a one-bit-per-pixel image. Assume black pixels are one and white pixels are zero, and the input image is a rectangular n by m array of ones and zeros.

The neighbors are arranged as:

$$\begin{array}{ccc} \mathbf{P_9} & \mathbf{P_2} & \mathbf{P_3} \\ \mathbf{P_8} & \mathbf{P} & \mathbf{P_4} \\ \mathbf{P_7} & \mathbf{P_6} & \mathbf{P_5} \end{array}$$

Define A(P) = the number of transitions from white to black in the sequence P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9, P_2.

Define B(P) = the number of black neighbors of pixels. = sum (P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9).

Pass 1: The pixel is black and has eight neighbors.

(0) $2 \leq B(P) \leq 6$

(1) A(P)=1

(2) Atleast one of P_2, P_4 and P_6 is white.

(3) Atleast one of P_4, P_6 and P_8 is white.

Pass 2: All pixels are again tested

(0) The pixel is black and has eight neighbors.

(1) $2 \leq B(P) \leq 6$

(2) $2 \leq B(P) \leq 6$

(3) Atleast one of P_2, P_4 and P_8 is white.

(4) Atleast one of P_2, P_6 and P_8 is white.

These steps are repeated until no image pixels are changed as detailed in algorithm 3.3.

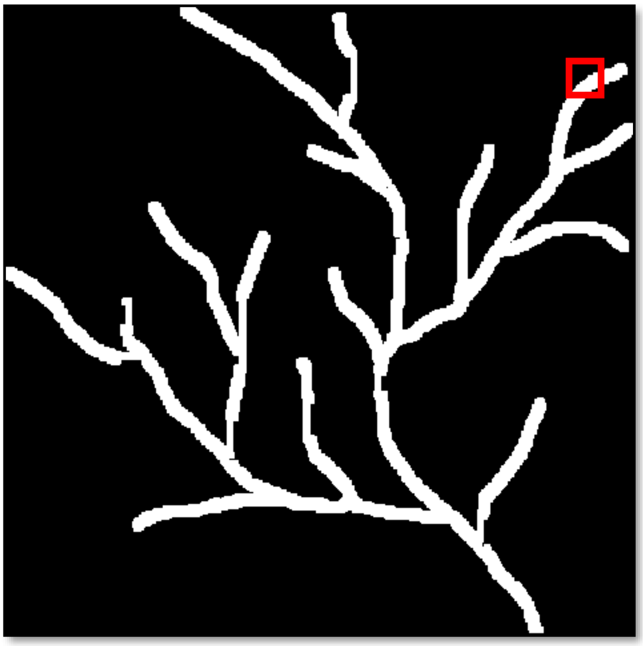
Algorithm 3.3: Algo_ Thinning

```
Consider   River       Dataset
           row, col    row size, column size
           i,j         Variables

Start        row,col:=   size(river)
                 for i=2 to row-1 do
                     for j= 2 to col-1 do
                         check for conditions stated for pass 1 to determine whether pixel
                         becomes a candidate for removal
                     End for
                 End for
                 for i=2 to row-1 do
                     for j= 2 to col-1 do
```

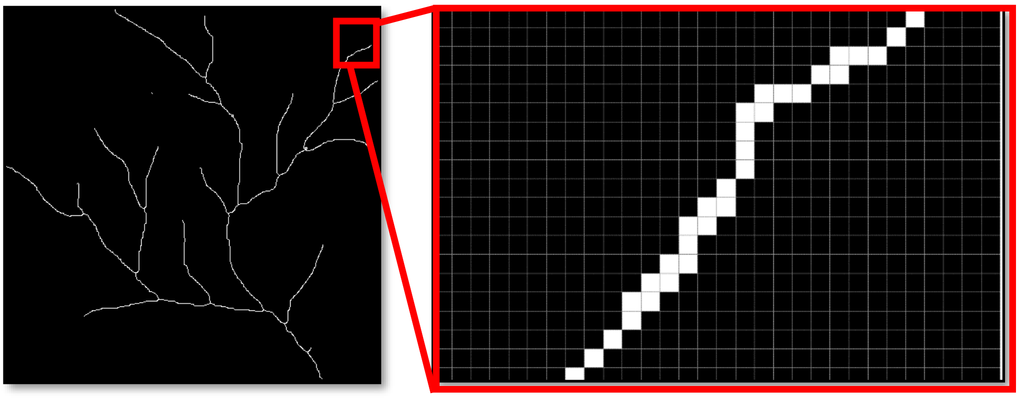
```
                    check for conditions stated for pass 2 to determine
               whether pixel
                    becomes a candidate for deletion
End                 End for
               End for
               repeat until no new pixel is marked for deletion.
```

Algorithm 3.3: Algorithm for Skeletonizing/ Thinning

Figure 3.6 a) (page no 116) above represents pre-processed river networks; figure 3.6 b) (page no 116) represents the result of the thinning processes; and figure 3.6 c) (page no 116) represents a demonstration of the thinning process.



(a)



(b) (c)

Figure 3.6: Digitized river network, thinned river network and demonstration of thinning process

d) Resolve Common Neighbors

On close observation of the digitized river network, it was found that there exist instances of structural orientation wherein certain pixels have common neighbors; although the presence of such situations would not to an extent influence the result, for ease of operation, the proposed research initiative simplifies the existence of all such instances.

For simplicity, for any significant pixel, if the number of neighbours varies from 3 to 8, then it has to be simplified. For any significant pixel with two neighbours P and Q, the m-Connectivity is defined as:

$$\mathbf{N_4(P) \cap N_4(Q) \neq NULL \ \ldots\ldots\ldots\ldots\ldots\ldots \ (Equation\ 3.14)}$$

where,

$N_4(P)$ represents four neighborhood of significant pixel P (top, down, left, and right pixel)

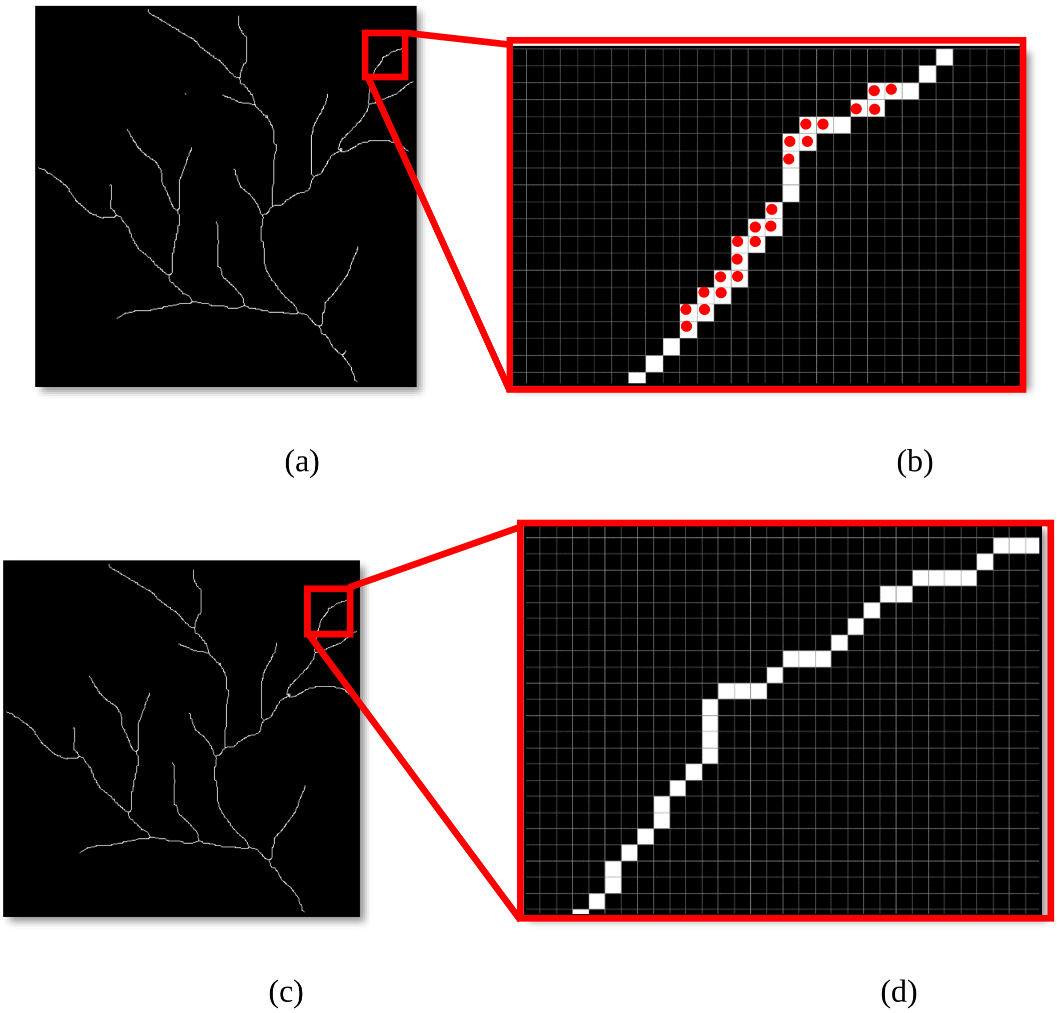
$N_4(Q)$ represents four neighborhood of significant pixel Q (top, down, left, and right pixel) i.e., there should not be any common N_4 elements between pixel P and Q.

The sequence of steps for resolving common neighbor is explained in algorithm 3.4 specified below.

Algorithm 3.4: Algo_ Resolve_Neighbor

```
Consider  river      Dataset
          row, col   row size and column size
          tmp        window of size 3x3

Start     for i=2 to i_row-1 do
              for  j=2 to col-1 do
          for any significant pixel with two neighbors P and Q

                      ensure  $N_4(P) \cap N_4(Q) \neq$ NULL  where
          $N_4(P)$ & $N_4(Q)$
                      represents four neighborhood of significant
          pixel
End                   End for
              End for
          End for
          save and display contour
```

Algorithm 3.4: Algorithm for Resolving Common Neighbor



(a) (b)

(c) (d)

Figure 3.7: Thinned river network with instances of m-connectivity, example of m-connectivity and demonstration of resolution of m connectivity

Figure 3.7 a) (page no 118) above represents a thinned river network with instances of m-connectivity and figure 3.7 b) (page no 118) explicitly reflects the presence of m-connectivity marked with red spots. Figure 3.7 c) (page no 118) represents a demonstration of the resolution process for removing m-connectivity and figure 3.7 d) (page no 118) explicitly reflects the removal of m-connectivity.

e) Spiral Traversal

On close observation of the different patterns of the river network, it has been found that the peripheral streams are radially oriented, as seen in figure 3.2 (page no 96), i.e., the peripheral streams in the river network radially converge to form steams of higher orders, as seen in figure 3.4 a (page no 110).

The overall effectiveness of the various stream ordering techniques largely depends on the ability of the data traversal technique deployed for identifying the peripheral streams. The traditional row major or column major traversal mechanism is found to be inefficient in achieving the same [69] [230]. Therefore, an alternative spiral traversal mechanism is proposed for attaining the same.

An adaptation of the spiral traversal proposed by Mohan et al [69] has been deployed here for determining the peripheral segments in the river network, as represented in figure 3.8 (page no 119). In this situation, spiral traversal proves more effective than the traditional row-column approach for identifying the peripheral segments, as it was observed that the peripheral stream is located towards the four peripherals of the dataset. The workings of the spiral traversal technique are explained in algorithm 3.5, which is specified below.

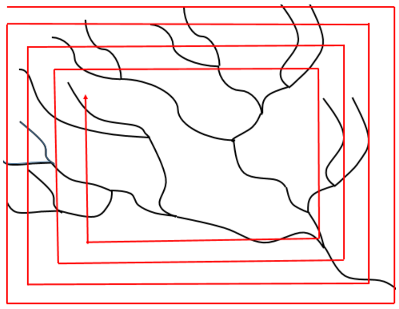


Figure 3.8: Spiral Traversal

Algorithm 3.5: Algo_Spiral_Traversal

Consider	river	Dataset
	row, col	row size and column size
	i,j	Variables

```
Start    row, col= size(river);
         calculate centroid:
         x_c=row/2;y_c=col/2;
         for travel from 1, 1 to x_c, y_c
             if x_c & y_c is reached
                 break;
             else
                 travel right-> travel down->travel left->
         travel up
                 on encountering a significant pixel stop for
                 performing desired operation
End          End if
         End for
```

Algorithm 3.5: Algorithm for Spiral Traversal

f) Stream Ordering

The proposed research initiative implements the following eight ordering techniques:

i) Classic Stream Order (Hack's Stream Order or Gravelius' Stream Order)

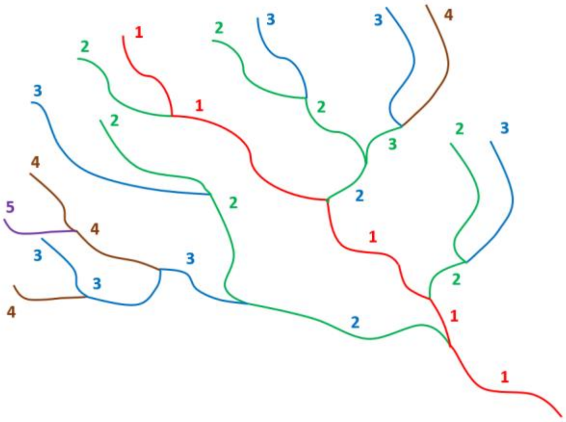


Figure 3.9: Classical Stream Order

As proposed by John Tilton Hack [14], these techniques start with the mainstream and then traverse backwards to associate orders with the river segments. It starts by assigning 1 to the mainstream, and on encountering a confluence while traversing backwards, either of the tributaries will continue with the order of the mainstream, whereas the other tributary will continue with order 2 (one plus the order of the mainstream). The process recursively repeats till all the segments are assigned an order number, as shown in figure 3.9 (page no 120). The working principle of the Classic Stream Ordering technique is explained in algorithm 3.6, which is specified below.

Algorithm 3.6: Algo_Classic Stream Order

```
Consider   f      function for stream order
           m      number of stream segments
           i      the main stream with order '1'
                  f(i)=1
           j, k   the tributaries to stream i

Start            for all 'm' streams while traversing backward do
                      if there exist 'j' & 'k'
                            f(j):= f(i); f(k):= f(i)+1;
                                        or
                            f(k):= f(i); f(j):= f(i)+1;
                 End if
End              End For
```

Algorithm 3.6: Classic Stream Order (Hack's Stream Order or Gravelius' Stream Order)

ii) Strahler Stream Order

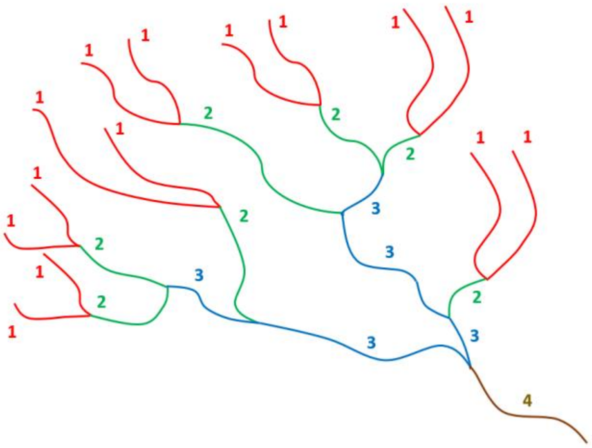


Figure 3.10: Strahler Stream Order

As proposed by Arthur Newell Strahler [231], this technique starts with the peripheral streams and then traverses forwards to associate orders with the river segments. It starts by assigning 1 to the peripheral stream, and on encountering a confluence, either of these actions may be initiated.

a) The resulting stream retains the highest order of the converging tributaries in a situation where the orders are not the same.

Or

b) The resulting stream is assigned an order that is equal to 1 plus the order of either of the tributaries in a situation where the orders of the converging tributaries are the same. The process recursively repeats till all the segments are assigned an order number, as shown in figure 3.10 (page no 122). The working principle of the Strahler Stream Ordering technique is explained in algorithm 3.7, which is specified below.

Algorithm 3.7: Algo_ Strahler Stream Order

Consider	F	function for stream order
	M	number of stream segments
	I	the resulting stream

j, k   the tributaries to i

```
Start   for all 'm' streams while traversing forward do
            if there exist 'j' & 'k'
                if $(f(k) \neq f(j))$
                    $f(i) := \max\{f(k), f(j)\}$;
                else
                    $f(i) := f(k)+1$ or $f(i) := f(j)+1$ ;
                End if
            else
                $f(i) := 1$;
            End if
End     End for
```

Algorithm 3.7: Strahler Stream Order

iii) Horton Stream Order

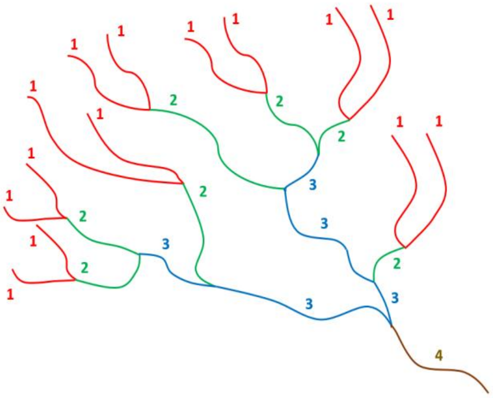


Figure 3.11 a): Strahler Stream Order 1st Pass

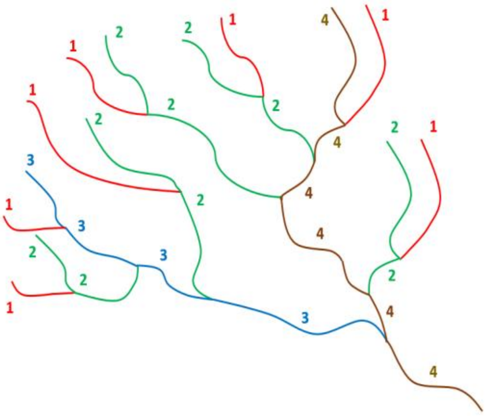


Figure 3.11 b): Horton Stream Order 2nd Pass

As proposed by Robert E. Horton, this technique assigns the same order for a river segment stretching from the source to the mainstream [231] [62]. The ordering takes place over two passes. In the first pass, the ordering of the river segments is done using the Strahler Stream Order.

On detecting the mainstream, the second pass starts. The Horton Ordering technique assigns the highest order to the mainstream, and on encountering a confluence while traversing backwards, either of the tributaries will continue with the order of the mainstream, whereas the other tributary will retain its Strahler Order. The process recursively repeats till the mainstream is tracked to its source, as represented in figures 3.11 a) and b) (page no 123).

Here, to avoid the selection of a false source at the confluence, either of the following criteria are adhered to:

a) Assign the order of the mainstream to the tributary with the highest Strahler in a situation where there are streams with different Strahler Order.

Or

b) Assign the order of the mainstream to the tributary with the highest flow rate or largest catchment area. This mandates the presence of an accumulation map for deriving quality decisions.

The working principle of the Horton Stream Ordering technique is explained in algorithm 3.8, which is specified below.

Algorithm 3.8: Algo_ Horton Stream Order

```
Consider  F    function for stream order
          M    number of stream segments
```

I the resulting stream

j, k the tributaries to i

```
Start   for all 'm' streams while traversing forward do
            if there exist 'j' & 'k'
                if ($f(k) \neq f(j)$)
                    $f(i) := \max\{f(k), f(j)\}$;
                else
                    $f(i) := f(k)+1$ or $f(i) := f(j)+1$ ;
                End if
            else
                $f(i) := 1$;
            End if
        End for
        for all 'm' streams while traversing backward do
            if there exist 'j' & 'k'
                if ($f(k) \neq f(j)$)
                    if($f(k) > f(j)$)
                        $f(k) := f(i)$;
                    else
                        $f(j) := f(i)$;
                    End if
                else
                    assess the flow rate or largest catchment of j & k
                    $f(k) := f(i)$; or $f(j) := f(i)$;
```

```
                        End if
End           End for
```

Algorithm 3.8: Horton Stream Order

iv) Shreve Stream Order

As proposed by Ronald L. Shreve [17] [18], this technique starts with the peripheral streams and then traverses forwards for associating orders with the other network segments. It starts by assigning 1 to the peripheral stream, and on encountering a confluence, it assigns the order that is equivalent to the sum of the orders of all the tributaries to the resulting segment. The process recursively repeats till all the segments are assigned an order number, as shown in figure 3.12 (page no 126). The working principle of the Shreve Stream Ordering technique is explained in algorithm 3.9, which is specified below.

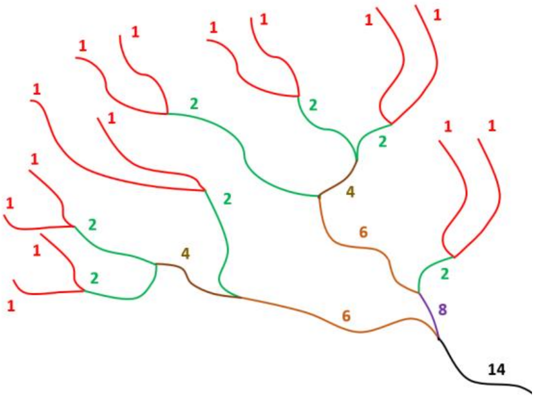


Figure 3.12: Shreve Stream Order

Algorithm 3.9: Algo_ Shreve Stream Order

```
Consider  f      function for stream order
          m      number of stream segments
          I      the resulting stream
          j, k   the tributaries to i

Start          for all 'm' streams while traversing forward do
                    if there exist 'j' & 'k'
                         f(i):= f(k) + f(j);
                    else
                          f(i):=1;
                    End if
End             End for
```

Algorithm 3.9: Shreve Stream Order

v) Scheidegger Stream Order

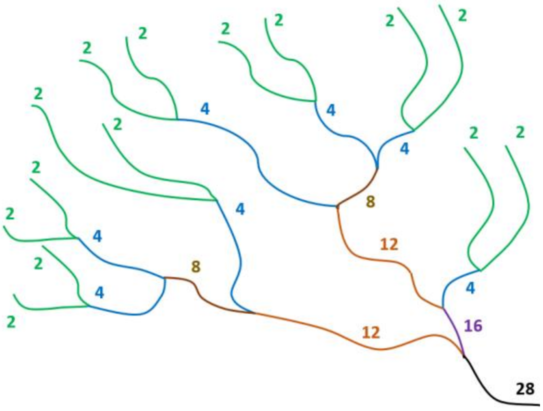


Figure 3.13: Scheidegger Stream Order

As proposed by Adrian E. Scheidegger [232] [19], this technique starts with the peripheral streams and then traverses forwards for associating orders with the other network segments. It starts by assigning 2 to the peripheral stream, and on encountering a confluence, it assigns the order that is equivalent to the sum of the orders of all the tributaries to the resulting segment. The process recursively repeats until all the segments are assigned an order number, as shown in figure 3.13 (page no 127). The working principle of the Scheidegger Stream Ordering technique is explained in algorithm 3.10, which is specified below.

Algorithm 3.10: Algo_ Scheidegger Stream Order

```
Consider  f      function for stream order
          M      number of stream segments
          I      the resulting stream
          j, k   the tributaries to i
```

```
Start       for all 'm' streams while traversing forward do
                if there exist 'j' & 'k'
                    f(i):= f(k) + f(j);
                else
                    f(i):=2;
                End if
End         End for
```

Algorithm 3.10: Scheidegger Stream Order

vi) Order by Path Length method

Also known as the topological ordering method [22] this method starts by assigning 1 to the mainstream, and on encountering a confluence while traversing backwards, it assigns the order that is equivalent to 1 plus the order of the mainstream to all the tributaries. The process recursively repeats until all the segments are assigned an order number, as shown in figure 3.14 (page no 128). The working principle of the Order by Path Length Method Stream Ordering technique is explained in algorithm 3.11, which is specified below.

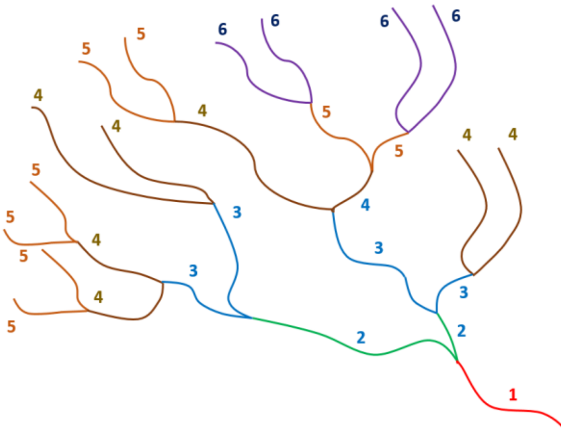


Figure 3.14: Order by Path Length Method

Algorithm 3.11: Algo_ Order by Path Length Method Stream Order

```
Consider    f       function for stream order
```

m number of stream segments

I the main stream with order ‘1’

F(i)=1

j, k the tributaries to i

```
Start    for all ‘m’ streams while traversing backward do
             if there exist ‘j’ & ‘k’
                 f(k):= i+1;  f(j):=i+1 ;
             End if
End      End for
```

Algorithm 3.11: Order by Path Length Method

vii) Consistent Stream Order

As proposed by Adrian E. Scheidegger in [232] and [19], this technique starts with the peripheral streams and then traverses forward to associate orders with the other network segments. Let Z be the resulting stream from the confluence of x and y. If n and m are the orders of x and y, respectively, then the order of stream z is calculated as:

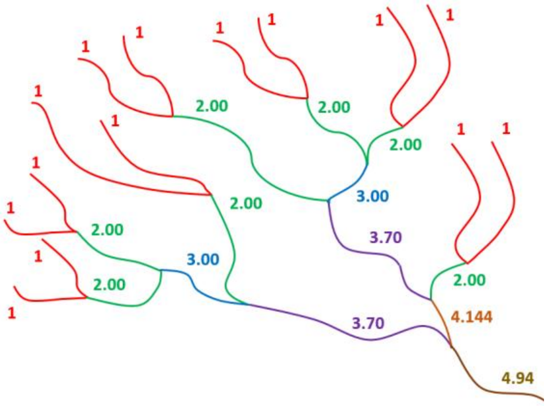


Figure 3.15: Consistent Stream Order

$$\mathbf{f(z)} = \frac{\mathbf{Log(2^m+2^n)}}{\mathbf{Log\ 2}} \text{ ………………. } \mathbf{(Equation\ 3.15)}$$

The process recursively repeats till all the segments are assigned an order number, as shown in figure 3.15 (page no 129). The working principle of the Consistent Stream Ordering technique is explained in algorithm 3.12, which is specified below.

Algorithm 3.12: Algo_ Consistent Stream Order

```
Consider  f      function for stream order
          m      number of stream segments
          i      the resulting stream
          x, y   the tributaries to z
Start            for all ‘m’ streams while traversing forward do
                     if there exist ‘x’ & ‘y’ with order ‘m’ & ‘n’ respectively
                         f(z) = $\frac{Log(2^m+2^n)}{Log\ 2}$
                     End if
End              End for
```

Algorithm 3.12: Consistent Stream Order

viii) Cumulative Stream Order

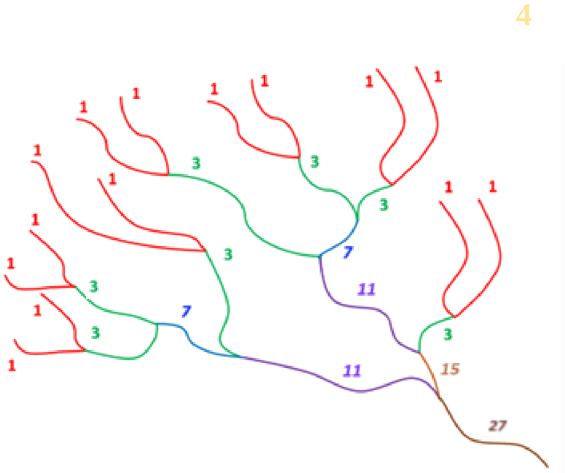


Figure 3.16: Cumulative Stream Order

As proposed by William L. Graf [71], this technique starts with the peripheral streams and then traverses forward to associate orders with the other network segments. Let Z be the resulting stream from the confluence of x and y. If n and m are the orders of x and y, respectively, then the order of stream z is calculated as

$$\mathbf{f(z) = n + m + 1} \text{ .... } \textbf{(Equation 3.16)}$$

The process recursively repeats till all the segments are assigned an order number, as shown in figure 3.16 (page no 130). The working principle of the Cumulative Stream Ordering technique is explained in algorithm 3.13 specified below.

Algorithm 3.13: Algo_ Cumulative Stream Order

```
Consider  f      function for stream order
          m      number of stream segments
          i      the resulting stream
          x, y   the tributaries to z

Start     for all 'm' streams while traversing forward do
              if there exist 'x' & 'y' with order 'm' & 'n' respectively
                  f(z) = $m + n + 1$
              End if
End       End for
```

Algorithm 3.13: Cumulative Stream Order

3.5 Considerations, Constraints, Development Environment

3.5.1 Considerations

To implement the various river ordering techniques, a randomly selected sample image of size 471 x 471 is selected and set as input. The sample input image type is BMP (bitmap).

In pursuit of attaining the objectives of the proposed research initiative, we implemented the following modules without the aid of predefined routines supported by the development platform:

a) Digitization of river network
b) Skeletonization of river network
c) Noise Removal
d) Ensuring m-connectivity
e) Spiral Traversal for detection of peripheral streams
f) Identification of intermediate-streams and main-stream
g) Assignment of Stream Order using various stream ordering techniques
h) Creation of stream attributes

3.5.2 Constraints

The following are the constraints on implementing the research initiatives:

a) Contour map derived from topographic sheet is considered an input sample image.
b) The feature of interest should hold the properties of connected components.
c) The sample image must be a binarized image, free from noise (particularly salt-and-pepper noise).
d) The pixel width of the feature of interest in the sample image is considered to be one and ensures m-connectivity between any two pixels in the neighborhood.

3.5.3 Development Environment

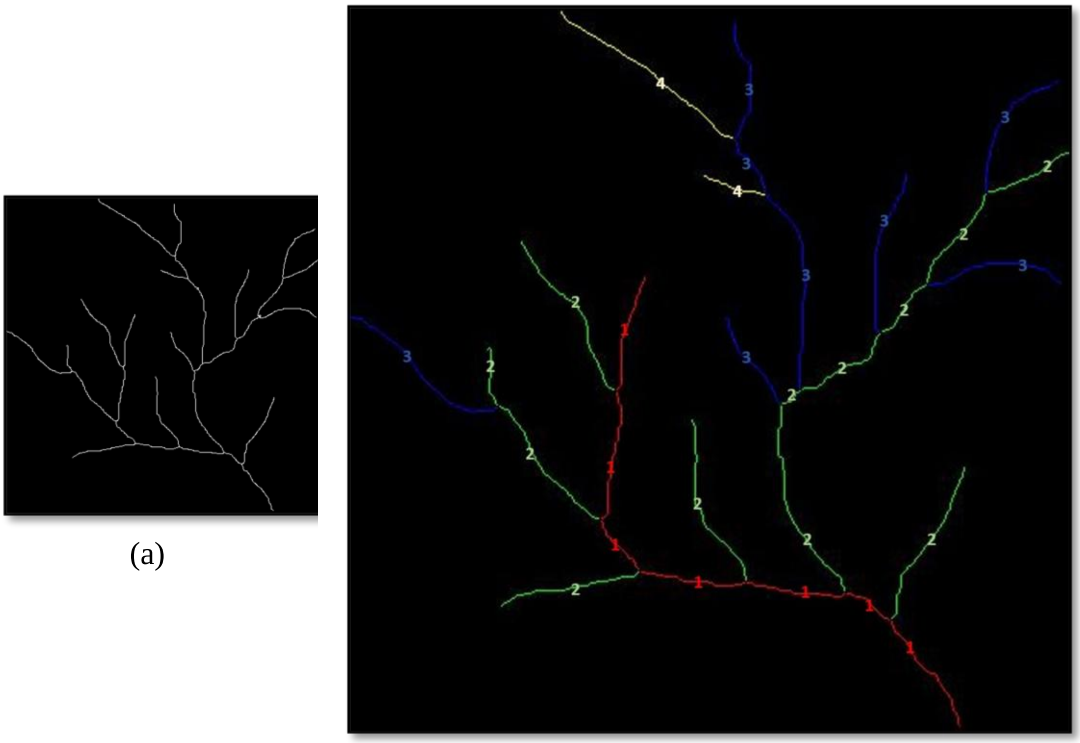
The description of the development environment for the proposed research initiative is as detailed below,

Experimental Configuration	Description	Criteria for Selection
• Processor	11th Gen Intel(R) Core i7-1165G7 @ 2.80GHz	• Facilitates faster execution of programming code
• RAM	16.0 GB (15.8 GB usable)	• Sustain applications requirements
• Operating System	Windows, 64-bit	• User friendly interface • Compatibility with application
• Development Software	MATLAB (2020a)	• Easy translation of concepts to executables • Rich with in-build libraries • Debugging ease • Scalable

3.6 Results and Discussions

3.6.1 Classic Stream Order (Hack's Stream Order or Gravelius' Stream Order)

Figure 3.17 a) (page no 134) represents the input to the Classic Stream Order algorithm. Figure 3.17 b) (page no 134) represents output from the Classic Stream Order. Tables 3.1 a) and b) (page no 135) represent the various attributes generated by the algorithm.



(a)

(b)

Figure 3.17: Input to and output from the automated process for Classic Stream Order

Although easy to understand, simple to use, and intuitive, one of the major limitations of this approach is the need for effective decision support capability at each confluence to determine the true source of the river, i.e., to decide which of the tributaries converging at the confluence would carry forward the order of the mainstream. The tributary that is placed upstream of the bifurcation generally continues with the order of the mainstream, and the other is assigned with the order one more than the order of the mainstream. This technique requires a forward recursive procedure for determining the actual order of the mainstream, which will definitely incur additional computational overhead.

Table 3.1 a): River Network ordering using Classic Stream Order algorithm

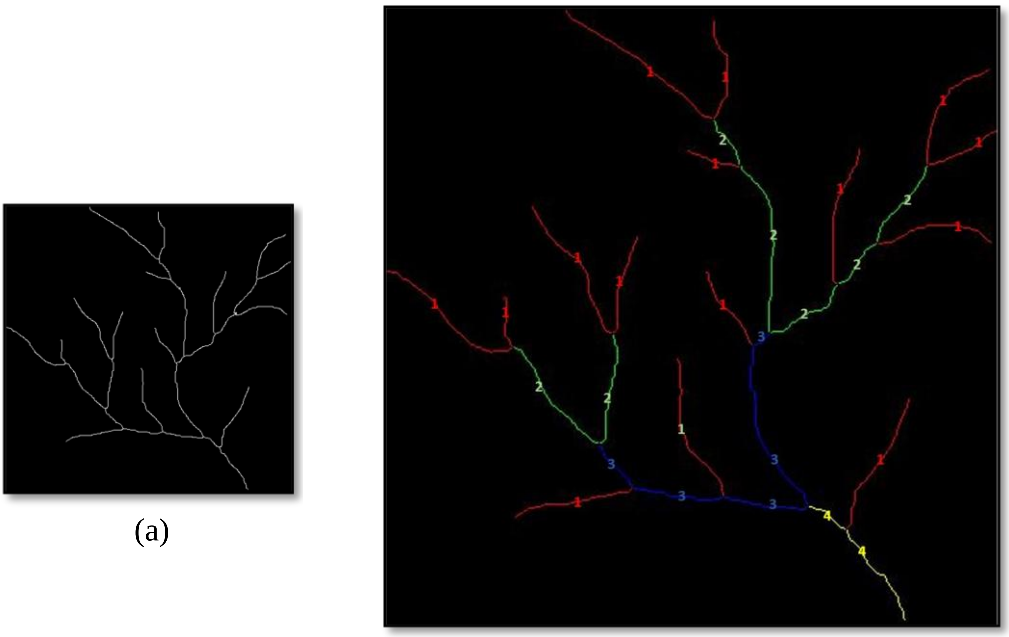
Sl	St Id	Order (S_ω)	Coordinate				Len	Sl	St Id	Order (S_ω)					Len
			s_x	s_y	e_x	e_y					s_x	s_y	e_x	e_y	
1	8	1	175	193	249	174	76	16	21	2	179	377	211	347	39
2	18	1	249	174	333	164	85	17	23	2	211	347	248	295	62
3	22	1	333	164	366	190	37	18	25	2	248	295	258	282	15
4	24	1	366	190	373	259	69	19	27	2	258	282	380	324	125
5	26	1	373	259	380	324	65	20	2	3	10	252	85	251	78
6	28	1	380	324	398	353	30	21	4	3	47	462	120	416	90
7	29	1	398	353	468	398	75	22	5	3	201	2	259	98	104
8	3	2	94	470	120	416	56	23	9	3	179	464	179	377	89
9	6	2	222	91	259	98	41	24	12	3	108	363	211	347	106
10	7	2	152	113	249	174	107	25	14	3	202	246	258	282	61
11	10	2	299	401	398	353	102	26	16	3	85	251	120	272	38
12	11	2	389	100	366	190	92	27	20	3	120	272	248	295	130
13	15	2	268	224	373	259	109	28	1	4	2	139	85	251	120
14	17	2	120	416	179	377	62	29	13	4	109	232	120	272	42
15	19	2	259	98	333	164	87								

Table 3.1 b): River Network characterization using Classic Stream Order algorithm

Sl	S_ω	N_ω	R_b	N_{Rb}	$R_b * N_{Rb}$	R_{bwmean}	L_ω	$L_{\omega mean}$	R_L	N_{RL}	$R_L * N_{RL}$	R_{Lmean}	Cl
1	1	7	-	-	-	-	437	62.43	-	-	-	4250.18	
2	2	12	0.58	19	11.08		897	74.75	1.20	1334	1597.29		
3	3	8	1.5	20	30		696	87	1.16	1593	1854.06		
4	4	2	4	10	40		162	81	0.93	858	798.83		
Total		**29**	**6.08**	**49**	**81.08**		**2192**	**305.18**	**3.29**	**3785**	**4250.18**		**437**
Mean		**7.25**	**2.03**			**1.65**			**1.10**			**1.12**	

3.6.2 Strahler Stream Order

Figure 3.18 a) (page no 136) represents the sample input to the Strahler Stream Order algorithm. Figure 3.18 b) (page no 136) represents output from the Strahler Stream Order. Tables 3.2 a) and b) (page no 137) represent the various attributes generated by the algorithm.



(a)

(b)

Figure 3.18: Input to and output from the automated process for Strahler Stream Order

One of the advantages of this approach is that it has a sound mathematical basis, and the traversal criteria are well-defined, making tracking of the true source relatively easy compared to that of the classic order technique. Here, all the streams are directed from the peripheral to the mainstream. This technique is relatively easy to understand and computationally simple. One of the major limitations of this technique is that, if there are streams with different orders arriving at a confluence, it takes into account the influence of only the highest order stream for deciding on the order of the resulting stream, while it ignores the lower order streams and also does not distinguish the main stream from other streams.

Table 3.2 a): River Network ordering using Strahler Stream Order algorithm

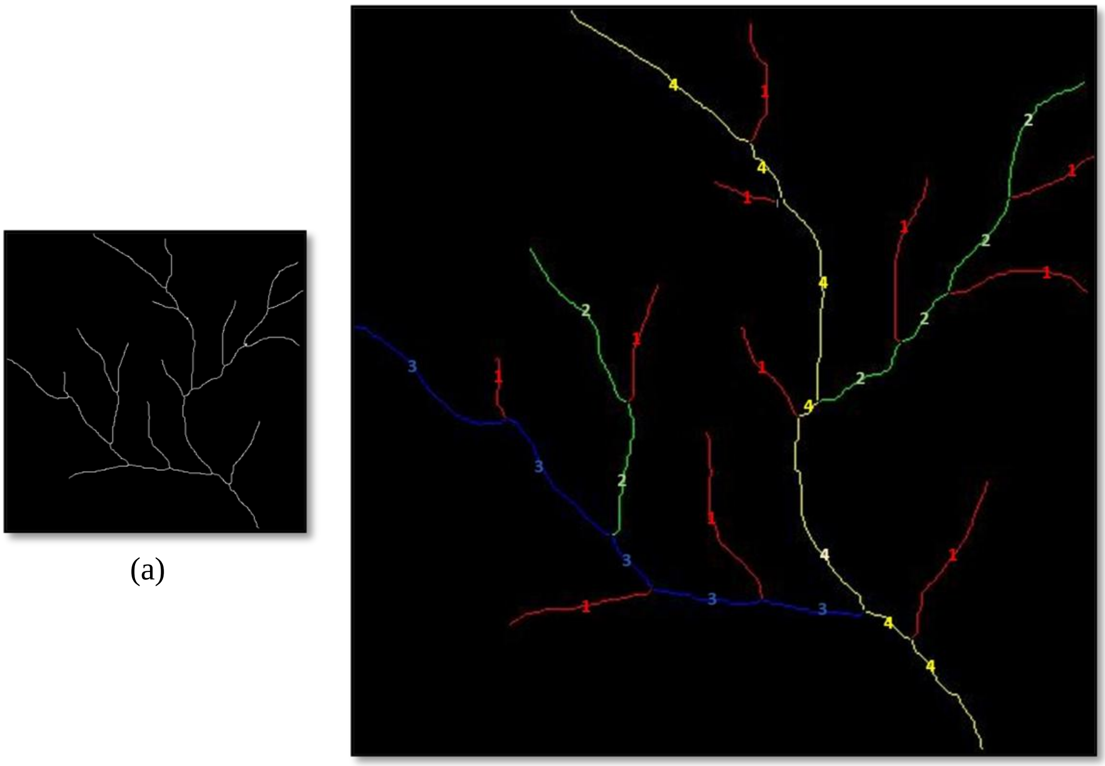
Sl	St Id	Order (S_ω)	Coordinate				Len	Sl	St Id	Order (S_ω)	Coordinate				Len
			s_x	s_y	e_x	e_y					s_x	s_y	e_x	e_y	
1	1	1	2	139	85	251	120	16	16	2	85	251	120	272	38
2	2	1	10	252	85	251	78	17	17	2	120	416	179	377	62
3	3	1	94	470	120	416	56	18	18	2	249	174	333	164	85
4	4	1	47	462	120	416	90	19	19	2	259	98	333	164	87
5	5	1	201	2	259	98	104	20	20	2	120	272	248	295	130
6	6	1	222	91	259	98	41	21	21	2	179	377	211	347	39
7	7	1	152	113	249	174	107	22	23	2	211	347	248	295	62
8	8	1	175	193	249	174	76	23	22	3	333	164	366	190	37
9	9	1	179	464	179	377	89	24	24	3	366	190	373	259	69
10	10	1	299	401	398	353	102	25	25	3	248	295	258	282	15
11	11	1	389	100	366	190	92	26	26	3	373	259	380	324	65
12	12	1	108	363	211	347	106	27	27	3	258	282	380	324	125
13	13	1	109	232	120	272	42	28	28	4	380	324	398	353	30
14	14	1	202	246	258	282	61	29	29	4	398	353	468	398	75
15	15	1	268	224	373	259	109								

Table 3.2 b): River Network characterization using Strahler Stream Order algorithm

Sl	S_ω	N_ω	R_b	N_{Rb}	$R_b * N_{Rb}$	R_{bwmean}	L_ω	$L_{\omega mean}$	R_L	N_{RL}	$R_L * N_{RL}$	R_{Lmean}	Cl
1	1	15	-	-	-	-	1273	84.87	-	-	-		
2	2	7	2.14	22	47.08		503	71.86	0.85	1776	1503.75		
3	3	5	1.40	12	16.8		311	62.20	0.87	814	704.60		
4	4	2	2.50	7	17.5			52.50	0.84	416	351.13		
Total		**29**	**6.04**	**41**	**81.38**		**2192**	**271.42**	**2.56**	**3006**	**2559.48**		**105**
Mean			**2.01**			**1.98**			**0.85**			**0.85**	

3.6.3 Horton Stream Order

Figure 3.19 a) (page no 138) represents the sample input to the Horton Stream Order algorithm. Figure 3.19 b) (page no 138) represents output from the Horton Stream Order. Tables 3.3 a) and b) (page no 140) represent the various attributes generated by the algorithm.



(a)

(b)

Figure 3.19: Input to and output from the automated process for Horton Stream Order

One of the advantages is that it prioritises the river segments in a manner that closely resembles natural occurrences.

Major limitations of this technique are:

a) It mandates that the segment orientation in the network satisfy “Horton Net." Horton Net is a network of streams where 1/β (where β is the bifurcation ratio) rivers of order n combine to form a river of order n+1.

b) Biasness either to the catchment area with a larger number of segments or to the flow accumulation may lead to the selection of a false source.

c) Computationally, this technique requires more time as it has to be executed over two passes.

d) Calls for manual intervention in deciding on the true source of the river.

Table 3.3 a): River Network ordering using Horton Stream Order algorithm

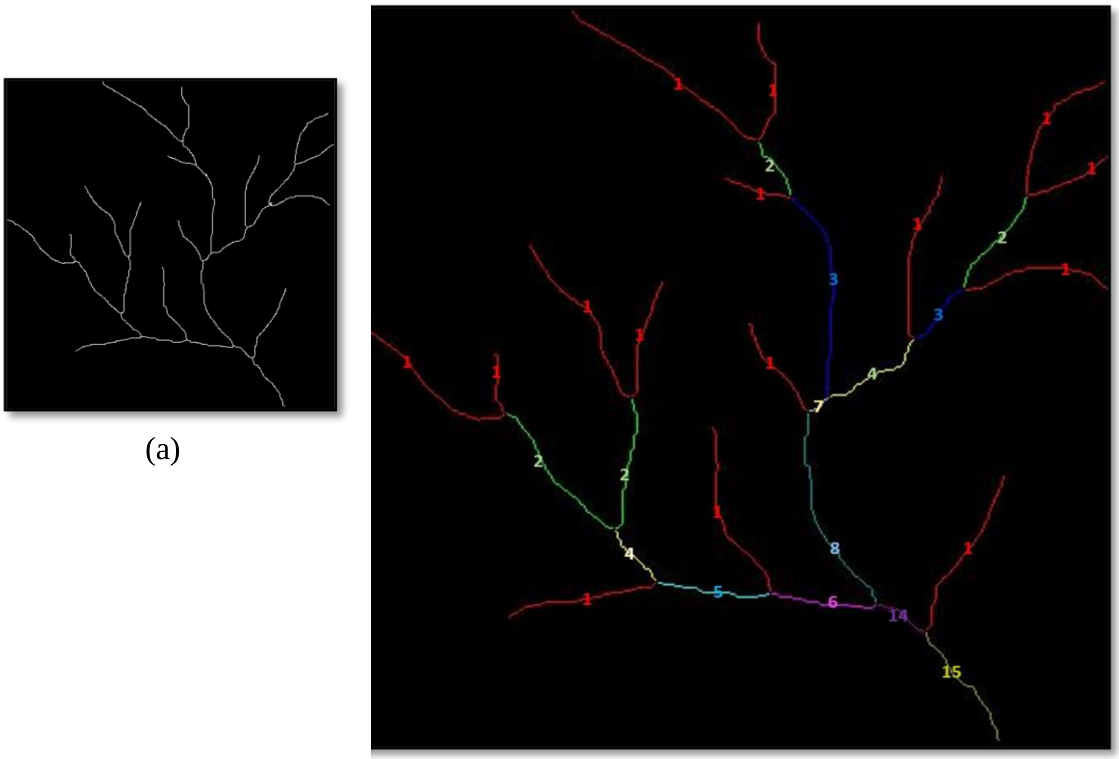
Sl	St Id	Order (S_ω)	Coordinate				Len	Sl	St Id	Order (S_ω)	Coordinate				Len
			s_x	s_y	e_x	e_y					s_x	s_y	e_x	e_y	
1	2	1	10	252	85	251	78	16	21	2	179	377	211	347	39
2	3	1	94	470	120	416	56	17	23	2	211	347	248	295	62
3	6	1	222	91	259	98	41	18	5	3	201	2	259	98	104
4	8	1	175	193	249	174	76	19	19	3	259	98	333	164	87
5	9	1	179	464	179	377	89	20	22	3	333	164	366	190	37
6	10	1	299	401	398	353	102	21	24	3	366	190	373	259	69
7	11	1	389	100	366	190	92	22	26	3	373	259	380	324	65
8	12	1	108	363	211	347	106	23	1	4	2	139	85	251	120
9	13	1	109	232	120	272	42	24	16	4	85	251	120	272	38
10	14	1	202	246	258	282	61	25	20	4	120	272	248	295	130
11	15	1	268	224	373	259	109	26	25	4	248	295	258	282	15
12	4	2	47	462	120	416	90	27	27	4	258	282	380	324	125
13	7	2	152	113	249	174	107	28	28	4	380	324	398	353	30
14	17	2	120	416	179	377	62	29	29	4	398	353	468	398	75
15	18	2	249	174	333	164	85								

Table 3.3 b): River Network characterization using Horton Stream Order algorithm

Sl	S_ω	N_ω	R_b	N_{Rb}	$R_b * N_{Rb}$	R_{bwmean}	L_ω	$L_{\omega mean}$	R_L	N_{RL}	$R_L * N_{RL}$	R_{Lmean}	Cl
1	1	11	-	-	-		852	77.45					
2	2	6	1.83	17	31.17		445	74.17	0.96	1297	1241.94		
3	3	5	1.20	11	13.20		362	72.40	0.98	807	787.78		
4	4	7	0.71	12	8.57		533	76.14	1.05	895	941.27		
Total		**29**	**3.75**	**40**	**52.94**		**1340**	**222.71**	**2.99**	**2999**	**2970.99**		**533**
Mean			**1.87**			**1.32**			**1**			**0.99**	

3.6.4 Shreve Stream Order

Figure 3.20 a) (page no 141) represents the sample input to the Shreve Stream Order algorithm. Figure 3.20 b) (page no 141) represents output from the Shreve Stream Order. Tables 3.4 a) and b) (page no 142 and 143, respectively) represent the various attributes generated by the algorithm.



(a)

(b)

Figure 3.20: Input to and output from the automated process for Shreve Stream Order

One of the major advantages of this technique is that it is easy to understand and simple to use, and it also takes into account the influence of all the converging streams when deciding on the order of the resulting stream. Computationally, it requires less time and space. One of the major limitations is that it also does not allow for the identification of the actual source.

Table 3.4 a): River Network ordering using Shreve Stream Order algorithm

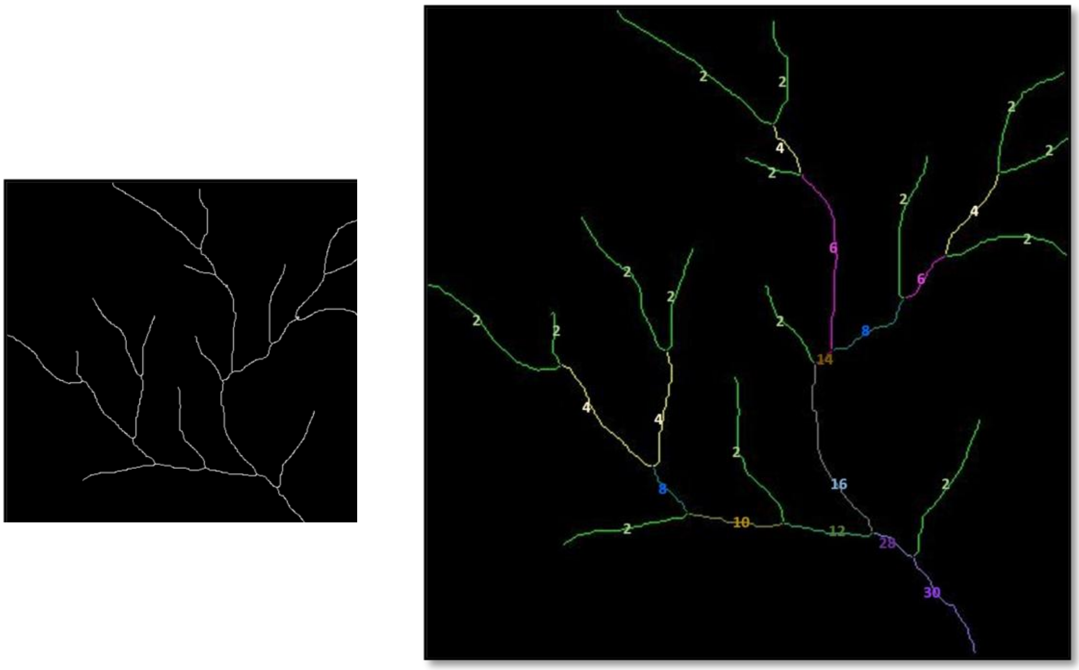
Sl	St Id	Order (S_ω)	Coordinate				Len	Sl	St Id	Order (S_ω)	Coordinate				Len
			s_x	s_y	e_x	e_y					s_x	s_y	e_x	e_y	
1	1	1	2	139	85	251	120	16	16	2	85	251	120	272	38
2	2	1	10	252	85	251	78	17	17	2	120	416	179	377	62
3	3	1	94	470	120	416	56	18	18	2	249	174	333	164	85
4	4	1	47	462	120	416	90	19	19	2	259	98	333	164	87
5	5	1	201	2	259	98	104	20	20	3	120	272	248	295	130
6	6	1	222	91	259	98	41	21	21	3	179	377	211	347	39
7	7	1	152	113	249	174	107	22	23	4	211	347	248	295	62
8	8	1	175	193	249	174	76	23	22	4	333	164	366	190	37
9	9	1	179	464	179	377	89	24	24	5	366	190	373	259	69
10	10	1	299	401	398	353	102	25	26	6	373	259	380	324	65
11	11	1	389	100	366	190	92	26	25	7	248	295	258	282	15
12	12	1	108	363	211	347	106	27	27	8	258	282	380	324	125
13	13	1	109	232	120	272	42	28	28	14	380	324	398	353	30
14	14	1	202	246	258	282	61	29	29	15	398	353	468	398	75
15	15	1	268	224	373	259	109								

Table 3.4 b): River Network characterization using Shreve Stream Order algorithm

Sl	S_ω	N_ω	R_b	N_{Rb}	$R_b * N_{Rb}$	R_{bwmean}	L_ω	$L_{\omega mean}$	R_L	N_{RL}	$R_L * N_{RL}$	R_{Lmean}	Cl
1	1	15	-	-	-		1273	84.87	-	-	-		
2	2	4	3.75	19	71.25		272	68	0.80	1545	1237.94		
3	3	2	2	10	20		169	84.5	1.24	441	548.01		
4	4	2	1	4	4		99	49.5	0.59	268	156.99		
5	5	1	2	3	6		69	69	1.39	168	234.18		
6	6	1	1	2	2		65	65	0.94	134	126.23		
7	7	1	1	2	2		15	15	0.23	80	18.46		
8	8	1	1	2	2		125	125	8.33	140	1166.67		
9	14	1	1	2	2		30	30	0.24	155	37.20		
10	15	1	1	2	2		75	75	2.50	105	262.50		
Total		**29**	**13.75**	**46**	**111.25**		**2192**	**665.87**	**16.27**	**3036**	**3788.19**		**75**
Mean		**2.9**	**1.375**			**2.42**			**1.81**			**1.25**	

3.6.5 Scheidegger Stream Order

Figure 3.21 a) (page no 144) represents the sample input to Scheidegger Stream Order algorithm. Figure 3.21 b) (page no 144) represents output from Scheidegger Stream Order. Table 3.5 a) & b) (page no 145 and 146, respectively) represents the various attributes generated by the algorithm.



(b)

Figure 3.21: Input to and output from the automated process for Scheidegger Stream Order

One of the major advantages of this technique is that it is easy to understand and simple to use, and it also takes into account the influence of all the converging streams when deciding on the order of the resulting stream. Computationally, it requires less time compared to the Shreve technique, as it uses even numbers to represent the order of the network segments. One of the major limitations is that it also does not allow for the identification of the actual source.

Table 3.5 a): River Network ordering using Scheidegger Stream Order algorithm

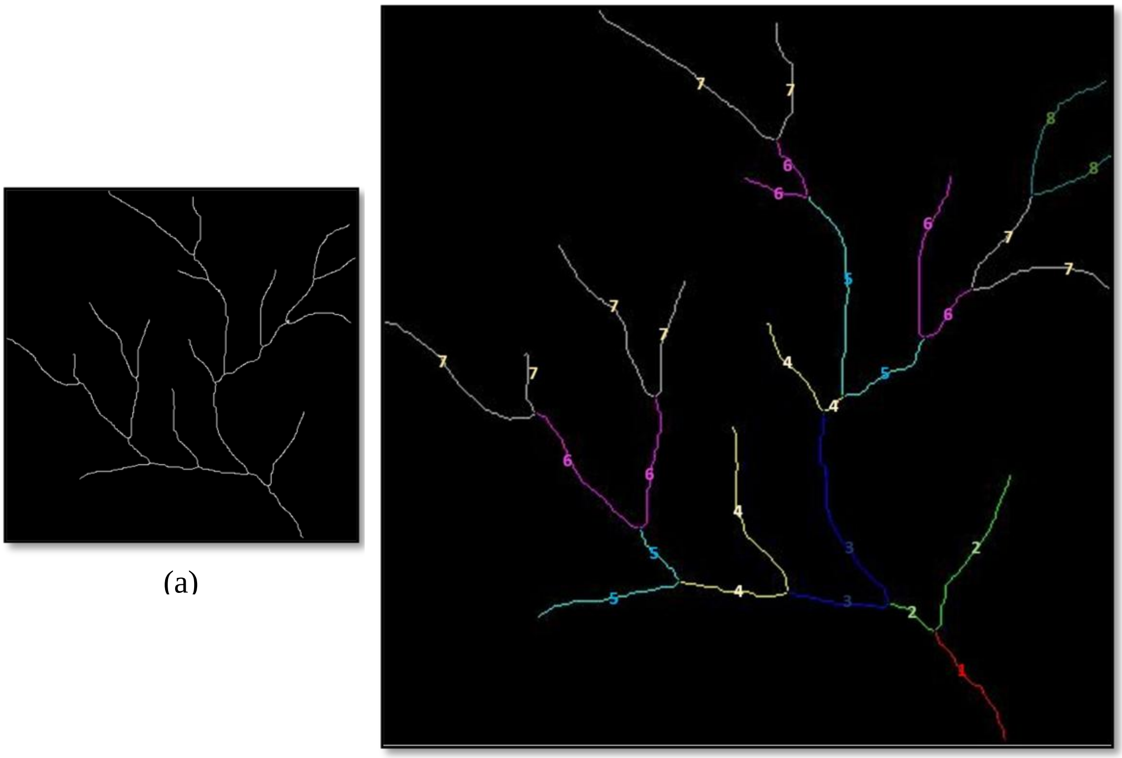
Sl	St Id	Order (S_ω)	Coordinate				Len
			s_x	s_y	e_x	e_y	
1	1	2	2	139	85	251	120
2	2	2	10	252	85	251	78
3	3	2	94	470	120	416	56
4	4	2	47	462	120	416	90
5	5	2	201	2	259	98	104
6	6	2	222	91	259	98	41
7	7	2	152	113	249	174	107
8	8	2	175	193	249	174	76
9	9	2	179	464	179	377	89
10	10	2	299	401	398	353	102
11	11	2	389	100	366	190	92
12	12	2	108	363	211	347	106
13	13	2	109	232	120	272	42
14	14	2	202	246	258	282	61
15	15	2	268	224	373	259	109
16	16	4	85	251	120	272	38
17	17	4	120	416	179	377	62
18	18	4	249	174	333	164	85
19	19	4	259	98	333	164	87
20	20	6	120	272	248	295	130
21	21	6	179	377	211	347	39
22	23	8	211	347	248	295	62
23	22	8	333	164	366	190	37
24	24	10	366	190	373	259	69
25	26	12	373	259	380	324	65
26	25	14	248	295	258	282	15
27	27	16	258	282	380	324	125
28	28	28	380	324	398	353	30
29	29	30	398	353	468	398	75

Table 3.5 b): River Network characterization using Scheidegger Stream Order algorithm

Sl	S_ω	N_ω	R_b	N_{Rb}	R_b * N_{Rb}	R_{bwmean}	L_ω	$L_{\omega mean}$	R_L	N_{RL}	R_L * N_{RL}	R_{Lmean}	Cl
1	2	15	-	-	-		1273	84.87	-	-	-		
2	4	4	3.75	19	71.25		272	68	0.80	1545	1237.94		
3	6	2	2	10	20		169	84.5	1.24	441	548.01		
4	8	2	1	4	4		99	49.5	0.59	268	156.99		
5	10	1	2	3	6		69	69	1.39	168	234.18		
6	12	1	1	2	2		65	65	0.94	134	126.23		
7	14	1	1	2	2		15	15	0.23	80	18.46		
8	16	1	1	2	2		125	125	8.33	140	1166.67		
9	28	1	1	2	2		30	30	0.24	155	37.20		
10	30	1	1	2	2		75	75	2.50	105	262.50		
Total		**29**	**13.75**	**46**	**111.25**		**2192**	**665.87**	**16.27**	**3036**	**3788.19**		**75**
Mean		**2.9**	**1.375**			**2.42**			**1.81**			**1.25**	

3.6.6 Order by path length method

Figure 3.22 a) (page no 147) represents the sample input to Order by path length Stream Order algorithm. Figure 3.22 b) (page no 147) represents output from Order by path length Stream Order. Table 3.6 a) & b) (page no 14) represents the various attributes generated by the algorithm.



(a)

(b)

Figure 3.22: Input to and output from the automated process for Order by path length Stream Order

This technique is relatively easy to understand and computationally requires less time and space. One of the major limitations is that it also does not allow for the identification of the actual source.

Table 3.6 a): River Network ordering using Order by Path Length Stream Order algorithm

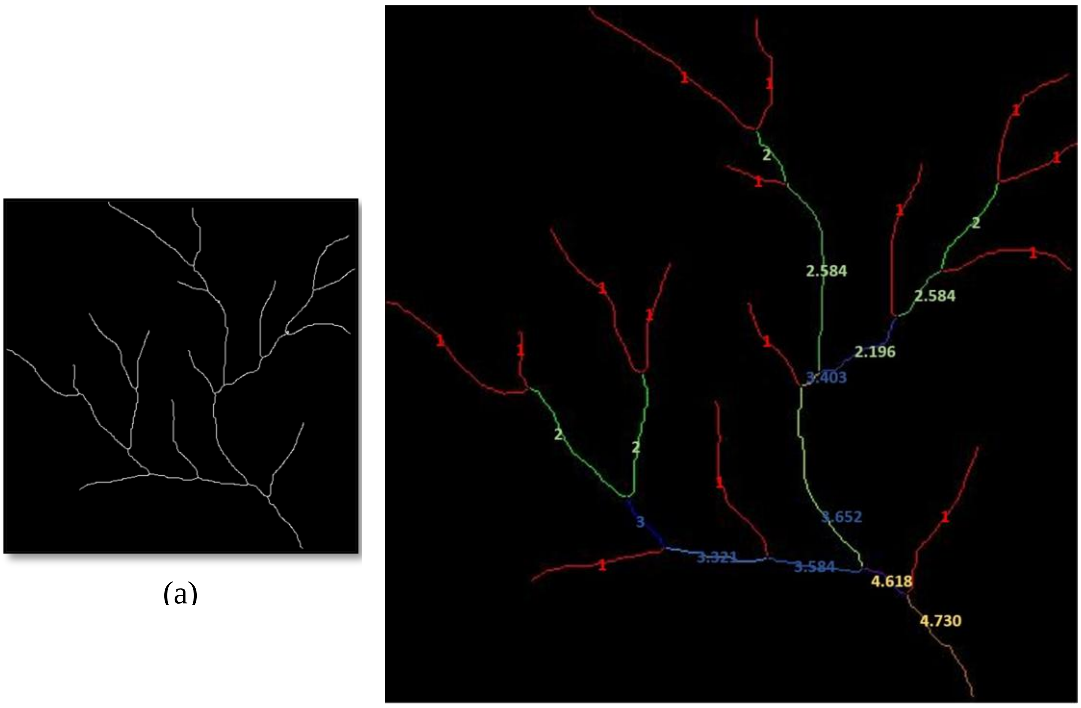
Sl	St Id	Order (S_ω)	Coordinate				Len	Sl	St Id	Order (S_ω)	Coordinate				Len
			s_x	s_y	e_x	e_y					s_x	s_y	e_x	e_y	
1	29	1	398	353	468	398	75	16	16	6	85	251	120	272	38
2	10	2	299	401	398	353	102	17	18	6	249	174	333	164	85
3	28	2	380	324	398	353	30	18	19	6	259	98	333	164	87
4	26	3	373	259	380	324	65	19	21	6	179	377	211	347	39
5	27	3	258	282	380	324	125	20	1	7	2	139	85	251	120
6	14	4	202	246	258	282	61	21	2	7	10	252	85	251	78
7	15	4	268	224	373	259	109	22	5	7	201	2	259	98	104
8	24	4	366	190	373	259	69	23	6	7	222	91	259	98	41
9	25	4	248	295	258	282	15	24	7	7	152	113	249	174	107
10	11	5	389	100	366	190	92	25	8	7	175	193	249	174	76
11	20	5	120	272	248	295	130	26	9	7	179	464	179	377	89
12	23	5	211	347	248	295	62	27	17	7	120	416	179	377	62
13	22	5	333	164	366	190	37	28	3	8	94	470	120	416	56
14	12	6	108	363	211	347	106	29	4	8	47	462	120	416	90
15	13	6	109	232	120	272	42								

Table 3.6 b): River Network characterization using Order by Path Length Stream Order algorithm

Sl	S_ω	N_ω	R_b	N_{Rb}	$R_b * N_{Rb}$	R_{bwmean}	L_ω	$L_{\omega mean}$	R_L	N_{RL}	$R_L * N_{RL}$	R_{Lmean}	Cl
1	1	1	-	-	-		75	75	-	-	-		
2	2	2	0.5	3	1.5		132	66	0.88	207	182.16		
3	3	2	1	4	4		190	95	1.44	322	463.48		
4	4	4	0.5	6	3		254	63.5	0.67	444	296.78		
5	5	4	1	8	8		321	80.25	1.26	575	726.67		
6	6	6	0.67	10	6.67		397	66.17	0.82	718	592.00		
7	7	8	0.75	14	10.5		677	84.63	1.28	1074	1373.61		
8	8	2	4	10	40		146	73	0.86	823	709.94		
Total		**29**	**8.42**	**55**	**73.67**		**2192**	**603.54**	**7.22**	**4163**	**4344.65**		**75**
Mean		**3.63**	**1.05**			**1.34**			**1.03**			**1.04**	

3.6.7 Consistent Stream Order

Figure 3.23 a) (page no 149) represents the sample input to the Consistent Stream Order algorithm. Figure 3.23 b) (page no 149) represents the output from the Consistent Stream Order algorithm. Tables 3.7 a) and b) (page no 150 and 151, respectively) represent the various attributes generated by the algorithm.



(a)

(b)

Figure 3.23: Input to and output from the automated process for Consistent Stream Order

One of the major advantages of this technique is that it takes into account the influence of all the converging streams when deciding on the order of the resulting stream. Computationally, it requires more time and space. One of the major limitations is that it also does not allow for the identification of the actual source.

Table 3.7 a): River Network ordering using Order by Consistent Stream Order algorithm

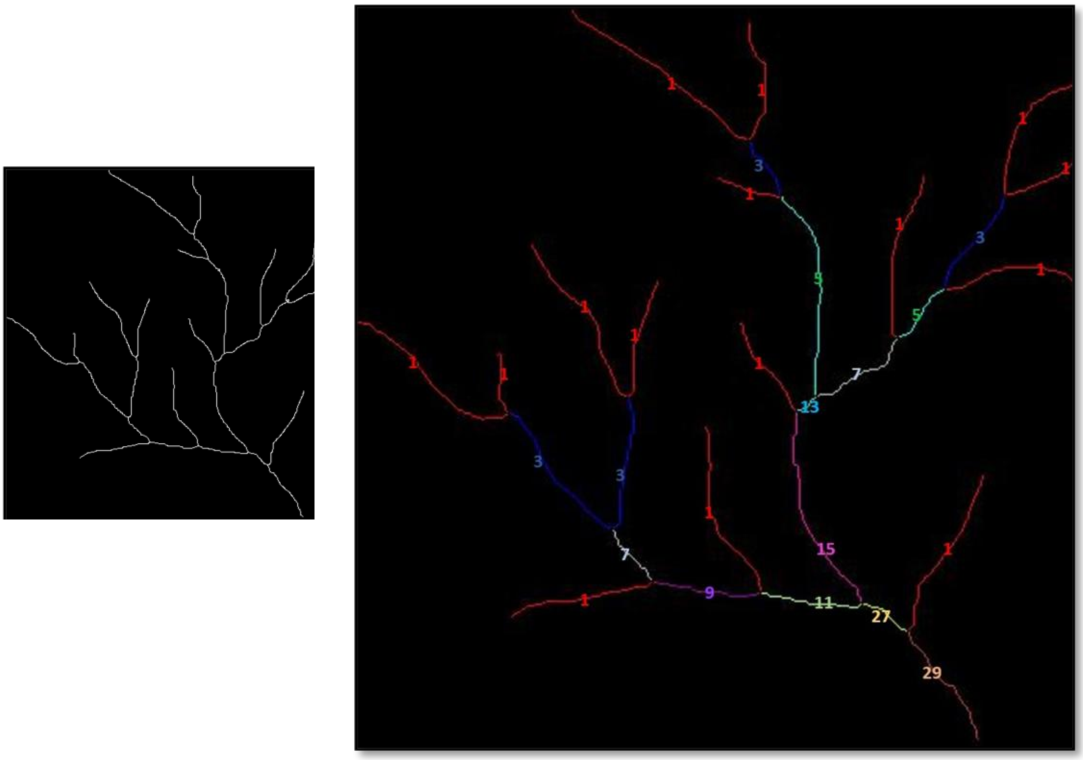
Sl	St Id	Order (S_{ω})	Coordinate				Len	Sl	St Id	Order (S_{ω})	Coordinate				Len
			s_x	s_y	e_x	e_y					s_x	s_y	e_x	e_y	
1	1	1	2	139	85	251	120	16	16	2.00	85	251	120	272	38
2	2	1	10	252	85	251	78	17	17	2.00	120	416	179	377	62
3	3	1	94	470	120	416	56	18	18	2.00	249	174	333	164	85
4	4	1	47	462	120	416	90	19	19	2.00	259	98	333	164	87
5	5	1	201	2	259	98	104	20	20	2.58	120	272	248	295	130
6	6	1	222	91	259	98	41	21	21	2.58	179	377	211	347	39
7	7	1	152	113	249	174	107	22	23	2.20	211	347	248	295	62
8	8	1	175	193	249	174	76	23	22	3.00	333	164	366	190	37
9	9	1	179	464	179	377	89	24	24	3.32	366	190	373	259	69
10	10	1	299	401	398	353	102	25	26	3.58	373	259	380	324	65
11	11	1	389	100	366	190	92	26	25	3.40	248	295	258	282	15
12	12	1	108	363	211	347	106	27	27	3.65	258	282	380	324	125
13	13	1	109	232	120	272	42	28	28	4.62	380	324	398	353	30
14	14	1	202	246	258	282	61	29	29	4.73	398	353	468	398	75
15	15	1	268	224	373	259	109								

Table 3.7 b): River Network characterization using Consistent Stream Order algorithm

Sl	S_{ω}	N_{ω}	R_b	N_{Rb}	$R_b * N_{Rb}$	R_{bwmean}	L_{ω}	$L_{\omega mean}$	R_L	N_{RL}	$R_L * N_{RL}$	R_{Lmean}	Cl
1	1	15	-	-	-		1273	84.87	-	-	-		
2	2	4	3.75	19	71.25		272	68.00	0.80	1545	1237.94		
3	2.584	2	2	6	12		169	84.50	1.24	441	548.01		
4	2.196	1	2	3	6		62	62.00	0.73	231	169.49		
5	3	1	1	2	2		37	37.00	0.60	99	59.08		
6	3.321	1	1	2	2		69	69.00	1.86	106	197.68		
7	3.584	1	1	2	2		65	65.00	0.94	134	126.23		
8	3.403	1	1	2	2		15	15.00	0.23	80	18.46		
9	3.652	1	1	2	2		125	125.00	8.33	140	1166.67		
10	4.618	1	1	2	2		30	30.00	0.24	155	37.20		
11	4.73	1	1	2	2		75	75.00	2.50	105	262.50		
Total		**29**	**14.75**	**42**	**101.25**		**2192**	**640.37**	**17.49**	**3036**	**3560.76**		**75**
Mean		**2.636**	**1.475**			**2.41**			**1.75**			**1.17**	

3.6.8 Cumulative Stream Order

Figure 3.24 a) (page no 152) represents the sample input to the Cumulative Stream Order algorithm. Figure 3.24 b) (page no 152) represents the output from the Cumulative Stream Order algorithm. Tables 3.8 a) and b) (page no 153 and 154, respectively) represent the various attributes generated by the algorithm.



(b)

Figure 3.24: Input to and output from the automated process for Cumulative Stream Order

One of the major advantages of this technique is that cumulative order has a direct relationship with discharge as it takes into account the influence of all the streams in the watershed when deciding on the order of the resulting stream. It is also relatively easy to understand and, computationally, requires less time and space. One of the major limitations is that it also does not allow for the identification of the actual source.

Table 3.8 a): River Network ordering using Order by Cumulative Stream Order algorithm

Sl	St Id	Order (S_ω)	Coordinate				Len	Sl	St Id	Order (S_ω)	Coordinate				Len
			s_x	s_y	e_x	e_y					s_x	s_y	e_x	e_y	
1	1	1	2	139	85	251	120	16	16	3	85	251	120	272	38
2	2	1	10	252	85	251	78	17	3	3	120	416	179	377	62
3	3	1	94	470	120	416	56	18	18	3	249	174	333	164	85
4	4	1	47	462	120	416	90	19	19	3	259	98	333	164	87
5	5	1	201	2	259	98	104	20	20	5	120	272	248	295	130
6	6	1	222	91	259	98	41	21	21	5	179	377	211	347	39
7	7	1	152	113	249	174	107	22	23	7	211	347	248	295	62
8	8	1	175	193	249	174	76	23	22	7	333	164	366	190	37
9	9	1	179	464	179	377	89	24	24	9	366	190	373	259	69
10	10	1	299	401	398	353	102	25	26	11	373	259	380	324	65
11	11	1	389	100	366	190	92	26	25	13	248	295	258	282	15
12	12	1	108	363	211	347	106	27	27	15	258	282	380	324	125
13	13	1	109	232	120	272	42	28	28	27	380	324	398	353	30
14	14	1	202	246	258	282	61	29	29	29	398	353	468	398	75
15	15	1	268	224	373	259	109								

Table 3.8 b): River Network characterization using Cumulative Stream Order algorithm

Sl	S_ω	N_ω	R_b	N_{Rb}	R_b * N_{Rb}	R_{bwmean}	L_ω	$L_{\omega mean}$	R_L	N_{RL}	R_L * N_{RL}	R_{Lmean}	Cl
1	1	15	-	-	-		1273	84.87	-	-	-		
2	3	4	3.75	19	71.25		272	68	0.80	1545	1237.94		
3	5	2	2	10	20		169	84.5	1.24	441	548.01		
4	7	2	1	4	4		99	49.5	0.59	268	156.99		
5	9	1	2	3	6		69	69	1.39	168	234.18		
6	11	1	1	2	2		65	65	0.94	134	126.23		
7	13	1	1	2	2		15	15	0.23	80	18.46		
8	15	1	1	2	2		125	125	8.33	140	1166.67		
9	27	1	1	2	2		30	30	0.24	155	37.20		
10	29	1	1	2	2		75	75	2.50	105	262.50		
Total		**29**	**13.75**	**46**	**111.25**		**2192**	**665.87**	**16.27**	**3036**	**3788.19**		**75**
Mean		**2.9**	**1.528**			**2.42**			**1.81**			**1.25**	

3.6.9 Stream Characterization

The following ten stream characteristics were determined as detailed in serials 3.2.1.1-3.2.1.11 above: Stream Order (S_ω), Stream Number (N_ω), Bifurcation Ratio (R_b), Streams participating in Bifurcation Ratio (N_{Rb}), Weighted Mean Bifurcation Ratio (R_{bwmean}), Stream Length (L_ω), Mean Stream Length ($L_{\omega mean}$), Length ratio (R_L), Streams participating in Length Ratio (N_{RL}), Weighted Mean Length Ratio (R_{Lwmean}) and Length of Main Channel (Cl) in all the eight ordering techniques, and the same is presented in tables 3.1-3.8.

3.7 Limitations of the research initiative

Below mentioned are some of the notable limitations of the proposed research initiative:

a) Variations in the pixel intensity values of the color shade representing the feature of interest in the topographic sheet greatly influence the result of the digitization process.

b) The overall effectiveness of the proposed method relies on the capability of the digitization process.

c) Inaccurate digitization will lead to erroneous results and improper stream characterization.

d) The use of inappropriate filters for the removal of noise may hamper the quality of the results.

e) The inability to identify the terminal or peripheral streams would disturb the overall effectiveness of the river ordering process.

f) Hydrological characterization, its simulations, and forecasting can only be achieved if the findings of the research initiative are combined with other datasets like DEM and satellite imagery.

g) Temporal analysis is relatively easy through the use of satellite images compared to topographic datasets, as these datasets are released at longer intervals.

3.8 Conclusion

This research initiative presents an effective and efficient alternative to traditional manual digitization by conceiving and realising knowledge-enabled automated computational programmes leveraging the capability of technological advancements for digitising river networks, their ordering, and the generation of representable characteristics. The intention of the research work is to limit human intervention in the digitization process transitively, leading to a reduction in effort, time, and cost requirements. In addition, it also helps in elevating the result's confidence.

In pursuit of the aforementioned objective, the proposed work has conceived and successfully realised modules for the digitization of river patterns, elimination of noises, skeletonization of digitized river patterns, resolving connected neighbors, traversal for detecting different types of streams, various stream ordering techniques, and generation of stream attributes. Here skeletonization has been performed using the Zhang-Suen thinning algorithm, connected neighbours have been resolved using m-connectivity, and an efficient spiral traversal has been used for traversal of the stream segments.

This work implements eight river ordering algorithms, namely Classic Stream Order, Strahler Stream Order, Horton Stream Order, Shreve Stream Order, Scheidegger Stream Order, Order by Path Length Method, Consistent River Ordering, and Cumulative Stream Order. The computational programme developed was capable of accurately and efficiently classifying the stream segment into various classes as per the guiding principles of the various ordering techniques and subsequently assigning unique color codes to them for ease of visual interpretation.

The work was further advanced to associate various stream characteristics such as Stream Order (Sω), Stream Number (Nω), Bifurcation Ratio (Rb), Streams participating in Bifurcation Ratio (NRb), Weighted Mean Bifurcation Ratio (Rbwmean), Stream Length (Lω), Mean Stream Length (Lωmean), Length ratio (RL), Streams participating in Length Ratio (NRL), Weighted Mean Length Ratio (RLwmean) and Length of Main Channel (Cl).

The proficiency of the conceived program was tested on a few samples of river segments, one of which is highlighted in Figure 3.4 a) (page no 110). The program was able to correctly realise the motives of all the proposed modules as well as generate values for various attributes related to the river segments, as represented in table 3.1-3.8.

Further, a qualitative comparative assessment of all of the river ordering techniques was performed with regard to attributes such as the method used, the number of passes required, the computational cost, its advantages and disadvantages, and the need for human intervention.

The outcome of the proposed research initiative may be combined with aspects such as the DEM and factual data from satellite imagery, which can be extremely useful in:

a) simulating hydrological processes and deriving various hydrological characteristics

b) studying morphological changes in the structure of the river pattern through temporal analysis

c) forecasting, planning, and containing the effect of disasters

d) planning sustainable developmental initiatives along the river network

e) studying geological changes due to erosion, transportation, and deposition through temporal analysis

Chapter 4

Extraction of Contour Lines, its Refinement and Attribute Generation

4.1 Introduction

A typical topographical sheet represents a wide variety of details, elaborately describing the relief of the landscape with the help of non-intersecting lines called contours [74] [233]. Contour lines are lines on the topographic sheet generated by connecting points at the same elevation with regard to a certain reference point. The topographic sheet not only includes contour lines but also other crucial features such as water bodies (i.e., rivers, lakes), route networks, nomenclature associated with features, bridges, temples, boundaries, and many more [234] [86].

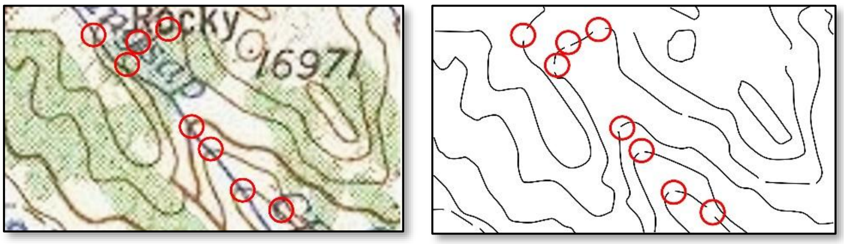
To efficiently manage content, these features are represented with the help of distinctive color codes in an overlapping manner [92]. One can easily distinguish various features through visual inspection. Research initiatives motivated by a morphological analysis of terrain demand the segregation of these features into distinctive thematic layers called vectors [79] [86]. It may be attained by deploying a tedious manual approach or by conceiving an effective automatic process. Traditional approaches demand tremendous effort, time and are often found to be subjected to or biased towards the experience and skill of the digitizer [226] [75].

On the contrary, automated processes are relatively inexpensive but demand greater effort and time for the conception and realization through optimal and efficient computational strategies [235] [95]. Further, to elevate the quality of the result, these processes are expected to seamlessly integrate or employ a decision support mechanism for effectively handling exceptions arising due to improper segmentation.

4.1.1 Problems associated with segmented contour lines

On critically assessing the sample segmented dataset of a contour map, some of the pertinent problems identified are highlighted below:

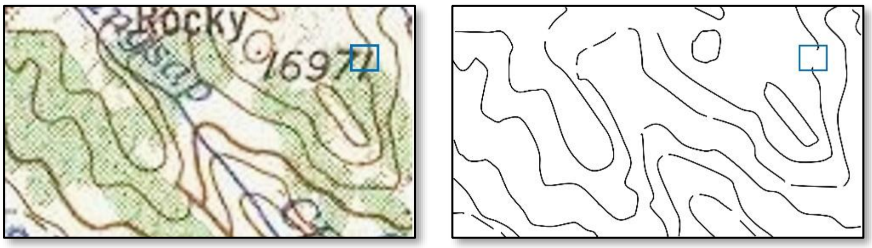
a) To efficiently manage contextual information, morphological features are often represented in an overlapping manner [92]. It is observed that at the point of intersection of features, the pixel intensity abruptly changes due to color imposition, and as a consequence of the same, the pixels in and around the point of intersection cannot be categorically placed in the dataset resembling contours. figure 4.1 a) (page no 160) represent a sample portion of a topographic sheet with overlapping features and figures 4.1 b) (page no 160) breakage in contour lines, respectively, demarcated with the help of red circles.



a) b)

Figure 4.1: a) Sample Topographic Sheet [1] b) Breakage created due to presence of overlapping features

b) To facilitate the 3D transformation of contour lines for generating elevating models commonly known as DEM, elevation values are associated with the contour lines at certain intervals [236]. Digitization of the same results in the creation of broken contour lines [78]. Figure 4.2 a) (page no 161) represents the sample portion of a topographic sheet with elevation values and figure 4.2 b) (page no 161) represents breakage in contour lines due to the same, demarcated with the help of a blue square.

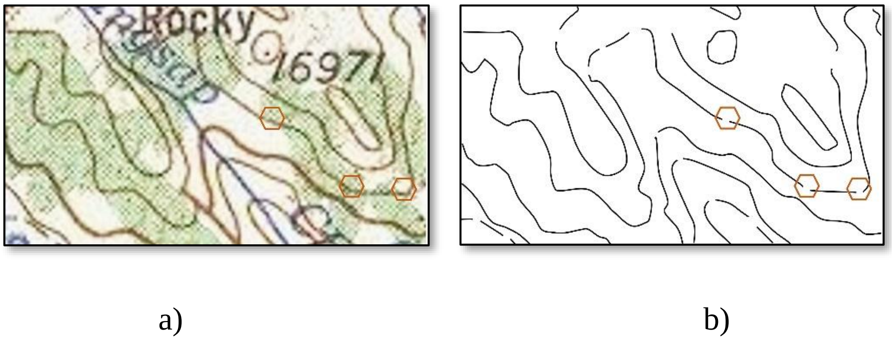


a) b)

Figure 4.2: a) Sample Topographic Sheet [1] b) Breakage point due to presence of elevation value

c) Automatic digitization greatly leverages the capabilities of the color segmentation process [75]. One of the key inputs to the segmentation process is the range of intensity values representing the feature to be segmented. To contain or limit the error of commission (false value), it becomes very important to certainly fix the range of intensity values representing the contours set as input. In the process, some significant pixels may be omitted (error of omission), leading to the generation of broken contour lines. Further, it is also to be understood that the range of intensity values representing contours is highly dynamic and greatly dictated or influenced by environmental factors such as printing provision, data capture tools, and age.

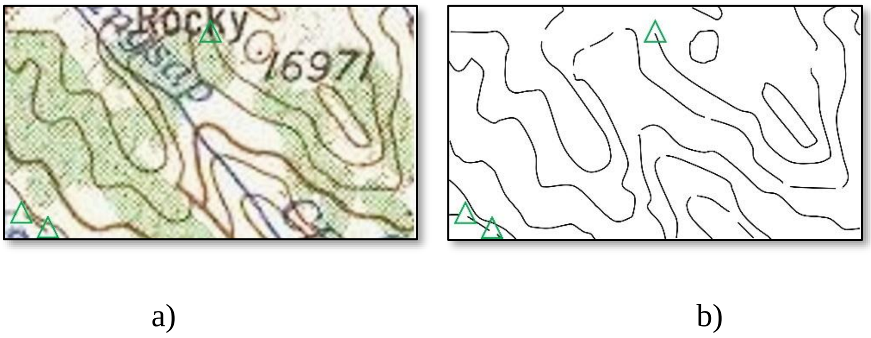
Figure 4.3 a) (page no 162) represents sample portions of a topographic sheet and figure 4.3 b) (page no 162) breakage in contour lines due to incorrect segmentation processes, demarcated with the help of a brown hexagon.



a) b)

Figure 4.3: a) Sample Topographic Sheet [1] b) Breakage point due to incorrect segmentation process

d) It has also been observed that due to restrictions on information representation, contours are poorly specified, leading to the generation of broken contour lines. Figure 4.4 a) (page no 162) represents the sample portion of a topographic sheet with missing content and figure 4.4 b) (page no 162) represents a contour line due to the same, demarcated with the help of the green triangle.



a) b)

Figure 4.4: a) Sample Topographic Sheet with missing content [1] b) Contours with missing content

Above specified are some of the notable causes for breakage of contour lines adversely affecting the continuity of contours.

4.1.2 Necessity for maintaining contour continuity

Research initiatives motivated toward the use of contour lines for geomorphological analysis demand the continuity of contours for the following reasons:

a) The contour line creates a true reflection of the very nature of the ground surface (reality) with its complete morphological description, such as mountains, plateaus, flat lands, etc.

b) Contour lines are the fundamental basis used for generating a 3D model known as the DEM [236]. It is achieved by elevating the contour lines in the z-plane with regard to the elevation value associated with it.

c) Further, the superimposition of satellite imaginary onto the 3D elevation models generated using contour lines enables the creation of the DSM.

d) Contour lines are extremely useful for the purpose of terrain analysis [237]. It is crucial for determining suitability, planning, and executing sustainable earthworks such as route network planning: road, rail, waterways, etc.

   - Infrastructure development planning: engineering work, construction, mining, etc.
   - Landscape planning: utilization, urban planning, etc.
   - Resource planning: reservoirs, dams, watersheds, catchment, etc.

e) Contour lines are extremely useful for gradient analysis of landscapes and determining geomorphological alignments.

f) Contour lines are also widely used for determining intervisibility between selected ground control points on the map. Such studies have proved effective in the design of overhead powerlines, ropeways, flyover bridges, and many more.

g) Elevation maps created with the aid of contour lines are extremely useful for hydrological analysis of river networks or waterways.

h) With regards to the information contained in the contour lines, integrity-preserving knowledge-based computational algorithms and procedures may be conceived and realised for the generation of additional morphological features (such as interpolated contour lines) useful for elevating the quality of 3D representation.

i) By integrating contours with geological data (such as soil and rock formation), hydrological data (such as watershed, river network, and reservoir), and metrological data (such as rainfall and precipitation), vulnerability mapping of the landscape can be done to determine the suitability of inhabitation.

j) An elevation model created using contour lines with satellite imaginary can be used for mapping and efficiently managing life stock (flora and fauna).

4.1.3 Challenges encountered

Some of the notable challenges faced during the reconstruction process were:

a) Features having a close resemblance (in terms of pixel intensity) with contours may be inappropriately segmented as contours, leading to errors of commission [81].

b) The dynamic nature of the range of intensity values pertaining to contours may lead to the omission of significant values leading to errors of omission.

c) One of the crucial factors on which the alignment of contours depends is gradient [238] [239]. It is observed that based on the gradient, the contours may be sparsely or closely placed. In situations where the contours are placed in close proximity, the segmentation process becomes extremely difficult.

d) A terminal point may be wrongly selected in the process of extending the continuity of contour lines due to computational lag.

e) A contour line may be prone to severe breakages wherein the decision support system may not be able to select appropriate points or may even fail to do the same for ensuring continuity.

4.1.4 Research Motivations

Through a thorough analysis of relevant literature, the following important derivations were made:

a) Preprocessing operations including color segmentation, binarization, thinning, removal of holes & bifurcations, and noise reduction are essential for reducing the computational overhead involved in processing a topographic sheet. While performing image segmentation, hard decisions should be avoided as intensity values are highly dynamic and vary from sample to sample.

b) The strategy for determining terminal points should be proximity-based, i.e., based on inter-terminal point distance.

c) Some of the crucial information that may be pursued for determining terminal points for reconnection may be:

- Morphological orientation of contour elements
- Direction of alignment of contours
- Geometrical orientation of contours
- Morphological information of adjacent contours

d) While performing reconnection, it is essential to preserve:

- Geometric structure of the contours
- Structural integrity of the contours

e) While performing reconnection, it is essential to address the following key aspects for reducing computational overhead:

- Use an optimal number of control points for retaining the characteristics of contours
- The design approach should be able to constrict the list of potential terminal points for reconnection
- Localization of the search space is crucial for managing computational complexity
- Selection of optimal points based on a gradient and distance metric

f) Some of the generic processes involved in similar research initiatives from the past are:

- Backtracking is essential for determining the nature and characteristics of contour lines
- The association of probability distribution with other contour features such as gradient and orientation would enable the selection of optimal terminal points.

4.2 Objectives of the proposed work

With regards to the above specified motivations, the objective of the proposed research initiative is to:

a) Hybridize Continuity-based and Gradient-based techniques for connecting broken contour pairs to ensure connectivity

b) Use gradient information to determine matching end-point pair

c) Apply curve drawing technique based on geometric characteristics for interpolating pixels between matched endpoints preserving integrity

d) The above-mentioned task may be achieved through use of concepts such as the Sign of Gradient (SG), Euclidean Distance (ED), and modified Bezier Curve (BC) drawing technique

4.3 Applications of the proposed research initiative

Contours in topographic sheet represents a prime geographical features of earth surface. An automatic extraction and representation of contours from topographic sheet has wide variety of applications in the field of engineering, planning, environment science, disaster management, recreation, and many more. The contours can be used to analyse the terrain of earth surface and aids to understand elevation changes and steepness of area of interest. This is particularly useful for land use planning, construction, etc. Similarly, the urban planner uses it for understanding the suitability of urban development for habitant considering drainage pattern and other potential challenges. The civil engineers use contour information during infrastructural development like construction of roads, railways, bridges, etc. In the field of hydrology, the feature can be used to model watershed, forecast flood pattern, and design control mechanism. Likewise, contours are also used in other disaster management situation like landslide to perform search and rescue and operation in challenging terrain areas. With not limited to, contour information are also extensively used by geologist for exploring mineral deposits and other geographical formations and agriculturalist for planning and estimation of efficient farming. In the field of tourism and recreation, contours aids in navigation for planning hiking, camping areas, and marking wilderness areas for ensuring safety.

4.4 Proposed Methodology

With regards to the above-specified significance of contour lines, the proposed research initiative is motivated towards designing and developing automatic procedures for elevating the quality of segmented contours through the below-mentioned sequence of steps:-

a) Segmentation of contour lines from the topographic sheet

b) Determining breakage along the segmented contour lines

c) Detecting terminal points of breakages with the objective of determining suitable points for reconnection

d) Establishing connectivity between suitable points based on proximity using a cost-effective algorithmic approach

The proposed techniques adheres to the following sequence of steps, as detailed in figure 4.5 (page no 169).

Below mentioned are the sequence of steps pursued for attaining the research objectives:

a) Digitalize Topographic Sheet: Here, 24 sample portions of different topographic sheets were selected and digitalized for processing.

b) Digitize Contour Lines: Contour lines were digitized into distinct vectors from the samples with the help of a color segmentation process utilizing the range of intensity values representing the said feature. It was found that the range of intensity values representing contours are dynamic and sample specific. Figures 4.6 a) and 4.6 b) (page no 170) represent a sample digitalized topographic sheet and its corresponding digitized grayscale contour map, respectively.

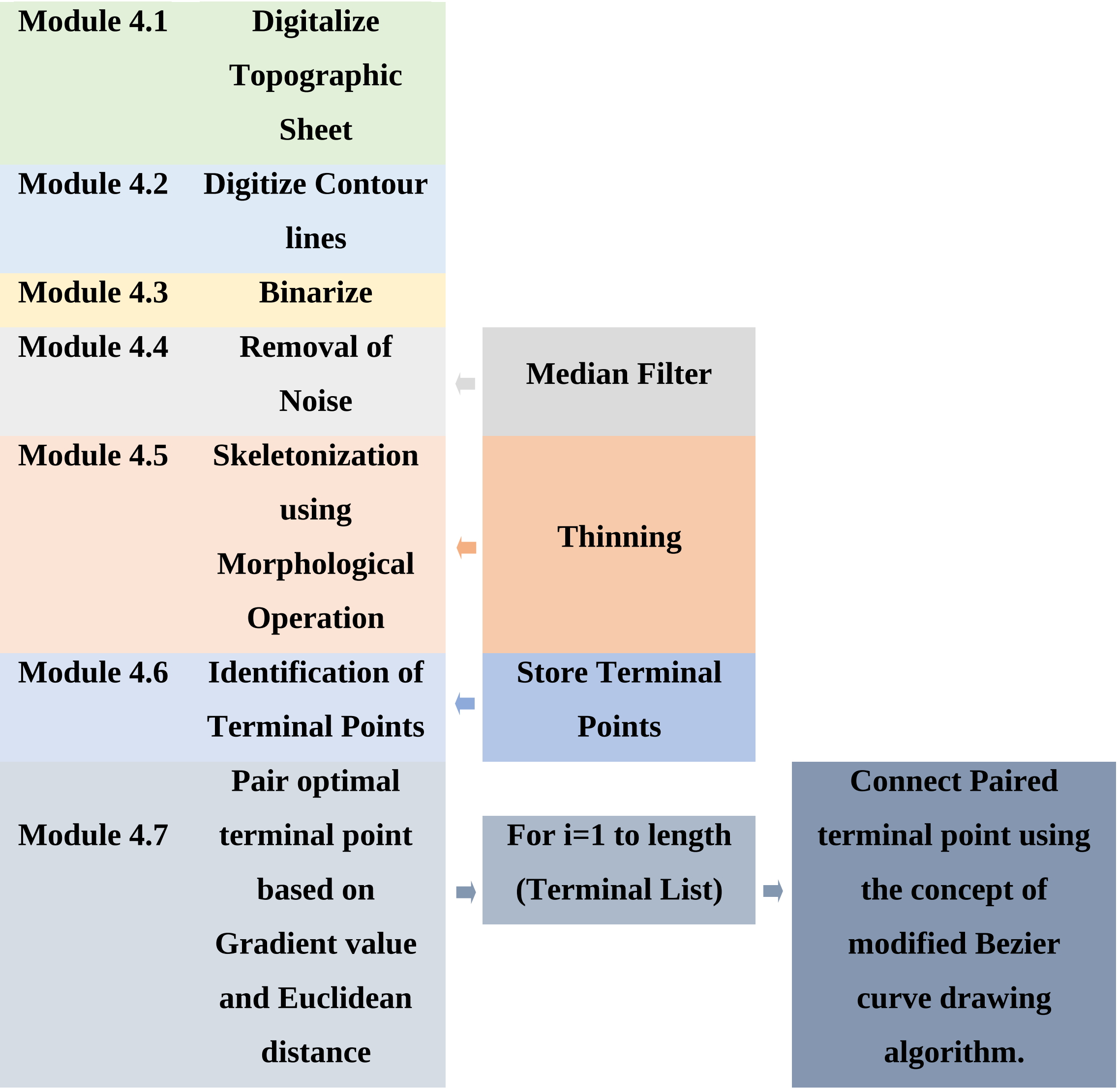
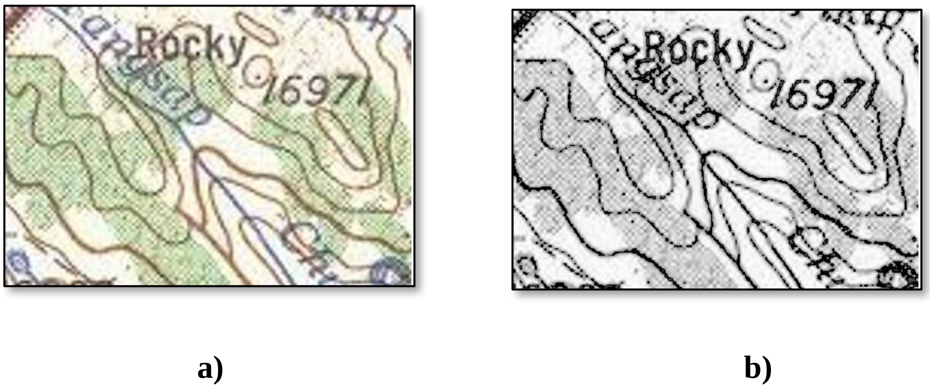

Module 4.1
Digitalize Topographic Sheet
Module 4.2
Digitize Contour lines
Module 4.3
Binarize
Module 4.4
Removal of Noise
Median Filter
Module 4.5
Skeletonization using Morphological Operation
Thinning
Module 4.6
Identification of Terminal Points
Store Terminal Points
Module 4.7
Pair optimal terminal point based on Gradient value and Euclidean distance
For i=1 to length (Terminal List)
Connect Paired terminal point using the concept of modified Bezier curve drawing algorithm.

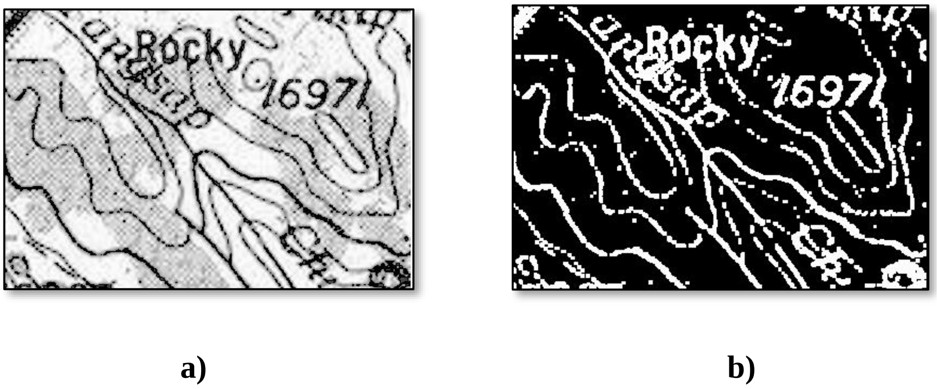
Figure 4.5: Flow diagram for the proposed research initiatives

c) **Binarize:** The digitized information was then binarized to simplify representation and to further ease computation. Here, the significant values are represented with '1' and the non-significant values are represented with '0'. Figures 4.7 a) and b) (page no 170) transformation of a digitized grayscale contour map into a binarized contour map.



a) **b)**

Figure 4.6: a) Sample Topographic Sheet [1] b) Digitized grayscale contour map

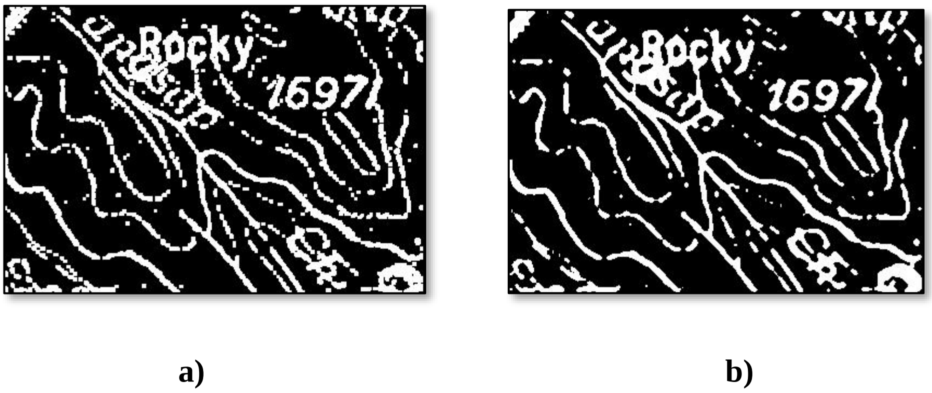


a) **b)**

Figure 4.7: a) Digitized grayscale contour map b) Binarized contour map

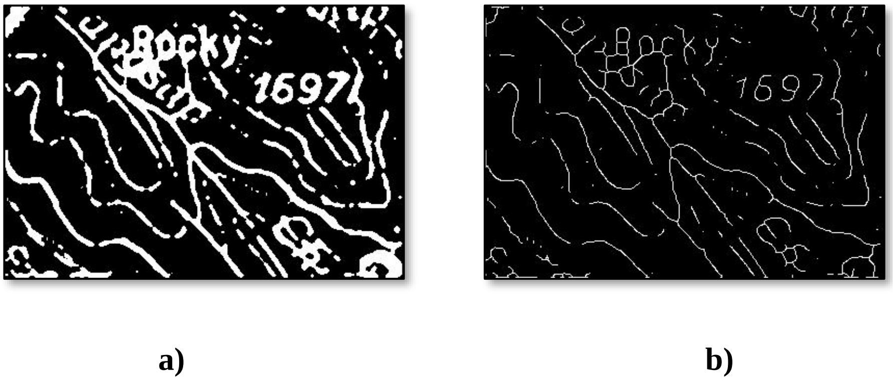
d) Noise Removal: The digitized vectors may have irrelevant content or information as a result of improper segmentation, the same needs to be effectively managed by using a purpose-specific noise removal technique. Here, median filtering has been used to reduce noise. Figure 4.8 a) (page no 171) represents a binarized contour map with noise, and figure 4.8 b) (page no 171) represents a pre-processed filtered contour map.

e) **Skeletonization of contour lines:** The contour lines are then skeletonized using morphological operators known as "thinning" to reduce the contours to a single pixel width. This would tremendously help in reducing execution time by effectively managing data complexity. Here, Zhang et al [76] thinning algorithm [239] is deployed to create a skeletonized representation of contour lines. Figure 4.9 a) (page no 171) represents a finely pre-processed contour map, and figure 4.9 b) (page no 171) represents its corresponding skeletonized contour map.



a) **b)**

Figure 4.8: a) Binarized contour map b) Pre-processed contour map



a) **b)**

Figure 4.9: a) Pre-processed contour map b) Skeletonized contour map

f) **Determination of Terminal Point:** Breakage points were detected along the contour lines through contour traversals, and subsequently, all the terminal points from the breakages were stored to facilitate the reconstruction process. To determine the terminal points from the preprocessed samples, a 3x3 mask, as represented in table 4.1 (page no 173), was deployed. The chosen dimension was found to be ideal for determining breakage. Considering the centre of the 3x3 mask as the coordinate for contention of breakage, the sum of the product of the values at neighbouring coordinates of the mask and the corresponding image coordinates was calculated, excluding the centre coordinate. The criteria set for confirming a contending coordinate as the terminal point was the number of neighbours being exactly equal to one. All the identified terminal points were then collected in the terminal list by storing their corresponding coordinate values.

Table 4.1:

3x3 Mask

1	1	1
1	1	1
1	1	1

g) **Fixing criteria for Reconnections:** Decisions for establishing the connection between terminal points were derived upon taking into consideration gradient and Euclidean distance. Here backtracking and gradient are used to determine the optimal terminal point pair.

i) Determining the gradients of the terminals: The gradient at each terminal point was calculated to determine the direction of flow. To achieve the same the coordinate was backtracked by a step for determining the adjacent coordinate. Consider, (x_1, y_1) is the terminal coordinate and (x_2, y_2) is the adjacent coordinate, then gradient m was estimated using equation 4.1.

$$\mathbf{m} = \frac{y_2 - y_1}{x_2 - x_1} \text{ .... } \textbf{(Equation 4.1)}$$

The gradients were then stored in the gradient list.

ii) Determining tan inverse: The arctangent value (i.e., the tan inverse of (Δy / Δx), slope) of every terminal point was maintained for lines with the same gradient and shortest Euclidean distance, as represented in equation 4.2.

$$\mathbf{tan^{-1} = arctan\ (m)}\ \ldots\ldots\ldots\ldots\ldots\ldots\ \textbf{(Equation 4.2)}$$

For lines with the same gradients and the shortest Euclidean distance, the arctangent values of the terminal points were compared. If the arctangent value matches, the terminal points are connected.

iii) Determining the Euclidean distance: For a Euclidean plane (in 2nd dimension), the distance d (p, q) between any two terminal points p and q with coordinates (p_x, p_y) and (q_x, q_y) respectively was calculated using equation 4.3.

$$\mathbf{d(p,\ q)^2 =\ (q_x\text{-}p_x)^2+(q_y\text{-}p_y)^2}\ \ldots\ldots\ldots\ldots\ldots\ldots\ \textbf{(Equation 4.3)}$$

iv) Determining optimal matches between terminal points: A terminal point was selected, and a best or optimal match that aligns in the same gradient and has the shortest distance based on the criteria specified in serial i)-iii) above (i.e., if the distance is found to be minimum and the gradient is found to be complementary (i.e. one is positive and the other negative)) was determined by executing a recursive search process as detailed in algorithm 4.1.

This step continues until the entire terminal list has been exhausted.

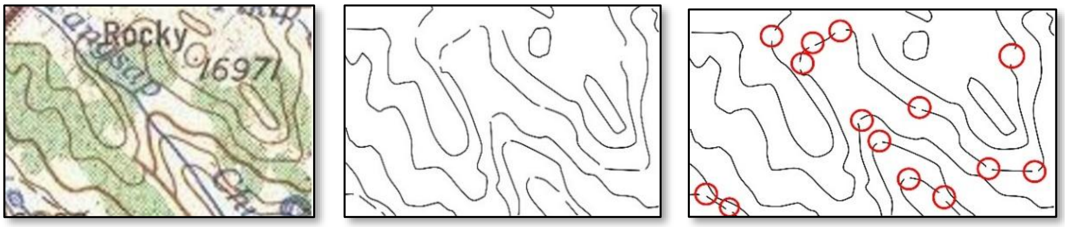
Algorithm 4.1: Algo_ optimal_pair_match

```
Consider   terminal points        Input data points
           MatchedTerminals       Output optimal points

Start      For each terminal points, i do
               distance = x ( let x be any large value)
               currentTerminal = terminals[i]
               if currentTerminal not in MatchedPoints(set), do
                   for k in terminals, do
                       workingTerminal = terminals[k]
                       if i!=k and workingTerminal not in MatchedPoints do
                           currDistance = eq. II for currentTerminal and workingTerminal.
                           if (currDistance<distance)and(product of both gradients of terminals in question= '-') do
                               distance = currDistance
                               store terminals in a list (matched)
                           End if
                           if (currDistance < distance) and (gradients of both terminals are equal) and (TanInverseValues of both terminals are equal) do
                               distance = currDistance
                               store terminals in a list(matched)
                           End if
```

```
                    End if
                End for

            End if
            Add terminals from matched to set MatchedPoints and append to a
            list (MatchedTerminals).
        End
end     for
```

Algorithm 4.1: Determining optimal matches between terminal points

Figure 4.10 a) (page no 175) represents a sample contour map with different geographic features like contours, and figure 4.10 b) (page no 175) represents its corresponding contour map with broken contours. Figure 4.10 c) (page no 175) highlights terminal points detected based on neighborhood operation.



a) b) c)

Figure 4.10 a) Sample topographic sheet [1] b) Pre-processed contour map c) Detection of terminal points

h) Establishing Connection between Terminal Points: Here, the terminal points having the shortest inter-terminal point distance are determined and joined using the concept of a modified Bezier curve. Reconnection of broken contour lines can be achieved either using knowledge of the existing neighbouring contours or simply by drawing a straight line between the optimal terminal pairs. The former approach is preferable as it maintains the integrity of contours, whereas the latter does not. For example, interpolating pixels between identified terminal pairs using the concepts of gradient direction, weighted directional angle, control points, etc. would help in maintaining the integrity of the generated contour with regards to existing contours.

The Bezier curve method uses the existing edge information for interpolating pixels [88] [89] [90], which is used for reconnecting curves. Since the curves are analogous to contours [89] [90], the concept adopted from the Bezier curve can be used to reconnect broken contours. A bezier curve is characterised by the number of control points [240] [222] [88], such as a two-point curve, a three-point curve, and a four-point curve. It uses the Bernstein polynomial, B(t), as the basis function. The Bezier curve of degree "n" is represented by:

$$\mathbf{Q(n) = \sum P_i * B(t), \text{ where, } i=1 \text{ to } n \text{ and } 0 \leq t \leq 1 \ldots\ldots\ldots\ldots \text{ (Equation 4.4)}} \text{ [240]}$$

where, B (t) is the basis function, P_i represents i^{th} control point where ‘i' varies from i= 1 to ‘n’ and ‘t’ is the parametric value that lies between 0 and 1. (Here, 0 is the starting point and 1 is the ending point.).

It is also to be noted that there are many pixels between each control point pair or line segment. Hence, to interpolate those pixels, De Casteljau's algorithm was used. This algorithm is mainly used to identify the pixel or point between any two control points. De Casteljau's technique is a recursive method for the evaluation of polynomials in Bernstein form or Bezier curves [240] [241] [222].

Furthermore, the basis function for calculating points can be expressed explicitly as follows:

$$\mathbf{B\ (t)} = \sum_{i=0}^{n} \binom{n}{i} (1-t)^{n-i} t^{i} P_{i}$$

$$= (1-t)^{n} P_{0} + \binom{n}{i} (1-t)^{n-1} t^{1} P_{1} + \cdots + \binom{n}{n-1} (1-t)^{n-1} t^{n-1} P_{n-1} + t^{n} P_{n},$$

.... (Equation 4.5) [240]

4.4.1 The Proposed refinement on Existing Bezier Curve Framework

The bezier curve is mainly used for drawing shapes, curves, edges, etc. Considering curves or edges that are similar to contours, Bezier curve techniques may be deployed for reconnecting broken contours as they use the concept of control points [240] [222, 80] [88] [89] [90] for retaining the continuity of broken contours. Through experimental runs, it was observed that the quality or contour continuity relies on the number of control points and the method used to calculate the control points [88] [89] [90]. It is implied that a very small number of control points leads to the creation of curves that do not adhere to the morphological structure of the contours, whereas a large number of contour points distorts the entire structure of the contours. Further, it was also observed that, due to effect of Bezier expansion, it is crucial to restrict the number of control points and their relative positioning for ascertaining the expected characteristics to be resembled by the contours. With regards to the above facts, in this research initiative, it was decided to limit the number of control points and suitably position them, considering gradient at terminal points and subsequent points, to facilitate the generation of integrity-preserving contours.

Upon close examination during the experimental run, it was imperially found that the ideal number of control points should be exactly 4. The experiment uses four control points to reconnect the broken contour. The first and last control points are identified as terminal points to be joined. The relative positions of the second and third points are calculated after extending the terminal points with the help of the slope values. The second control point (newControlPointsX, newControlPointsY) is computed using the following:

newControlPointsX= firstTerminalsXValue + d * firstTerminalSlopeX

………. (Equation 4.6)

newControlPointsY= firstTerminalsYValue +d * firstTerminalSlopeY

………. (Equation 4.7)

Where, d is a constant value.

The same procedure is adopted for calculating the third control point. The following is the framework implemented for reconnecting broken contours as represented in algorithm 4.2.

Algorithm 4.2: Algo_Contour_Reconnection

Consider	*deCasteljau()*	*Returns (x,y) for plotting*
	(x, y)	*Coordinate location*
	numSteps	*No. of points to be drawn*
	t	*Temporary variable*
Start	*drawTheCurve():*	

```
        numSteps = x ( let x be any large value), tobeDrawn = set()
        for k in range(numSteps):
            t = float(k) / (numSteps - 1)
            x = (deCasteljau(coorArrX, 0, n - 1, t))
            y = (deCasteljau(coorArrY, 0, n - 1, t))
            tobeDrawn.add((x,y))
    def deCasteljau(coorArr, i, j, t):
        if j == 0:
            return coorArr[i]
        return deCasteljau(coorArr, i, j - 1, t) * (1 - t) +
        deCasteljau(coorArr, i + 1, j - 1, t) * t
End
```

Algorithm 4.2: Reconnecting broken contours based on control point

The variable ‘to_be_Drawn’ contain all the points for joining the terminal points. The variable ‘coorArrX’ contains ‘x’ values of the two terminals and one control point. The variable ‘coorArrY’ contains ‘y’ values of the two terminals and one control point.

4.5 Considerations, Constraints, Development Environment

4.5.1 Considerations

The input reference map for the proposed work was taken from the online open library provided by University of Texas, Austin. The scale of representation for the map was 1:250000. A sizable portion of 100 x 100, 24 samples were created from the topographic sheet and set as input.

4.5.2 Constraints

The following are the constraints on implementing the research initiatives:

- The feature of interest should hold the properties of connected components.
- The sample image should be binarized image and free from noise (particularly salt-and-pepper noise).
- The feature of interest should be skeletonized to create a single-pixel-width representation for ease of complexity management and should ensure m-connectivity between any two pixels in the neighborhood.
- The proposed methodology is currently capable of efficiently handling datasets of size 100 x 100 and can be further extended to work with variable-sized data.
- The implementation is accomplished in two phases. In the first phase, all pre-processing, including the identification of the optimal terminal point pair is achieved and in second phase, all optimal points are connected.

4.5.3 Development Environment

The description of the development environment for the proposed research initiative is as detailed below,

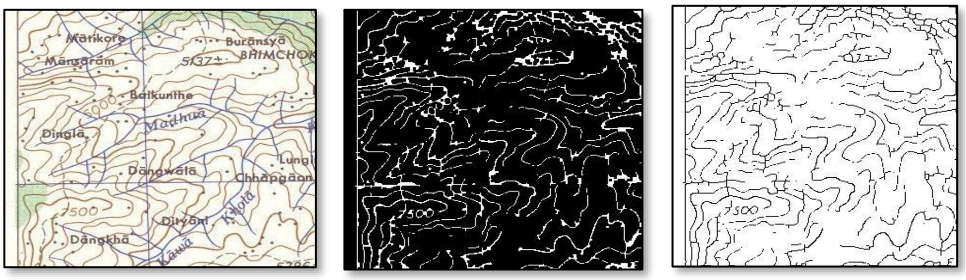
Experimental Configuration	**Description**	**Criteria for Selection**
• Processor	11th Gen Intel(R) Core i7-1165G7 @ 2.80GHz	• Facilitates faster execution of programming code
• RAM	16.0 GB (15.8 GB usable)	• Sustain applications requirements
• Operating System	Windows, 64-bit	• User friendly interface

• Development Software	MATLAB (2020a)	• Compatibility with application • Easy translation of concepts to executables • Rich with in-build libraries • Debugging ease • Scalable
	Python 3.9.10	• Open-source programming language • Rich with in-build libraries • Optimized running time complexity

4.6 Results and Discussions

Contour lines in the topographic sheet are represented with shades of brown. Figure 4.11 a) (page no 182) represents a sample topographic sheet containing contours. For computational ease, the image was converted into an HSV (Hue-Saturation-Value) model. The advantage of converting an image into an HSV image is that it reduces the aliasing effect. The brown contours were extracted using H (hue) values from 0.0 to 0.3 and V (value) values from 0.0 to 0.7, as shown in figure 4.11 b) (page no 182). The appearance of thick contours and noise is clearly visible in the resultant image. The noises were removed using a median filter, and the contours were thinned using the Zhang-Suen thinning algorithm. The sample pre-processed image so obtained is shown in figure 4.11 c) (page no 182).

The result of the thinning process is a set of contour lines represented using singly connected pixels. It was observed that contours that appear continuous in the image may also be broken. Some contours may be broken with smaller gaps, whereas others may be broken with considerably larger gaps. The smaller gaps between broken contours can be easily filled by applying a closing morphological operation. However, reconnecting contours with larger gaps is a difficult task.



a) b) c)

Figure 4.11: a) Sample Topographic Sheet [1] b) Binarized version c) Thinned Image

The proposed work established connectivity between broken contours using the concept of control points. Initially, the terminal points were identified, and then the gradients of the terminal points were calculated using backtracking. Based on the sign of the gradient and Euclidean distance, terminal points were matched. Finally, the matched terminal points were connected using the modified Bezier curve function (with the number of control points set to 4) that identifies the intermediate pixel based on selected control points. Figure 4.12 (page no 183) is the raster representation of the contour map extracted from the scanned topographic sheet, which contains many broken contours. It also highlights the different sections of broken contours that are reconnected using the aforementioned technique.

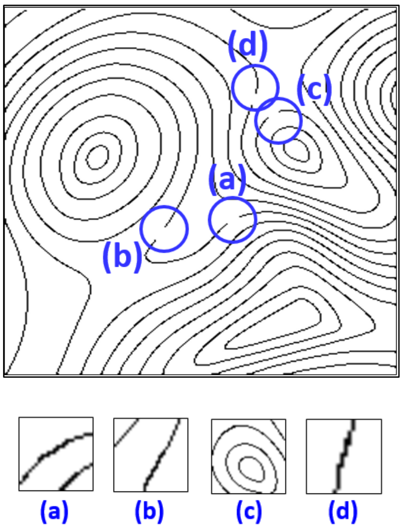
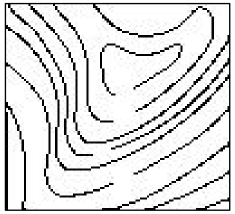
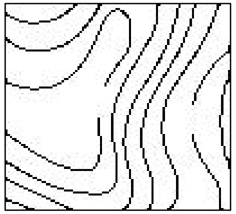
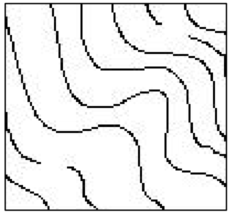
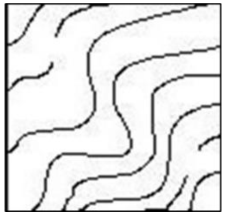
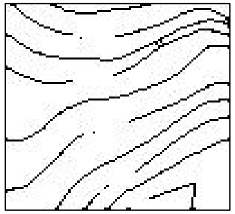
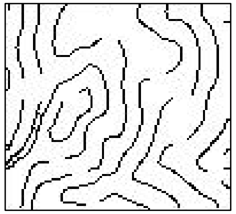
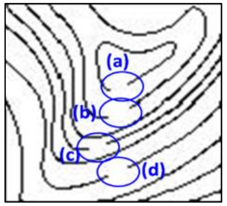
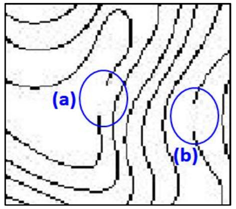
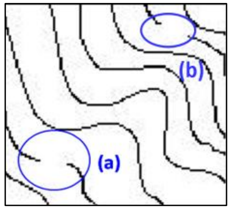
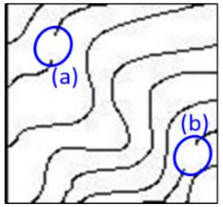
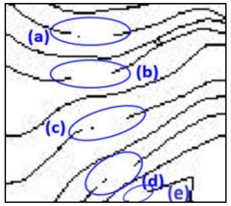
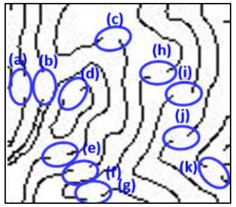
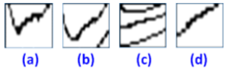
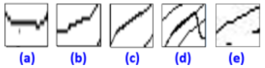
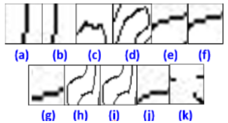
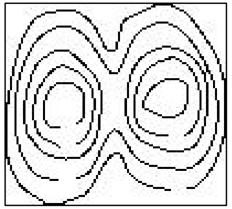
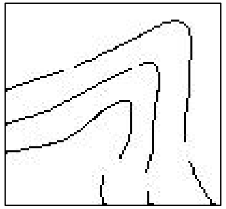
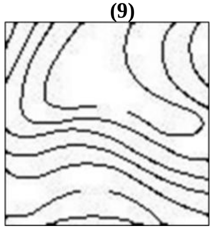
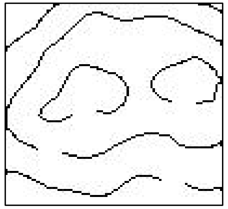
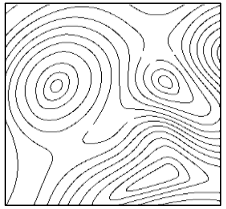
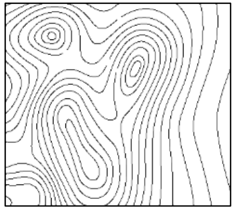
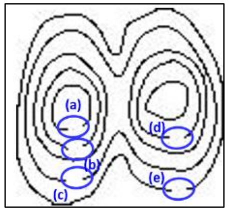
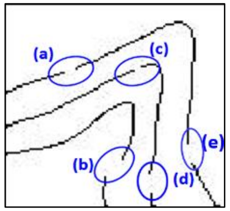
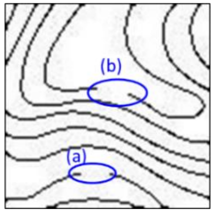
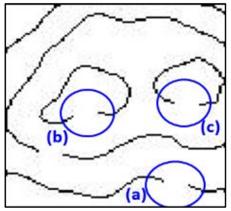
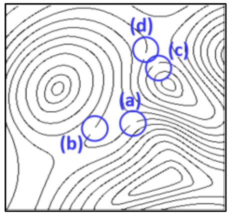
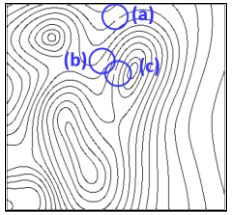
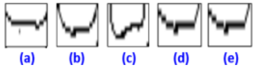
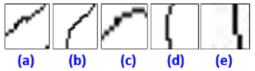
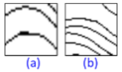
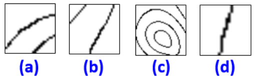
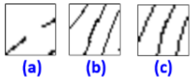


Figure 4.12: A representation of contour map and its connected broken contour

The applied technique was tested on a variety of broken contours, and the results were found to be acceptable. Table 4.2 (page no 184) highlights the various test cases containing broken contour points with demarcation of breakages and corresponding output image segments generated through processing that reflect the status of broken segments (either connected, not connected, or wrongly connected) highlighted in SS (Segment Status). Here, for simplicity, total gaps, correctly filled gaps, wrongly filled gaps, unfilled gaps, and accuracy are represented using, ^, *, $, # and % respectively, and highlighted in the OP (output section). 24 images with 95 breakages were considered to test the effectiveness of the proposed technique. A total of 83 breakages were accurately reconnected, 12 breakages were not connected, and 1 breakage was miss-connected, with 86.31% accuracy and an error rate of 13.68%. The detailed performance of an experiment is given in table 4.3 (page no 186). The same is graphically depicted in figure 4.13 (page no 188).

Table 4.2: Sample Test Cases

	(1)	(2)	(3)	(4)	(5)	(6)
Samples						
Breakage detected						
CS						
OP	$4^{\wedge}$ 4^{*} $0^{\$}$ $0^{\#}$ $100^{\%}$	$2^{\wedge}$ 2^{*} $0^{\$}$ $0^{\#}$ $100^{\%}$	$2^{\wedge}$ 1^{*} $0^{\$}$ $1^{\#}$ $50^{\%}$	$2^{\wedge}$ 2^{*} $0^{\$}$ $0^{\#}$ $100^{\%}$	$5^{\wedge}$ 3^{*} $1^{\$}$ $1^{\#}$ $60^{\%}$	$11^{\wedge}$ 10^{*} $0^{\$}$ $1^{\#}$ $60^{\%}$

	(7)	(8)	(9)	(10)	(11)	(12)
Samples						
Breakage detected						
CS						
OP	$5^{\wedge}$ 5^{*} $0^{\$}$ $0^{\#}$ $100^{\%}$	$5^{\wedge}$ 5^{*} $0^{\$}$ $0^{\#}$ $100^{\%}$	$2^{\wedge}$ 2^{*} $0^{\$}$ $0^{\#}$ $100^{\%}$	$3^{\wedge}$ 3^{*} $0^{\$}$ $0^{\#}$ $100^{\%}$	$4^{\wedge}$ 4^{*} $0^{\$}$ $0^{\#}$ $100^{\%}$	$3^{\wedge}$ 2^{*} $0^{\$}$ $1^{\#}$ $100^{\%}$

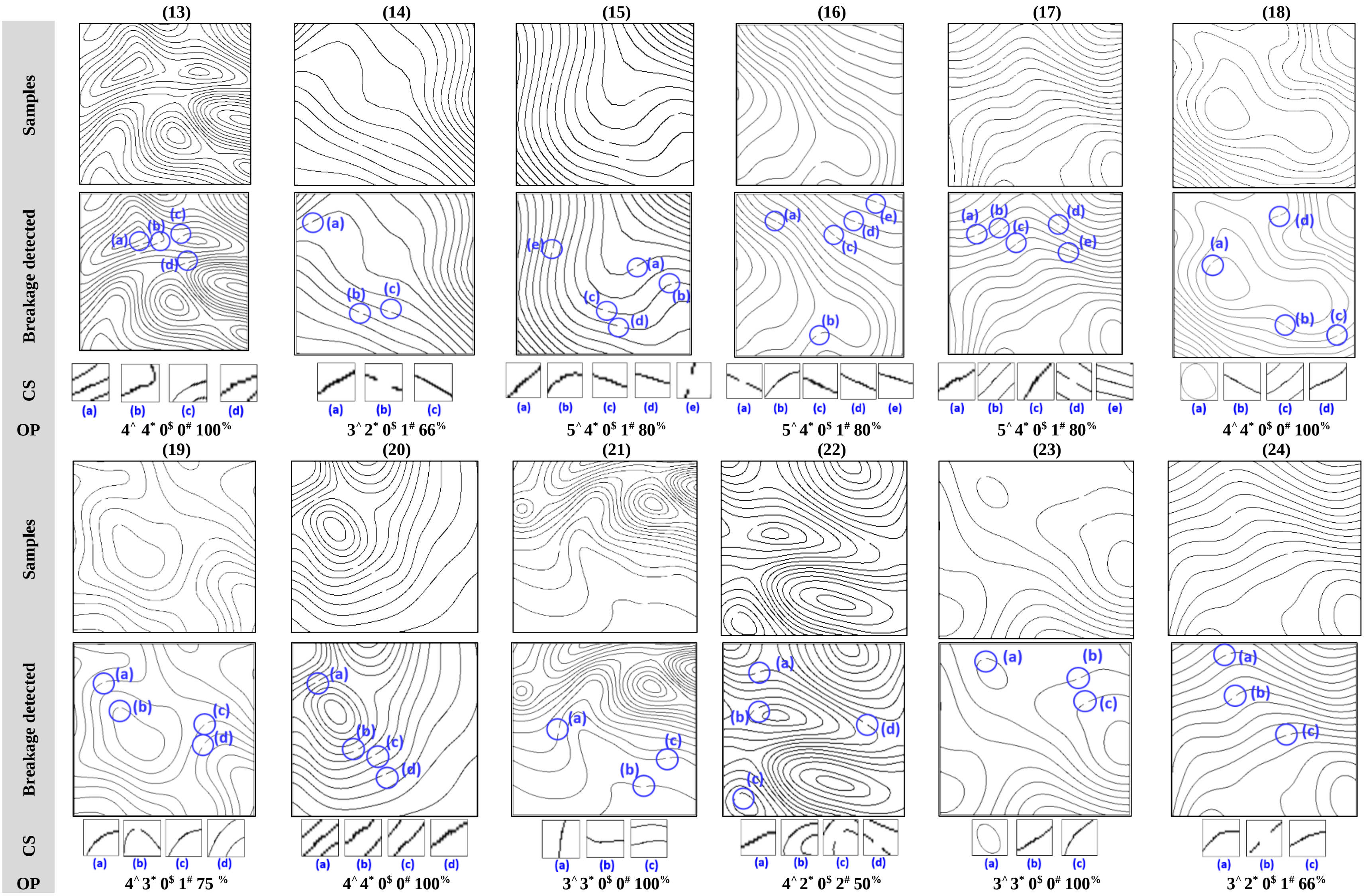

Samples
Breakage detected
CS
OP
(13)
(a) (b) (c) (d)
4^ 4* 0$ 0# 100%
(14)
(a) (b) (c)
3^ 2* 0$ 1# 66%
(15)
(a) (b) (c) (d) (e)
5^ 4* 0$ 1# 80%
(16)
(a) (b) (c) (d) (e)
5^ 4* 0$ 1# 80%
(17)
(a) (b) (c) (d) (e)
5^ 4* 0$ 1# 80%
(18)
(a) (b) (c) (d)
4^ 4* 0$ 0# 100%
(19)
(a) (b) (c) (d)
4^ 3* 0$ 1# 75 %
(20)
(a) (b) (c) (d)
4^ 4* 0$ 0# 100%
(21)
(a) (b) (c)
3^ 3* 0$ 0# 100%
(22)
(a) (b) (c) (d)
4^ 2* 0$ 2# 50%
(23)
(a) (b) (c)
3^ 3* 0$ 0# 100%
(24)
(a) (b) (c)
3^ 2* 0$ 1# 66%

Table 4.3: Performance report of proposed modified Bezier Curve framework

Sample No.	**Total Gaps (^)**	**Correctly Filled Gaps (*)**	**Wrongly Filled Gaps ($)**	**Unfilled Gaps (#)**	**Accuracy (In %)**
1	4	4	0	0	100
2	2	2	0	0	100
3	2	1	0	1	50
4	2	2	0	0	100
5	5	3	1	1	60
6	11	10	0	1	60
7	5	5	0	0	100
8	5	5	0	0	100
9	2	2	0	0	100
10	3	3	0	0	100
11	4	4	0	0	100
12	3	2	0	1	100
13	4	4	0	0	100
14	3	2	0	1	66
15	5	4	0	1	80
16	5	4	0	1	80
17	5	4	0	1	80
18	4	4	0	0	100
19	4	3	0	1	75
20	4	4	0	0	100
21	3	3	0	0	100
22	4	2	0	2	50
23	3	3	0	0	100
24	3	2	0	1	66
Total	95	82	1	12	86.31

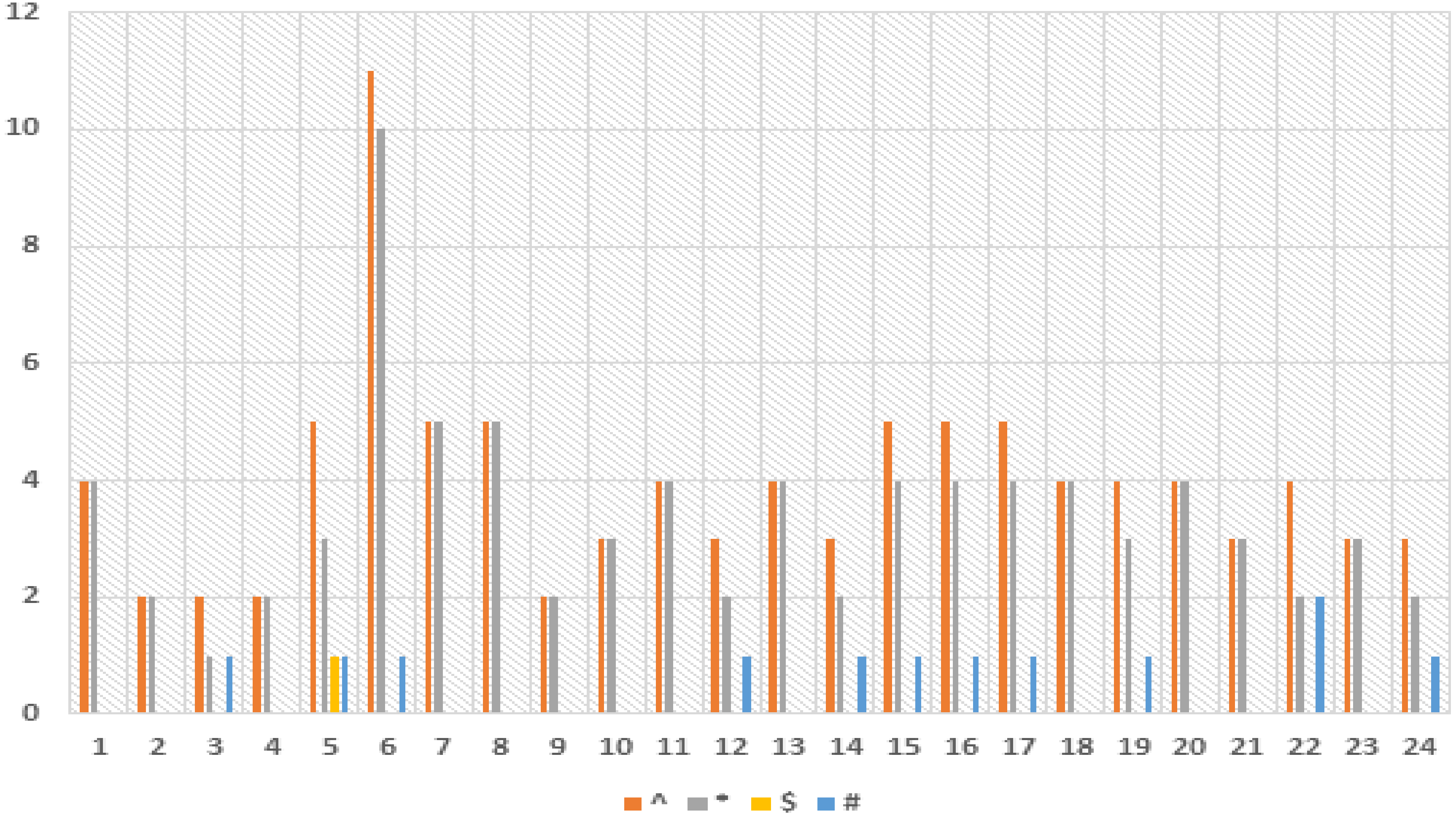


Figure 4.13: Graphical representation of Total Gaps (^), Correctly Filled Gap (*), Wrongly Filled Gap ($), Unfilled Gap (#) and Accuracy %.

4.7 Limitations of the research initiative

The proposed research initiative successfully reconstructs broken contour lines using modified Bezier curve that maintains the continuity of contour but at high computational cost. The reconnection can also be achieved by drawing a straight line for situations if breakage is less and this will optimize the overall computational cost. However, it fails to maintain the continuity of the contour. In a situation where the breakage of the contour is very large, the adopted methodology fails to identify optimal pair and reconstruct the broken contour. In addition, the computational cost is also exponentially increased as search space is increased.

4.8 Conclusion

Advancements in computational support and programming paradigms have greatly facilitated the automation of digitization techniques, enabling faster and more accurate extraction of features for use in GIS-based applications. Overcoming the lags of manual digitization in semi-automated and automated processes requires less time and helps elevate the quality of the deliverables. Such techniques may be further enriched to perform complex operations such as broken line reconnection, the creation of various thematic layers, and the creation of elevation models.

The proposed research initiative digitalizes contour maps, performs colour segmentation for extracting contours, binarizes the same, performs pre-processing (thinning and error removal), determines breakages and terminal points, fixes optimal pairs, and eventually ensures connectivity between suitable terminal pairs using modified Bezier curve drawing technique.

Drawing motivation from the previously executed initiatives, the proposed research work was able to efficiently manage computational complexity through the following inherent approaches:

a) Firstly, the feature map was skeletonized, wherein the contour lines were represented using a single-pixel width to avoid the computation of a densely pixeled feature set. This task of skeletonizing was performed during the pre-processing of the sample set and not during the actual processing time. Skeletonization also helped in determining terminal points. Further, the determination of control points and their linkage to maintain contour continuity was also facilitated.

b) Then the ideal terminal pairs were determined based on both proximity and the gradient of the terminal points, thereby, improving the chances of terminal points being reconnected to ensure contour continuity.

c) The initiative, based on empirical studies, was able to successfully control the dynamic nature of the Bezier curve for determining the ideal number of control points to be reconnected for ensuring contour continuity.

d) Eventually, to ensure that the generated contour lines-maintained integrity (resemblance to adjacent contour lines), the control points were appropriately positioned relative to adjacent contours.

The proposed technique was tested using 24 samples with 95 breakages; it could successfully establish 82 connections, 12 breakages were not connected, and 1 was wrongly connected. An accuracy of 86.31% was achieved with an error rate of 13.68%, which is appreciable. As the proposed research initiative draws its motivation from various directions provided by the previously executed works, purpose-specific advancements were made to the individual generic techniques to best fit the problem at hand. However, based on a subjective assessment (through estimated vision density characteristics), the proposed technique stands to be 86.31% accurate.

The proposed work may be further advanced to handle other linear features such as rivers, road networks, etc. Furthermore, the digitised data may assist in developing a fully automated computational system that is capable of transforming topographic sheet into a 3D model.

Chapter 5

Extraction of Associated Text, its Refinement and Attribute Generation

5.1 Introduction

The topographic sheet represents various morphological features. These features are superimposed on one another for ease of representation and containment. Visual interpretation of such representations can be done with relative ease, but on the contrary, it may be difficult for a computational system to identify the same. This calls for the conceptualization and implementation of a knowledge-based abled computational system for efficient localization and recognition. The process of selecting features of interest from a topographic sheet is termed digitization. Digitization may be done or performed using manual, semi-automated, or completely automated computational processes. With the advent of systems with enhanced computational ability, an efficient automatic knowledge-based digitization process may be conceived and realized for digitizing these features. Enabling a computation process and identifying a feature requires training the computation process with substantial and volumetric reference data. Thus, it mandates the building of reference data for each and every identifiable feature. One of the important natural features that a topographic sheet represents are the contour lines. A contour line, also known as an isoline, is formed by joining points that are at equal elevation from a particular reference level. Generally, sea level is considered as the reference. Maps that represent only the information related to contours are known as "contour maps." The two important characteristics of a contour map are the contour lines and their corresponding elevation details.

Subsequently, a DEM can be generated from the contour map by elevating the contour lines in 3D with elevation detail as the variable [242], as shown in figure 5.1 (page no 192). This 3D model may directly or indirectly contribute to many GIS applications. Hence, accurately understanding the elevation data based on which the contours will be projected in a 3D space is a critical task.

To achieve the set objectives, it is crucial to first localize the elevation value and then recognize the same. The localization of an elevation value is important as it provides information based on which the corresponding contour will be projected in 3D space and its recognition would quantify the exact value for projection. Consequently, these details may also serve as a scientific basis for the generation of intermediate elevation values (interpolated) that are not implicitly represented on the contour map. The result of localization is the approximate coordinate location represented by (x, y). This coordinate location is then mapped to the nearest contour based on the distance parameter.

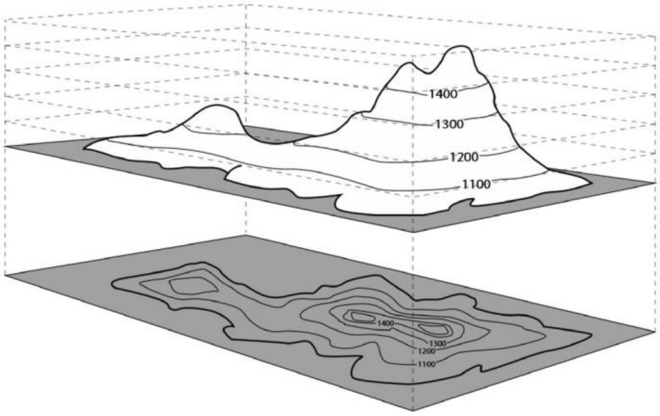


Figure 5.1: Sample DEM from contour map, Sample parts of contour map containing contours and its corresponding elevation value [243]

A topographic sheet hosts various morphological features that effectively describe the terrain. This multi-faceted information content not only elevates human perception but also provides ample direction for research initiatives. Out of all possible attributes based on utility, contours have a wide set of applications. A contour is characterized by its coordinate system and, most importantly, its elevation value. Upon successful attainment of these two attributes, creating a fully automatic 3-D projection system may be achieved with relative ease. In contrast to the traditional manual approach, this research initiative puts forward an efficient technique for localization based on both morphological as well as advanced multi-layered computational model. Thereafter, the latest multi-layered computational model enabled with augmented data obtained through various transformations has been used for recognition. Further, the extracted details can be used to project the contours in 3-D space to create DEM.

5.2 Objectives of the proposed work

The proposed research initiative objectifies methodologies for efficient localization and recognition of elevation detail associated with corresponding contours in a contour map using basic image processing techniques like image morphology as well as advanced multi-layered computational model enabled with augmented data for elevating the performance.

5.3 Applications of the proposed research initiative

DEM is crucial for various applications such as terrain modelling, hydrological modelling, and path optimization, to name a few. Automatically and accurately creating DEMs from topographic sheets could contribute a lot to many GIS applications. The efficient identification of text and elevation details aids in digitization and labelling contours and other topographic morphological features, thereby helps to create of DEM.

Here, the work has been suitably portioned into two identifiable sections dedicated for localization and recognition as detailed below in section 5.1.3 and 5.1.4 respectively.

5.4 Text Localization

Artificial intelligence-based object localization computational processes are finding increasing applications in visual recognition systems. These visual recognition systems are used for various purpose, such as auto-navigation, facial recognition systems, character recognition (hand-written and typed text) [244], robotics, speech processing [245], and biometric-based identifications, to name a few. The ability of such a process is judged based on several aspects such as computational cost, space, time, accuracy, and suitability to the application's requirements. It is not always necessary to address all possible aspects highlighted above; a trade-off can be made based on the needs of the applications. With technological advancements and the ability of the computational system to handle complex networks, the inception of various deep learning-based architectures has been proposed with varied abilities. These architectures are basically improvisations on traditional networks. Some of the notable examples include CNN, R-CNN, Fast R-CNN, and Faster R-CNN. In initiatives related to GIS and Geological Studies, TS features and related attributes act as a crucial scientific basis for drawing quality decisions related to activities such as geo-morphological planning, infrastructural developments, and studies related to the earth sciences. Each of these notable features may be considered an object of interest. These features are not readily interpretable by the computational program. In order to facilitate the same, it is essential to make the computational program gain knowledge about the aspects that distinguish one feature from another. For ease of interpretation, these features are represented using distinctive color codes, as represented in figure 5.2 (page no 195). This may be achieved by making the network learn about the aspects, enabling it to identify the features. TS is released in printed form, which needs to be scanned, converted, and stored in a computer-readable format before the features can be drawn and used for inferential analysis.

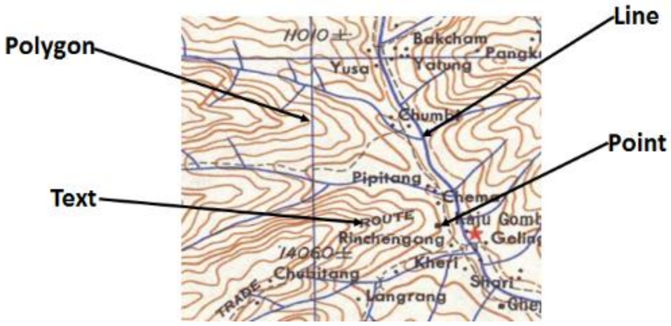


Figure 5.2: Feature Set

Feature extraction can be done manually, or a computer-aided automated digitization process may be deployed. The manual process is relatively expensive, with low confidence in the results obtained. The most reliable way is to conceptualize learning-based computational processes that generate advanced trained operators that mimic the manual process and automatically trace different features and create a repository of the same. This approach will be relatively inexpensive and generate high-quality results. But there are several issues to be addressed, as detailed below. A simple color segmentation process can be executed to extract specific features into distinctive vectors (i.e., a vector is a thematic layer representing the feature(s) of interest).

Color segmentation proves effective in situations where there are no overlapping features. In situations where the features overlap as highlighted in figure 5.3 (page no 196), segmentation often fails to deliver the expected results due to variations in intensity. Complexity intensifies when correction mechanisms are to be incorporated into the segmentation process. Blurring, aliasing, and mixing the colors of adjacent pixels further would, to a certain extent, solve the problem but lead to an increase in computational complexity.

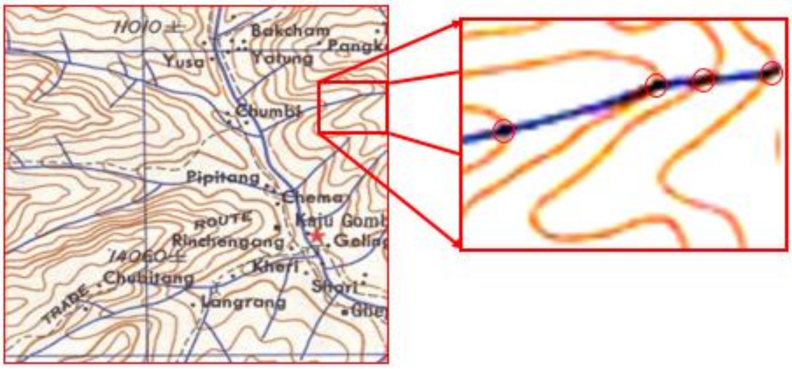


Figure 5.3: Overlapping of Feature Set

This implies a cohesive feature recognition computational process is to be relied upon that takes into consideration multiple aspects for identifying various features. With regards to the above stated, this section of the research initiative aims at localizing text present in the TS by creating a minimum bonding box that encapsulates within it the text features present in the TS using different versions of CNN and then determining their suitability for its aid in extracting text. The model was trained on a novel dataset specifically prepared for the research initiative; further data augmentation was performed to increase the dataset size. The proposed work also substantially contributes to the generation of a text feature dataset from natural images applicable to TS.

5.4.1 Proposed methodology for Text Localization using Image Morphology

Initially, the topographic sheet is scanned and saved in an image sample format. The input image is binarized and pre-processing to filter noise and protrusion. The binarization is performed using a simple thresholding operation. The noise is removed by applying median filter. The small protrusions are removed using basic morphological operations like erosion and dilation. Then, the structural operator is prepared based on the analysis of the object of interest, i.e., elevation detail. The pre-processed image is filtered with a structural operator to retain only the object of interest. Finally, the result is processed to identify the location of the elevation detail. The proposed work may be schematically represented using figure 5.4 (page no 197).

Text Localization in Topographic Sheet using Morphological Processing				
Module 1	**Module 2**	**Module 3**	**Module 4**	**Module 5**
Image interpretation and preprocessing	**Extraction of morphological Features**	**Refinement of extracted features and attributes generation**	**Localization and identification of Text**	**Generation of additional features for better visualization**

Figure 5.4: Methodology flow diagram for overall study

(i) *Image Acquisition:* At first, a paper-based topographic sheet is scanned and stored on a computer in a digital format. The contour lines and associated elevation values in the topographic sheet are represented using brown or shades of brown. An intensity-based color segmentation technique has been deployed to identify the feature pertaining to the contour lines. Here, the accuracy of the segmentation process greatly relies on the value of the threshold value set. Hence, approximation technique is used to derive the value of the threshold. It takes into account the mean value of the intensity, with the range defined by $\mu \pm \sigma$, where σ represents the deviation.

Since, the threshold value is influenced by different environment factors such as camera lens aperture setting, or object texture, so to contain the effect of these factors the image was transformed into an HSV color model. With the values of hue ranging from 0.0 to 0.3 and value ranging from 0.0 to 0.7, all brown color features were extracted.

(ii) *Image Pre–processing:* The segmented image that highlighted the contour and the associated elevation was found to has several inherent lags, such as broken features and noises. A need-based pre-processing technique was deployed to binarize, thin, filter noise, erode, dilate, and fill the holes to elevate the quality followed by location of the ROI.

Zhang-Suen thinning algorithm was applied to the binarized image to obtain image features with a single pixel width. This algorithm has been preferred over other algorithms as it does not generate branches and it also generate smooth curves.

Further, salt and pepper noise has been removed using median filter. Subsequently, a hole-filling algorithm has been applied to the thinned image to fill the features that contain holes in it. It is observed that some of the digits in the elevation detail, like 4, 0, 9, 8, 6, and intersecting features, were getting filled. Since the elevation value is represented by a mix of numbers between 0-9, there is a high probability that at least one of the above-identified characters will be used.

After a hole-filled image is generated, a morphological erosion operation is applied to it using 3x3 and 5x5 structural masks iteratively, which are selected based on the width of contour lines. The objective of this operation is to remove the contour lines, thereby retaining only the elevation value. A series of erosion operations is called for until contours are not removed. This may result in the loss of portions of elevation value. To recover the same, a series of dilation operations have been performed using large-sized squared structural operators like 5x5, 7x7, etc. masks. The objective of this operation was to grow the area covered by the elevation value to make it a single connected component. Finally, an image was obtained with many blobs, and each blob represents the elevation values in an uninterpretable form. Each blob was considered to be a single connected component. For example, if an image contains five blobs, then there are five connected components representing the five number elevation details present in the referenced map.

(iii) *Text localization:* The objective of this step is to determine the coordinate location of the elevation value. In order to validate that the blobs represent the elevation value, a background subtraction was performed. To subtract the background, the blob image was superimposed on the original binary image, and only those locations (x, y) were retained in the original binary image in which blobs were present. The following steps were followed, as shown in algorithm 5.1, for subtracting the background to validate whether the blob represents an elevation value or not.

Algorithm 5.1: Algo_ ROI_Extraction

```
Consider  Blob Image,  Input images
          Binary
          Image
          row,col      row size, column size
          i, j         temporary variables
Start          (row, col) ← Size (Binary Image)
               for i = 1 → row do
                      for j = 1 → col do
                           if Blob_Image(i, j)! = ObjectP ixel, then
                                 Binary_Image(i, j) ← 0
                           end if
                      end for
               end for
               Display 'Binary Image'
End
```

Algorithm 5.1: Region of Interest (ROI) extraction using background subtraction

After validating the identified elevation value, exact coordinate location (x, y) was obtained with the help of connected component concepts. Initially, a labelling algorithm was used to label the different connected components of the blobbed image. Based on the label of connected components, the centroid of a blob was determined. The centroid of each connected component is calculated as shown in equation 5.1:

$$\bar{X} = \left[\frac{\sum_{i=1}^{n} X_i}{n}\right], \bar{Y} = \left[\frac{\sum_{i=1}^{n} Y_i}{n}\right]$$ **………………. (Equation 5.1)**

where (X_i, Y_i), represents a significant pixel, 'n' represents the total number of significant pixels for specific connected components, and $(\bar{X}, \bar{Y})$ represents the centroid of the connected component.

The following are the steps followed, as shown in algorithm 5.2, to identify the location of the elevation values.

Algorithm 5.2 Algo_ Localization_of_elevation_value_or_text_feature

Consider	*binary_image*	Input images
	row,col	*row size, column size*
	i, j, L, num, count, x_value, y_value	*temporary variables*
	x_{mean} , y_{mean}	*Location of feature*

```
Start    Read the "binary_image"
         Extract size, [row, col] = size (ip)
         Label the connected components,  [L, num] = bwlabel
         (binary_image, 8), where L=Labelled image  and num= no. of
         connected components

         while num != 0 do
                 Temp=num
                 for i=1 to row do
                   for j=1 to col do
                         if L (i, j) = = Temp do
```

```
                        count = count+1;
                        x_value = x_value + j;
                        y_value = y_value + i;
                    End if
                End for
            End for
        End while
        Calculate the Centriod,
                $x_{mean}$ = x_value/count;
End             $y_{mean}$= y_value/count;
          Display the centroid in image, ($x_{mean}$, $y_{mean}$)
```

Algorithm 5.2: Localization of elevation value or text feature

5.4.2 Proposed methodology for Text Localization using Deep Learning technique

CNN [246] [118] has evolved from the traditional neural network system (as shown in figure 5.5, (page no 203)) wherein multiple hidden layers are replaced by multiple convolutional layers organized in a feed-forward structure enabled with intensive deep learning. Convolution is done through several purpose-specific filters of designated sizes that pan through the images, enhancing attributes of interest. These filters may be of variable sizes to inculcate a greater number of aspects of interest. The initial convolution layers may start with the simplest of the features, such as edges and basic geometric shapes, and with the progression of the network, more complex features may be tracked. With every progression, different aspects may be looked at by adjusting the filters in the convolution layers. It can also be used for efficiently handling morphological disturbances of objects [247] [248]. This implies that the ability of the network greatly relies on the effectiveness of the convolution layers.

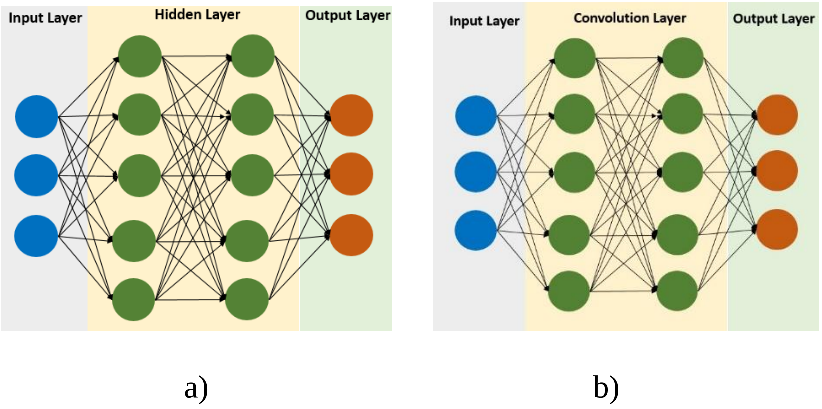


Figure 5.5: a) Traditional Neural Network b) Convolutional Neural Network

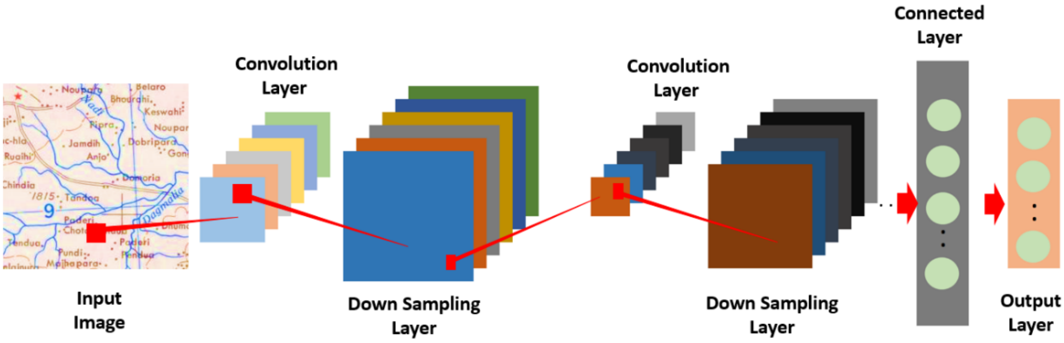


Figure 5.6: Architecture of CNN [117]

As a consequence of deep architecture as shown in figure 5.6 (page no 203), optimizing ability and learning capacity CNN is extensively used in the domain of digital image processing for applications such as object recognition, feature extraction and reconstruction to name a few [249] [250].

This research initiative intends to investigate three different variations of CNN for localizing text attribute in the TS name R-CNN, Fast-R-CNN and Faster R-CNN and judge their efficiency and effectiveness.

i) Region-Based Convolution Neural Network (R-CNN)

This approach enhances the ability of CNN by introducing regional proposals. This selective greedy [148] based approach starts with the selection of small regions that are of arbitrary dimensions for ease of localization [251]. The architecture of RCNN is shown in figure 5.7(page no 204). The effectiveness of the process also depends on how effectively these regions are selected. This step is followed by a bottom-up approach where, over a series of iterations, these regions are merged to create a cohesive region based on similarity defined in terms of texture and size. To increase efficiency, the entire process is iterated, taking samples from all possible regions into consideration. Eventually, the execution terminates with the creation of a single region. Preference should be given to the merger of smaller regions instead of letting a bigger region engulf all other smaller regions. Its ability to learn, aided by a hierarchical structure, enables the generation of effective regional proposals. The steps involved in localization of text features in topographic sheet using a R-CNN is represented in figure 5.8 (page no 205).

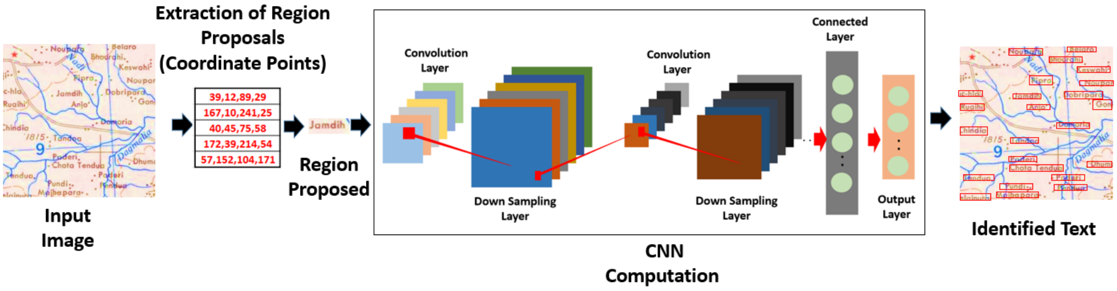


Figure 5.7: Architecture of R-CNN [117]

Text Localization in Topographic Sheet using R-CNN

Step 1	Step 2	Step 3	Step 4	Step 5

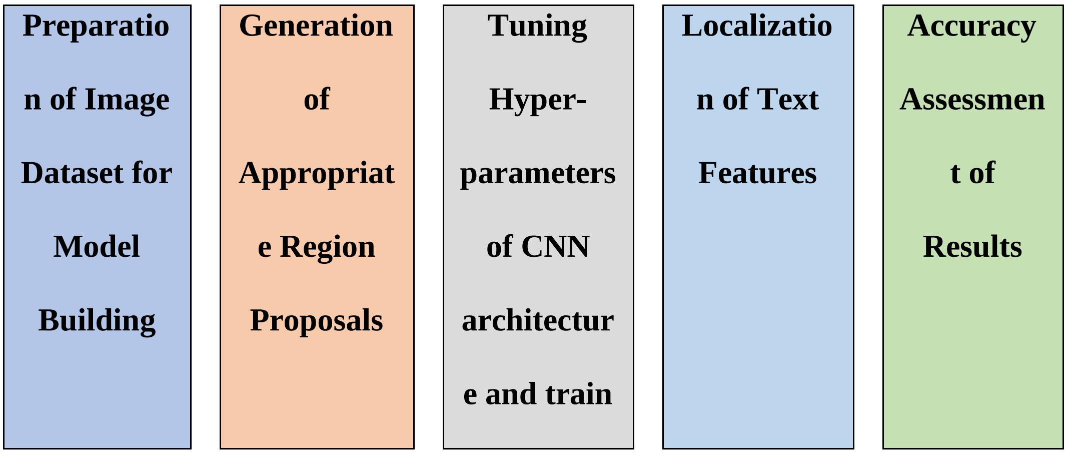

Preparation of Image Dataset for Model Building
Generation of Appropriate Region Proposals
Tuning Hyper-parameters of CNN architecture and train
Localization of Text Features
Accuracy Assessment of Results

Figure 5.8: Text Localization in Topographic Sheet using R-CNN

Once the input image is read in step 1, in step 2, a selective search mechanism is adopted in order to build a pool of region proposals for the input image. Its ability to group the region based on the suitability of the classification process enhances its capability. Grouping is performed based on similarity. The similarity here is in context with texture. The function of similarity can be expressed as:

Similarity (region 1, region 2) =Texture (region 1, region 2) +Size (region 1, region 2) [252]

………………. (Equation 5.2)

This not only allows for the generation of suitable candidates but also reduces the execution time by reducing the search space for localization. In step 3, the region proposals are aligned into fixed resolutions. This is performed to create a suitable, attributed feature representation with the region proposals. In step 4, after determining the confidence of individual region proposals, with the aid of bounding box regression and greedy non-maximal suppression, a suitable bounding box is created that encapsulates the text appropriately. In situations where the proposals are inadequate, R-CNN, instead of opting for an unsupervised approach, adopts a supervised approach for pre-training to refine the proposals [121].

Although R-CNN enhances the ability of the traditional approach, it suffers from several inherent lags. Firstly, it has a pipelined architecture with multiple stages, and without sharing computation, it executes the convolution network for each of the proposals, resulting in an increased time requirement. Although the selective search approach generates proposals at greater speed, it proves ineffective if the proposals are redundant and do not involve any learning, which may in turn result in the selection of inappropriate region proposals. The presence of a large number of regional proposals leads to an increased space requirement. Subsequently, increasing the training time as well as the testing time leads to a slower localization rate.

ii) Fast Region-Based Convolution Neural Network (Fast R-CNN)

Fast R-CNN has evolved from R-CNN with enhanced object localization capabilities like single-stage training, faster localization, and the non-requirement of storage space for features. Contrary to R-CNN, the input image itself is sent to the CNN instead of the region proposals for generating the convolutional feature map. A convolution feature map is generated by sending the input image over several convolutions and pooling layers. Region proposals are then extracted from the Convolutional Feature Map and subsequently wrapped in squares. Eventually, the RoI pooling layers adjust the region proposals into fixed sizes, and later they are sent to the Fully Connected (FC) layers, resulting in two different output layers. The first output layer is the probability distribution (per RoI), which is discrete in nature, and the second output layer is the bounding-box regression offsets. To improve the quality of the region proposals, softmax layers are used for determining the class for region proposals and to estimate the offset value for the bounding box. The architecture of fast RCNN is presented in figure 5.9 (page no 207).

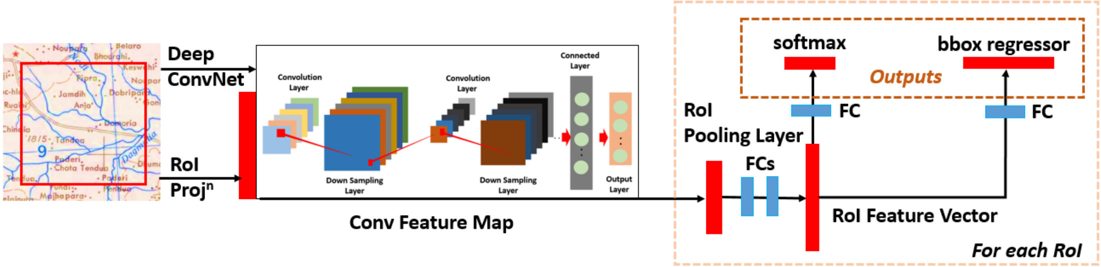


Figure 5.9: Architecture of Fast R-CNN [126]

In comparison to R-CNN, this technique takes reasonably less execution time. This may be a consequence of the following regions: First, R-CNN performs several convolution operations (each for a region proposal), whereas Fast R-CNN performs only one convolution operation. Second, it does not require said number of region proposals and instead itself generates region proposals from the Convolutional Feature Map. Third, fast R-CNN takes considerably less time in training and testing as well.

iii) Faster Region-Based Convolution Neural Network (Faster R-CNN)

Faster R-CNN incorporates a fast neural network in its architecture in figure 5.10 (page no 208), which proves effective in managing the increased time requirement of search techniques present in R-CNN and Fast R-CNN. The architecture has two different components. The first component is a fully CNN-based detector for proposing regions, and the second component is a fast R-CNN-based detector that makes use of the proposed regions for detecting the object.

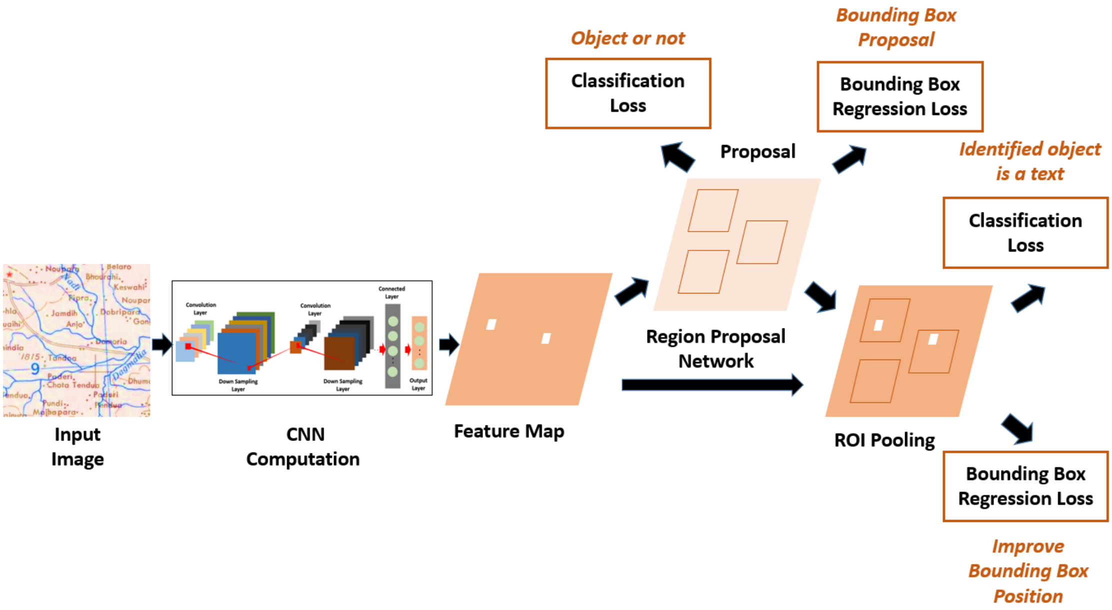


Figure 5.10: Architecture of Faster R-CNN [121]

It introduces a RPN in its architecture. This RPN is placed immediately after the CNN. RPN maps the output of the eventual layer of CNN to a feature map. It creates region proposals as output with a certain objectness score after taking an image as an input. This objectness score is a soft-max probability that determines whether a box qualifies to be a region proposal or not based on a certain threshold value.

The region proposal is created by navigating a sliding window over the feature map based on an arbitrary fixed-ratio anchor. Eventually, the region proposals are supplied to the ROI pooling layer, followed by classification and bounding box regression [121].

5.4.3 Proposed methodology for Text Localization using Faster R-CNN with resnet-50/101-feature pyramid network

Dataset for text detection was curated by cropping 255x255 resolution 200 images out of topographic sheets available in digital format. Along with images the dataset contains annotations of texts on an image i.e., the minimum and maximum (x, y) coordinates that describe position of texts in the image which follows the standard Common Objects in Context (COCO) dataset format for detection. Texts annotated included numbers and words. Biasness in dataset is inherently not present since we just have one class to predict i.e., texts and no text is left unannotated.

The detection model trained is a Faster R-CNN [121] with resnet-50/101-feature pyramid network(R-50-FPN/R-101-FPN) [122] [253] as the backbone of the detection model i.e., as the feature extractor. The entire model as a whole was adopted from Detectron2 with the backbone of the network being pretrained on Imagenet Database [254] with over 1000 object classes including texts. Figure 5.11 (page no 210) is a high-level pipeline of the detector model. Since the backbone is already pretrained, hence just fine tuning it with the dataset curated was enough. Fine tuning the feature extractors along with the object classification and bounding box regression ends was part of the training. The alternate training method of sharing CNN weights between the RPN and the detector network [121] is taken care by the Detectron2 framework. The model is trained with 180 images and validated with 20 images. Table 5.1 (page no 210) displays the network and hyper parameters adjusted and experimented to get maximum average precision (AP) on the validation set or otherwise avoid overfitting of data. Optimizer used was Stochastic Gradient Descent (SGD), Cross Entropy Loss for classifier and Smooth L1 Loss for the bounding box regression as loss functions.

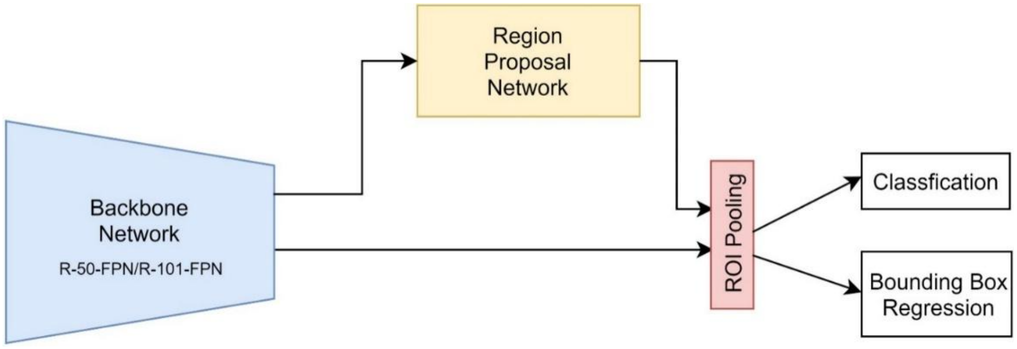


Figure 5.11: Detector Pipeline

Table 5.1: Detector Network and Hyper-parameters

Network	Backbone	Learning rate	Epochs	Region Proposals
Faster RCNN	R-50-FPN	0.001	1500	256
Faster RCNN	R-101-FPN	0.001	1500	256

5.5 Considerations, Constraints, Development Environment

5.5.1 Considerations, Constraints, Development Environment for Text Localization using Image Morphology

i) Considerations

The input reference map for the proposed work was taken from the University of Texas, Austin. The scale of representation for the map was 1:250000. A sizable portion of a 300 x 300 subset consisting of 30 images was created from the reference map and set as input. The sample input image type is JPG (Joint Photographic Experts Group).

ii) Constraints

The following are the constraints on implementing the research initiatives:

- The proposed methodology is currently capable of efficiently handling datasets of size 300 x 300 and can be further extended to work with variable-sized data.
- The sample image is converted to an HSV image to mitigate environmental lightening issues that may arise during the scanning of topographic sheets.

iii) Development Environment

The description of the development environment for the proposed research initiative is as detailed below:

Experimental Configuration	**Description**	**Criteria for Selection**
• Processor	11th Gen Intel(R) Core i7-1165G7 @ 2.80GHz	• Facilitates faster execution of programming code
• RAM	16.0 GB (15.8 GB usable)	• Sustain applications requirements
• Operating System	Windows, 64-bit	• User friendly interface • Compatibility with application
• Development Software	MATLAB (2020a)	• Easy translation of concepts to executables • Debugging ease • Scalable • Rich with in-build libraries

5.5.2 Considerations, Constraints, Development Environment for Text Localization using Deep Learning technique

i) Considerations

The input reference map for the proposed work was taken from the University of Texas, Austin. The scale of representation for the map was 1:250000. A sizable portion of a 300 x 300 subset consisting of 50 images was created from the reference map and set as input. To increase the size of the dataset to implement deep learning techniques, a data augmentation was performed with varied size filters. The sample input image type is JPG.

ii) Constraints

The following are the constraints to implement the research initiatives:

- unavailability of a standard dataset to train the machine learning model. To mitigate the issue, a custom-labeled dataset was prepared.
- The trained model is limited to English alphabets aligned horizontally.
- The proposed methodology is currently capable of efficiently handling datasets of size 300 x 300 and can be further extended to work with variable-sized data.

iii) Development Environment

The description of the development environment for the proposed research initiative is as detailed below:

Experimental Configuration	**Description**	**Criteria for Selection**
• Processor	11th Gen Intel(R) Core i7-1165G7 @ 2.80GHz	• Facilitates faster execution of programming code
• RAM	16.0 GB (15.8 GB usable)	• Sustain applications requirements

• Operating System	Windows, 64-bit	• User friendly interface • Compatibility with application
• Development Software	Python 3.9.10 in Google Colab	• Open-source programming language • Rich with in-build libraries routines for implementing deep learning technique. • Provides GPU (Graphical Processing Unit)

5.5.3 Considerations, Constraints, Development Environment for Text Localization using Faster R-CNN with resnet-50/101-feature pyramid network

i) Considerations

The input reference map for the proposed work was taken from the University of Texas, Austin. The scale of representation for the map was 1:250000. A sizable portion of a 300 x 300 subset consisting of 50 images was created from the reference map and set as input. To increase the size of the dataset to implement deep learning techniques, a data augmentation was performed with varied size filters. The sample input image type is JPG.

ii) Constraints

The following are the constraints to implement the research initiatives:

- unavailability of a standard dataset to train the machine learning model. To mitigate the issue, a custom-labeled dataset was prepared.
- The trained model is limited to English alphabets aligned horizontally.

- The proposed methodology is currently capable of efficiently handling datasets of size 300 x 300 and can be further extended to work with variable-sized data.

iii) Development Environment

The description of the development environment for the proposed research initiative is as detailed below:

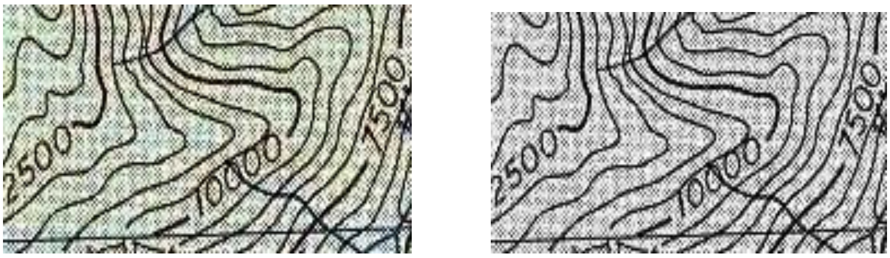
Experimental Configuration	**Description**	**Criteria for Selection**
• Processor	• Tesla K80 GPU • 11th Gen Intel(R) Core i7-1165G7 @ 2.80GHz	• Facilitates faster execution of programming code
• RAM	• 12GB NVIDIA • 16.0 GB (15.8 GB usable)	• Sustain applications requirements
• Operating System	Windows, 64-bit	• User friendly interface • Compatibility with application
• Development Software	Python 3.9.10	• Open-source programming language • Rich with in-build libraries routines for implementing deep learning technique. • Provides GPU

5.6 Results and Discussions

5.6.1 Results and Discussion for Text Localization using Image Morphology

The methods discussed earlier have been tested with different samples constructed from different contour maps and have proven to be effective by giving consistent and similar results. The following are the results obtained at different stages:

(i) RGB (Red-Green-Blue) to gray scale conversion: The property of an input and output image is an 8-bit unsigned integer format, so the pixel value will range from 0 to 255. The RGB image is converted into a gray image by averaging the three planes of the input image, as shown in figure 5.12 (page no 215).

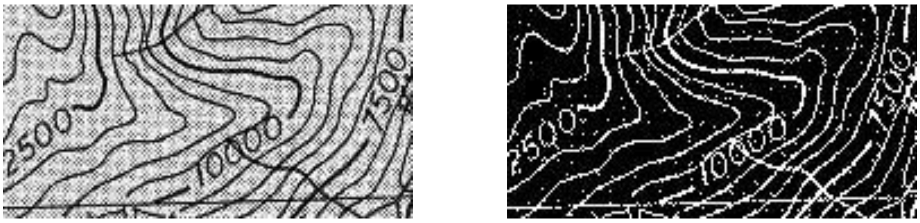


(a) Input: Color (RGB) Image. (b) Output: Gray Image.

Figure 5.12: Some samples of prepared input image.

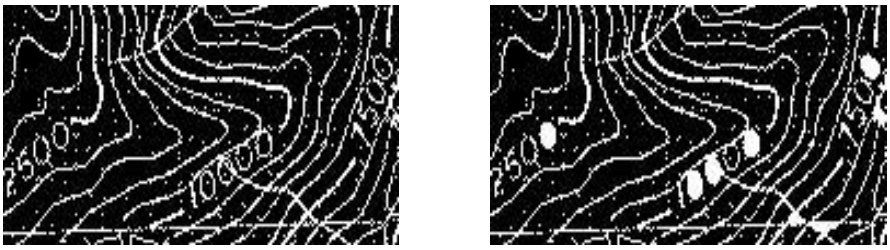
(ii) Binarization: For this, an RGB image is converted to an HSV color model. Then, the ROI was extracted based on the hue and value information of the HSV color model of the scanned topographic sheet (figure 5.13 (page no 216)). So, considering 0.0 6 hue, 60.3, and 0.0 6 value, 6 0.7, all ROI were extracted.

(iii) Hole filling: A hole is a set of connected components that cannot be reached by starting the traversal from the edge of an image. Such connected components are replaced with significant pixels, as shown in figure 5.14 (page no 216). This operation is performed in order to retain information that may be lost due to erosion.



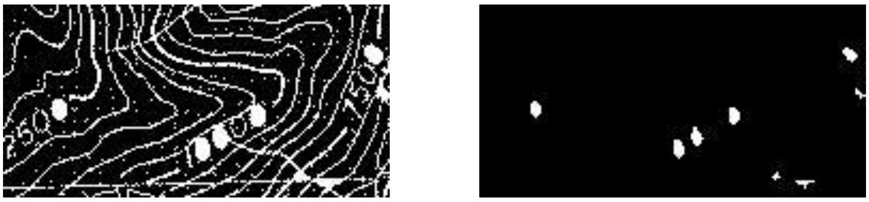
(a) Input: Gray Image. (b) Output: Binary Image.

Figure 5.13: Conversion of gray scale image to binary.



(a) Input: Binary Image. (b) Output: Filled Image.

Figure 5.14: Filling holes of the features.

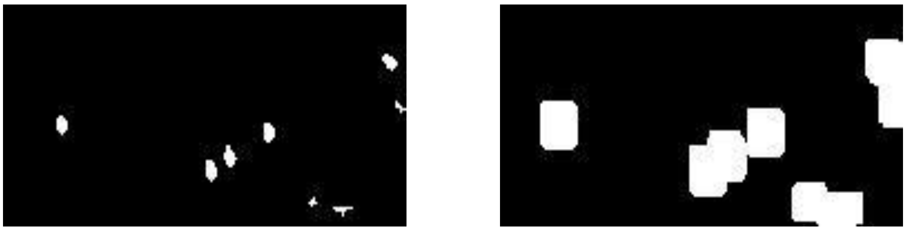


(a) Input: Filled Image. (b) Output: Eroded Image.

Figure 5.15: Result of series of erosion operation using 3x3 mask or 5x5

(iv) Erosion operation: The region of interest lies in areas where the density of significant pixels is considerably high as it represents elevation value, and the areas with lower densities of significant pixels represent contours. Such contour areas are truncated by iteratively applying 3x3 square erosion morphological operations as shown in figure 5.15 (page no 216).

(v) Dilation operation: The lost information of elevation value was recovered by iteratively using morphological dilation operations with higher-order structural operators of sizes 5x5 and 7x7 and of arbitrary shapes. The main objective of this operation is to achieve a single connected component for each elevation value, as shown in figure 5.16 (page no 217).

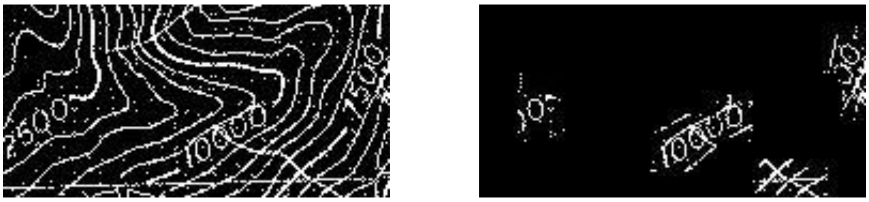


(a) Input: Eroded Image. (b) Output: DilatedImage (*or LabelledImage*).

Figure 5.16: Result of series of dilation operation using 5x5 mask

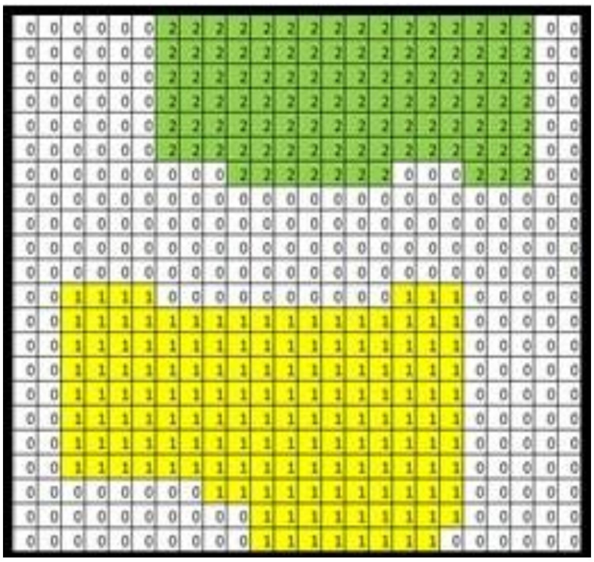
(vi) Background subtraction operation*:* It is important to ensure that the connected components represent elevation values. To validate it, the areas of connected components are not manipulated; however, the remaining areas in the input image are nullified. Sometimes during background subtraction, a few unwanted features appear (as shown in figure 5.17 (page no 218)), which are removed based on the area of the features, i.e., features with a lower pixel count are nullified.

(vii) Localization of elevation value: The location of the elevation value is represented by a pair of coordinates. A sample of labelled connected components is shown in figure 5.18 (page no 218), which contains two components labelled as 1 and 2, respectively. Hence, based on the value of the labelled connected component, the centroid is easily calculated by collecting all (x_i, y_i) and then averaging it by the number of pixels for the respective connected components.



(a) Input: Binary Image. b) Output: Background Subtracted Image.

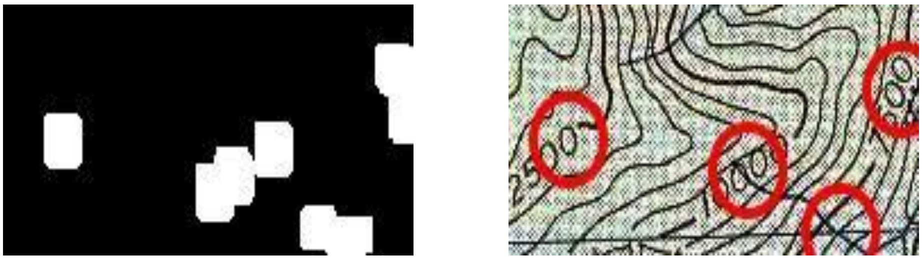
Figure 5.17: Result of background subtraction using masked image



0	0	0	0	0	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	0	0
0	0	0	0	0	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	0	0
0	0	0	0	0	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	0	0
0	0	0	0	0	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	0	0
0	0	0	0	0	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	0	0
0	0	0	0	0	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	0	0
0	0	0	0	0	0	0	0	0	2	2	2	2	2	2	2	0	0	0	2	2	2	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0

Figure 5.18: Result of background subtraction using masked image

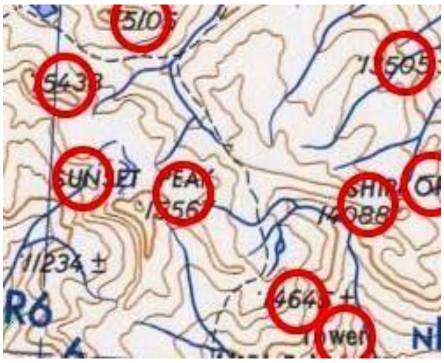
The same procedure was applied to “Labelled Image” as shown in figure 5.19 (page no 219) below:



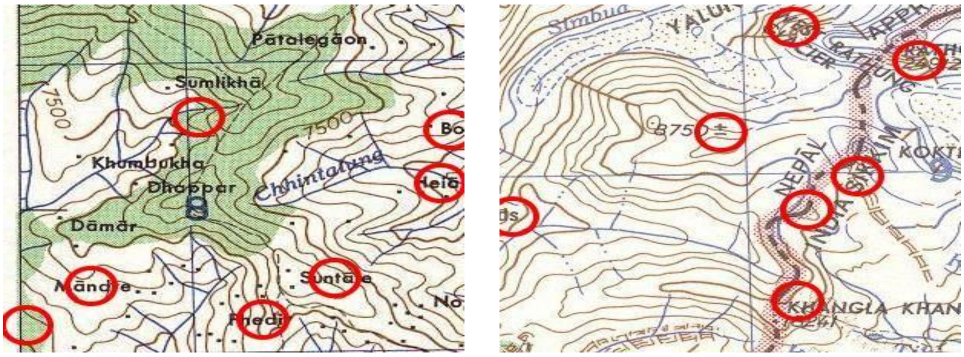
(a) Input: Labelled Image. (b) Output: Localized Image.

Figure 5.19: Result of localized text feature

The same procedure was applied randomly to sub-images of similar topographic sheets that contain many such elevation values or text features. The results obtained were considerably sound, as shown in figure 5.20 (page no 220). The identified features are encircled with red on a map. An experiment was conducted by randomly selecting 30 samples of topographic sheets from a dataset of topographic sheets. There were 100 features present in these sub-images, out of which 85 have been correctly classified. The details of the experiment are given in table 5.2 (page no 220). Furthermore, it is to be noted that the result obtained contains some misclassification, which may be due to improper segmentation arising due to the overlapping of elevation value with other morphological features. Figure 5.21 (page no 220) highlights the performance of the adopted method for localizing elevation values or text features.



(a) Sample Output 1.



(b) Sample Output 2 (c) Sample Output 3

Figure 5.20: Results obtained from different samples of TS

The localized elevation values are then associated with the nearest contour. This may be performed using any distance metric, like Euclidean distance. This association is important as it acts as a basis for projecting the contours in a 3D space.

Table 5.2: Feature extraction results for different samples of topographic sheet.

No. of features	Classified correctly	Unclassified	Misclassified
100	85	15	58

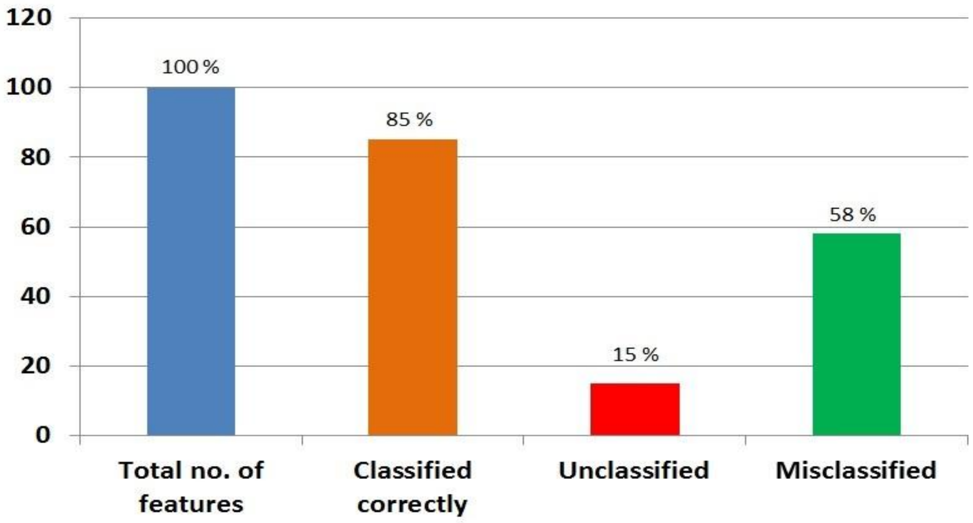


Figure 5.21: Performance for localizing the Features in TS

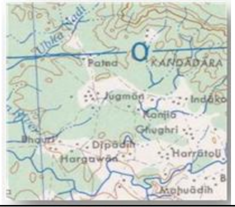
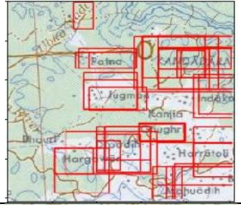
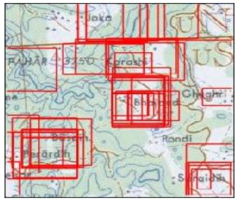
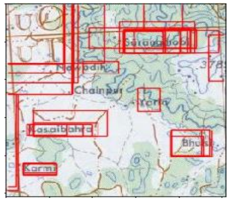
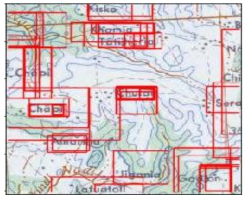
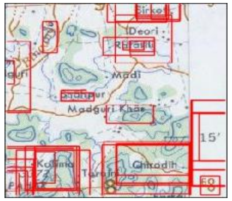
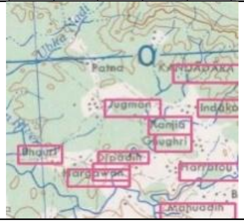
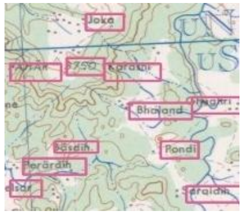
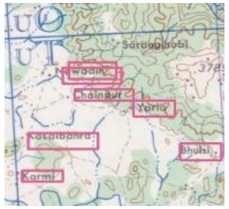
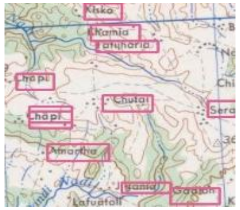
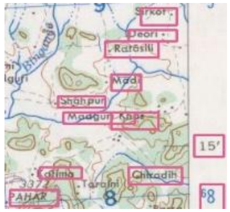
5.6.2 Results and Discussion for Text Localization using Deep Learning

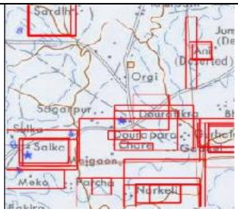
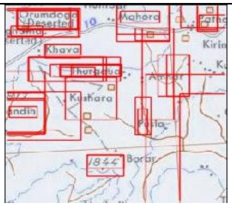
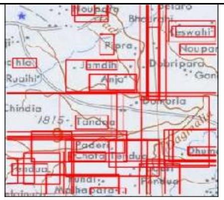
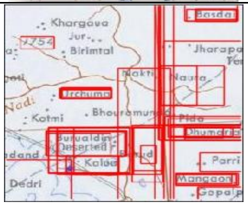
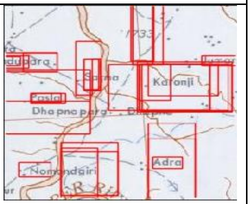
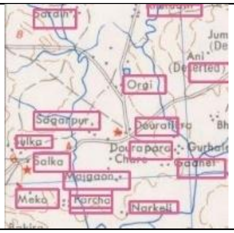
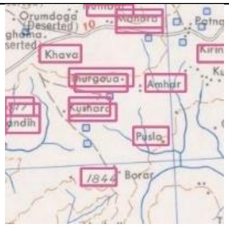
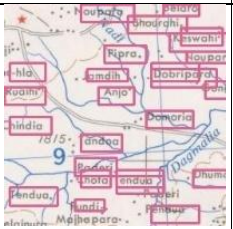
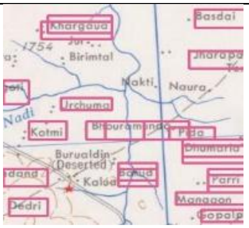
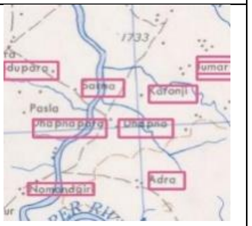
An experiment was conducted by building a model based on faster R-CNN architecture which was trained using 52 different samples of raw topographic sheets with 15-20 text features in each image. Altogether, the Faster R-CCN model was trained with 1200 text feature. To increase the size of the dataset for modeling, data augmentation was performed where filters of various sizes (3x3, 5x5, and 7x7) were used. The coefficient values of the filter determine the nature of the operation to increase the dataset size. The nature of the operation includes smoothing and sharpening. The advantage of processing is that it not only generates multiple samples, thereby increasing the dataset size, but also normalizes the noise present in the raw image. The designed model's efficiency was tested with 15 different, unknown samples of topographic sheets, each of which had 15-20 text features in all possible orientations.

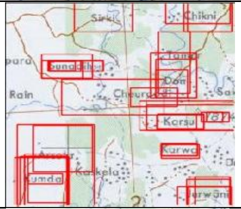
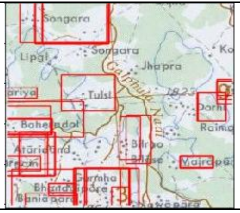
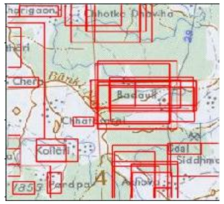
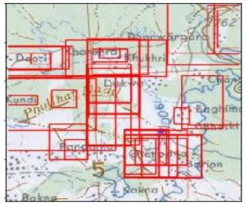
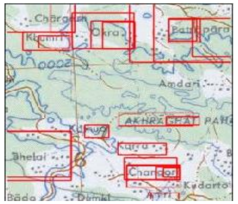
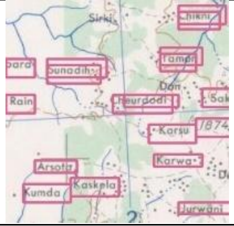
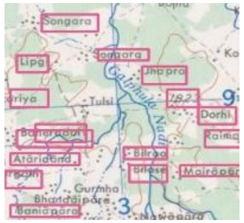
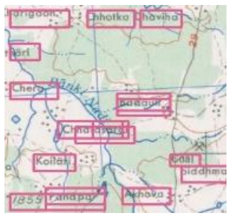
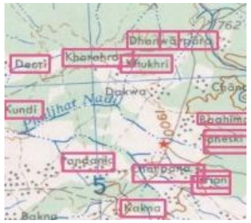
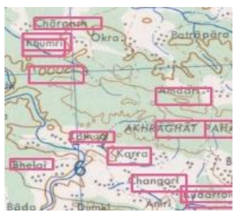
To validate the performance of Faster R-CNN, the result of the same is compared with the R-CNN model with the set of training and testing data. The details of the experiment are highlighted in table 5.3 (page no 223). It has been observed that out of 243 text features in test samples, 125 text features were correctly classified by R-CNN and 176 text features were correctly classified by Faster RCNN. This reflects that the accuracy of the faster R-CNN model is 72.42%, whereas the accuracy of R-CNN is 51.44. This performance was achieved using a limited dataset. It is to be noted that the data for training and test samples were of poor quality and from a complex environment. Furthermore, by increasing the size of training samples and applying various preprocessing techniques, the performance is expected to improve. The performance comparison between R-CNN and Faster R-CNN is shown in figure 5.22 (page no 226).

Through experimental results, it has been observed that the various CNN models effectively detect objects in an image. Out of various existing versions of the CNN model, like basic CNN, R-CNN, Fast R-CNN, and Faster R-CNN, it was observed that Faster R-CNN, amongst all the models, is the most effective model in terms of its localization capability with minimal error. In addition, it was also observed from figures 5.23 (page no 226) and 5.24 (page no 226) that the unclassified and misclassified error counts are comparatively lower in faster RCNN than in RCNN.

Table 5.3: Performance Comparison of RCNN and Faster RCNN for Text Feature Localization in Topographic Sheet

Sample Image		**1**	**2**	**3**	**4**	**5**
Input Image						
RCNN Output						
FRCNN Output						
Total Text		12	12	8	16	15
Unclassified	**RCNN**	2	6	2	6	3
	FRCNN	2	2	2	6	3
Misclassified	**RCNN**	2	3	6	4	1
	FRCNN	0	1	0	0	0
Classified Correctly	**RCNN**	10	6	6	10	12
	FRCNN	10	6	6	10	12

Sample Image		6	7	8	9	10
Input Image						
RCNN Output						
FRCNN Output						
Total Text		18	17	26	24	12
Unclassified	RCNN	9	9	15	13	5
	FRCNN	4	7	4	9	4
Misclassified	RCNN	0	2	4	0	0
	FRCNN	0	1	0	0	0
Classified Correctly	RCNN	9	8	11	11	7
	FRCNN	14	10	22	15	8

Sample Image		11	12	13	14	15
Input Image						
RCNN Output						
FRCNN Output						
Total Texts		17	21	14	15	16
Unclassified	RCNN	2	0	0	3	0
	FRCNN	0	0	0	0	1
Misclassified	RCNN	7	12	3	9	8
	FRCNN	3	5	1	4	6
Classified Correctly	RCNN	10	9	11	4	8
	FRCNN	14	15	13	11	10

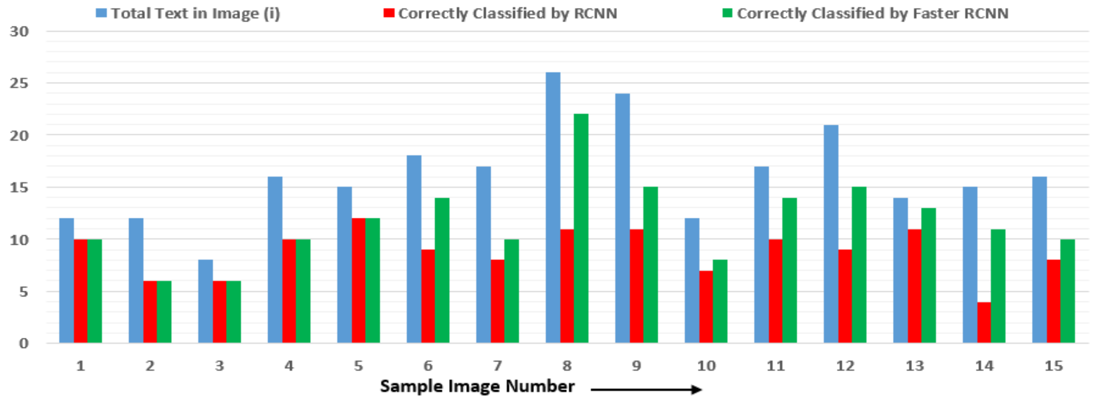


Figure 5.22: Comparative analysis for correctly classified text features using RCNN and Faster RCNN

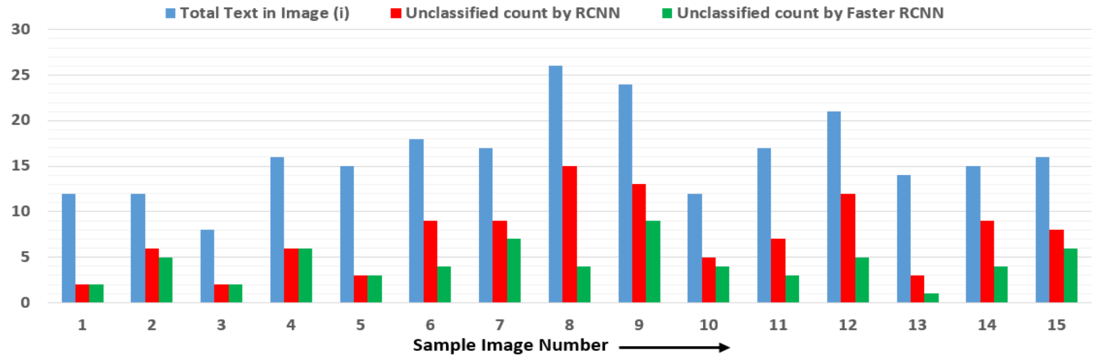


Figure 5.23: Comparative analysis for unclassified text features using RCNN and Faster RCNN

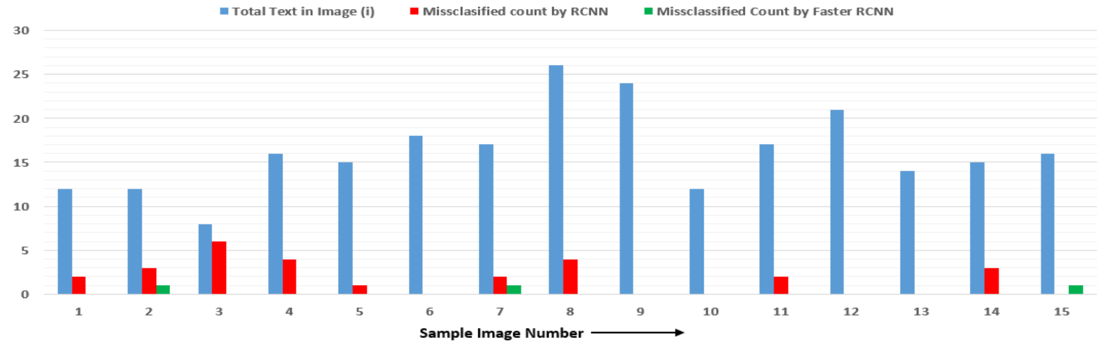


Figure 5.24: Comparative analysis for misclassified text features using RCNN and Faster RCNN

5.6.3 Results and Discussion for Text Localization using Faster R-CNN with resnet-50/101-feature pyramid network

Table 5.4 (page no 227) displays network vs metrics. The inference in table 5.4 (page no 227) is according to the COCO metric evaluation standard for detection. Other than AP on validation set, it is also relevant to calculate the total loss and false negative classification on the training set. Faster R-CNN [121] incorporates multitask loss hence total loss is the combination of loss on object classifier and bounding box regression ends. False negative metric gives the idea of how many regions in actual were texts were not classified as texts. Figures 5.25 (page no 228), 5.26 (page no 229), 5.27 (page no 229), and 5.28 (page no 229) are the total loss vs epoch, false negative score vs epoch for Faster RCNN R-50-FPN and Faster RCNN R-101-FPN networks, respectively. The oscillating value of total loss seen in figure 5.27 (page no 228) could be avoided by scheduling learning rate to decay for consequent epochs and is left for further experimentation. Training time with R-50-FPN (50 layers) took around 14 minutes and with R-101-FPN (101 layers) took around 35 minutes on a 12GB NVIDIA Tesla K80 GPU with 4 workers. Faster RCNN + R-101-FPN returns best AP value of 79.6% against IoU value >= 0.7.

Table 5.4: Average Precision (IoU >= 0.7)

Network	**Metrics**					
	AP	**AP50**	**AP75**	**APs**	**APm**	**APl**
Faster RCNN + R-50-FPN	76.042	98.947	94.076	76.124	76.300	80.00
Faster RCNN + R-101-FPN	79.609	98.998	97.890	80.300	78.767	90.00

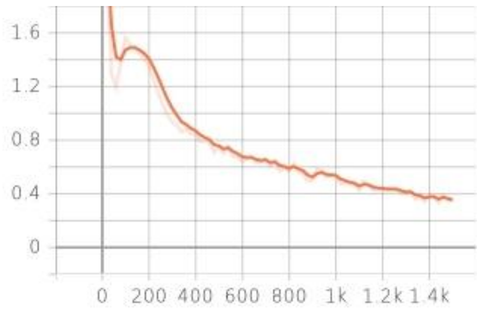


Figure 5.25: Total Loss Vs Epoch

(Faster RCNN + R-50-FPN)

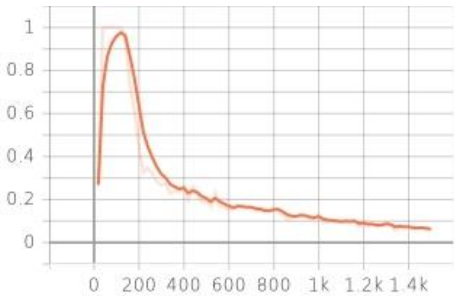


Figure 5.26: False Negative Vs

Epoch (Faster RCNN + R-50-FPN)

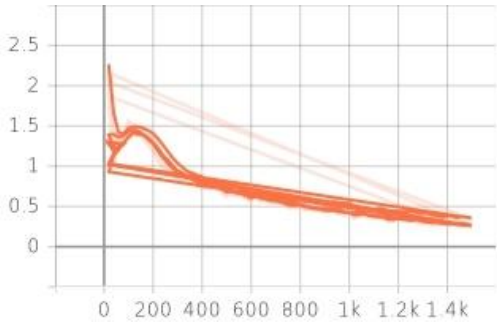


Figure 5.27: Total Loss Vs Epoch

(Faster RCNN + R-101-FPN)

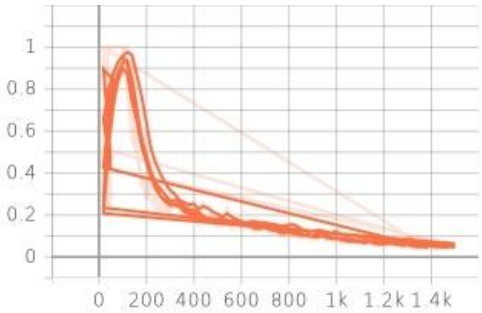


Figure 5.28: False Negative Vs

Epoch (Faster RCNN + R-101-FPN)

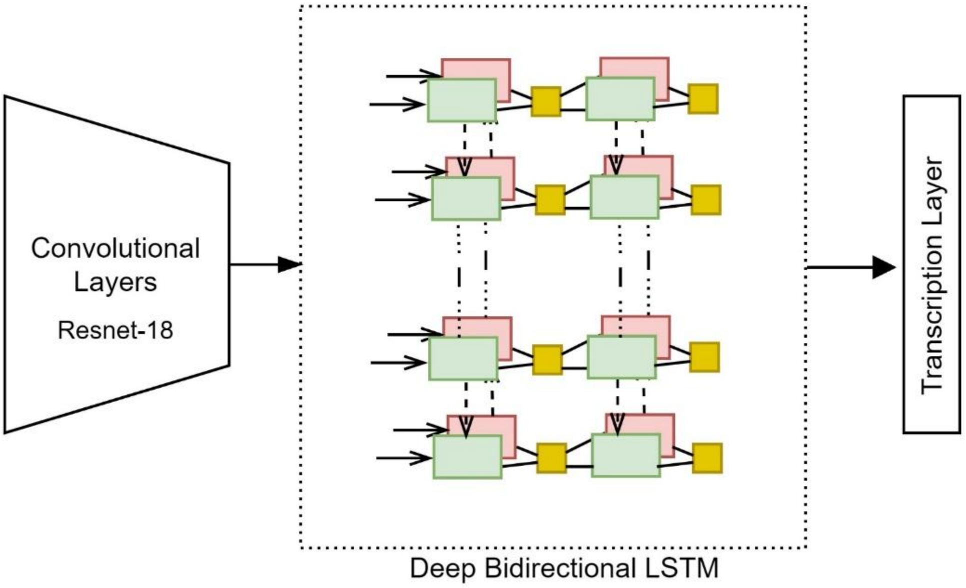


Figure 5.29: Recognizer Pipeline[121]

5.7 Text Recognition

In this research initiative, various deep learning models have been pursued and thoroughly assessed considering referenced data specifically built to serve the purpose of research objectives. The cross-sectional assessment was performed to determine the optimal recognition model best fit for the problem in hand. Further, the reference data was subjected to various transformations for increasing the volume and variety of the dataset.

5.7.1 Text Recognition using Convolutional Recurrent Neural Network (CRNN)

Dataset for text recognition was curated by cropping out regions on topographic sheet only containing text. 500 images were collected and annotated. Annotation of an image is a single word containing combination of characters.

Model of CRNN (Convolutional Recurrent Neural Network) [159] is adopted for recognizing text. Images of variable dimension are resized to 50X200 pixel. 450 images are used for training and remaining 50 for validation. A pretrained Resnet-18 [255] without classifier head is used as the feature extractor layer. The feature sequence is broken down into a set number of frames i.e., the input size of the consequent RNN layer. A forward and a backward LSTM are stacked together to form a bidirectional LSTM which supplies contexts in both directions. This layer predicts a series of labels corresponding to per frame of feature sequence input. The LSTM has 256 hidden layers which is set as a hyper parameter while training. In between the feature extractor and sequence predictor layers is a map to sequence network as described in [159] for turning feature sequence into feature maps for propagating error differentials from recurrent to convolutional layers. The top layer is a lexicon free based transcription layer that unifies per frame feature sequence prediction into a single label. The single label will be the recognition of text in image. Predicting a label sequence adopts a Connectionist Temporal Classification (CTC) loss as proposed in [256]. CTC basically maximizes the likelihood of a label sequence i.e., the ground truth conditioned over per frame predicted label. The network is designed to be end to end trainable. Error differentials in recurrent layers are calculated using Back Propagation through time (BPTT), in convolutional neural network layers using back propagation and in transcription layer using a forward backward algorithm [256]. SGD is used as the optimizer with momentum and the objective function is the CTC loss [256]. Table 5.5 (page no 230) displays the best hyper parameters experimented over multiple trainings to achieve maximum accuracy on validation set. The network is limited to English vocabulary of capital, small alphabets and a blank. Figure 5.29 (page no 228) shows the recognizer pipeline.

Table 5.5: CRNN Hyper-parameters

Learning Rate	**Epochs**	**Weight Decay**	**Momentum**	**Drop out**	**Clip Norm**	**Bi-directional**	**Hidden Size**
0.02	1000	1e-5	0.9	0.1	5	True	256

Table 5.6: CRNN Accuracy

Accuracy	
Validation set	45%
Train set	60%

Table 5.6 (page no 230) displays the train and validation set accuracy. On training the network for more than 1000 epochs the validation accuracy remained unchanged while the training loss plateaued (Figure 5.30 (page no 231)). From figure 5.31 (page no 231) the validation loss seemed to oscillate with no significant change. The cropped images from topographic sheets were low resolution and required to be resized to higher dimension for training. The dataset contains low dimensional images having few characters which might get pixelated on resizing to the extent that receptiveness of characters might get lost and may work as wrong representations which further suffers accuracy of model. Increasing training data would possibly increase training and validation accuracy. Training time was around 30 minutes on 12GB NVIDIA Tesla K80 GPU with 2 workers.

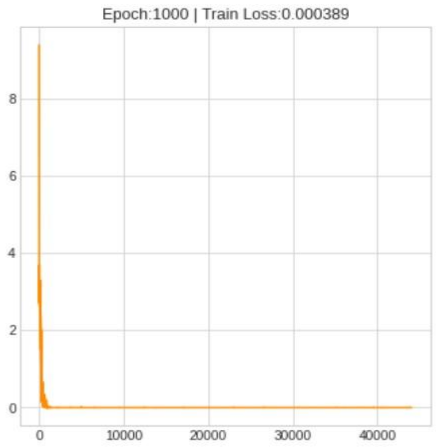


Figure 5.30: Training Loss vs (Epoch*Epoch Length) (CRNN)

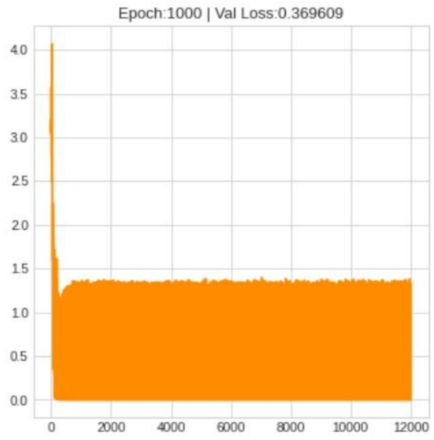


Figure 5.31: Validation Loss vs (Epoch*Epoch Length) (CRNN)

5.7.2 Text Recognition using YOLOv8

YOLO, which stands for You Look Only Once, divides the input image into a grid consisting of cells. Each cell in the grid is used for the purpose of predicting bounding boxes and class probabilities for objects present within that cell. YOLO uses predefined boxes of different sizes and aspect ratios called anchor boxes. These anchor boxes are used by the grid cell for predicting multiple bounding boxes by adjusting the coordinates and dimensions of the anchor boxes. The number of anchor boxes per grid cell is fixed beforehand. For each grid cell, YOLO predicts multiple bounding boxes. Each bounding box is characterized by its center coordinates, width, height, and confidence score. The confidence score represents the probability of containing an object and the accuracy of the bounding box. Additionally, each bounding box predicts class probabilities for different object classes. After predictions, non-maximum suppression is applied to remove redundant bounding boxes by considering their overlap and selecting the ones with higher confidence scores. It ensures that each object is represented by only one bounding box, eliminating duplicates. YOLO is trained using labelled images with bounding box annotations. During the course of training, the network learns to predict bounding box coordinates, confidence scores, and class probabilities. The loss function combines localization loss (box coordinate regression), confidence loss, and classification loss. During inference, the trained YOLO model takes an input image and performs forward propagation. It generates predictions for bounding boxes, confidence scores, and class probabilities. Non-Maximum Suppression is then applied to obtain the final object detection results. The architecture of YOLO is shown in figure 5.32 (page no 233).

YOLOv8n, YOLOv8s, and YOLOv8m are different versions of YOLOv8 architecture. These models are distinctly identified by suffixes indicating their size - "n" for nano, "s" for small, and "m" for medium. Different variations aim to achieve a balance between model dimensions, mean Average Precision (mAP), and inference time, customizing each to specific requirements. YOLOv8n is tailored for edge devices with limited computational resources, providing a balance between model size and inference speed. YOLOv8s is designed for scenarios where a compact model is essential, ensuring faster inference times. YOLOv8m strikes a balance between model size and accuracy, making it suitable for applications with a moderate amount of data. Choosing the appropriate model size hinges on specific requirements, taking into account factors such as model size vs. mAP, inference time, and data availability. Generally, larger models achieve higher mAP, while smaller models prove more efficient for storage and real-time applications.

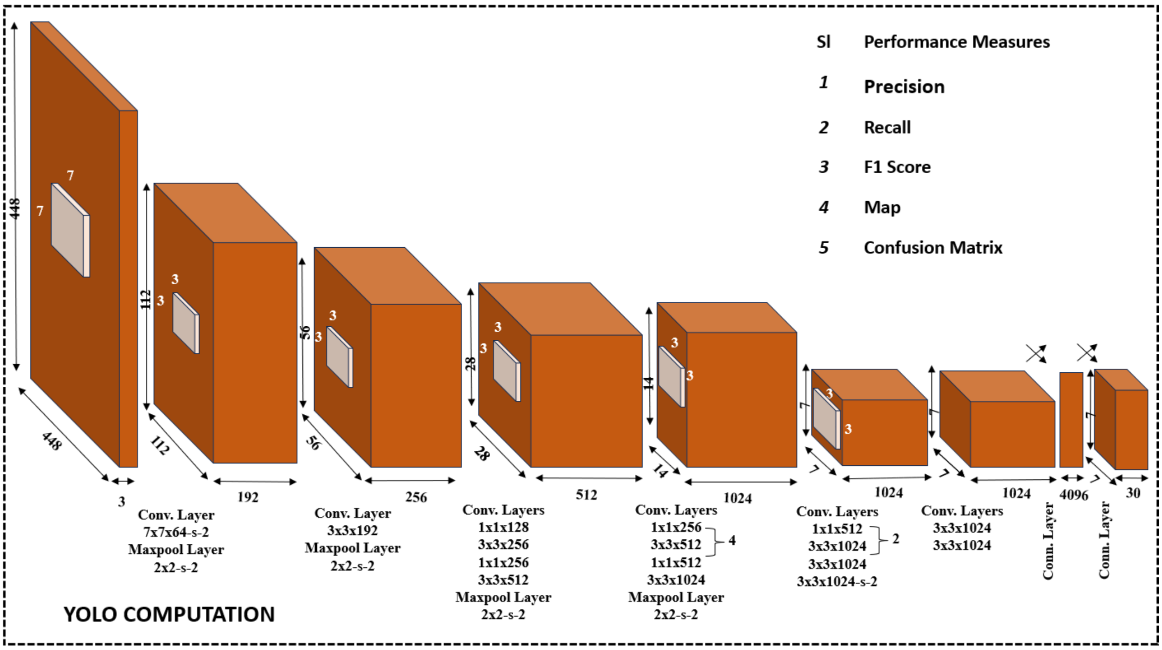


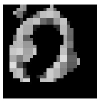
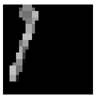
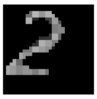
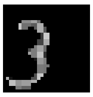
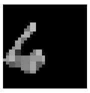
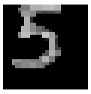
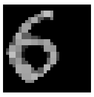
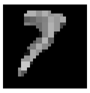
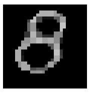
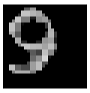
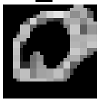
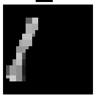
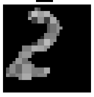
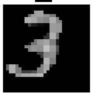
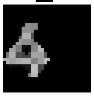
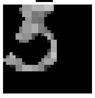
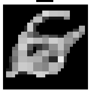
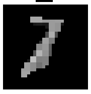
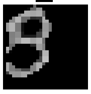
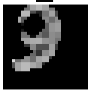
Figure 5.32: Recognition of numeric data using Yolo Architecture

5.7.2.1 Data Augmentation

The automatic vectorization of elevation values demands enough data to build a detection and recognition machine learning model. Since, the elevation values in topographic map intersects with other morphological features, the existing dataset may not adequately extract the significant feature automatically at the time of training the model, thereby degrading the performance of model. Hence, a customed dataset is essential that aids in automatic vectorization process for digitizing elevation values present in topographic map.

To address the aforementioned, this section presents an effort towards building a comprehensive reference dataset for numeric data (0-9) used for building various quantitative values represented in the topographic sheet. A variety of topographies were carefully inspected, and 1322 different variations of numeric data (0-9) were detected, as shown in table 5.7 (page no 234). Thereafter, these 1322 variations were appropriately resized and pre-processed to facilitate augmentation through various transformation measures. Transformation was achieved through smoothing, sharpening, rotation, noise elimination, logarithmic, power, exponential, and gamma operations on the 1322 initial, resulting in the generation of 26400 additional variations. This dataset may be of great use for training localization and recognition model.

Table 5.7: Samples of elevation digits extracted from localized topographic map

									
Sample_1	Sample_1	Sample_1	Sample_1	Sample_1	Sample_1	Sample_1	Sample_1	Sample_1	Sample_1
									
Sample_2	Sample_2	Sample_2	Sample_2	Sample_2	Sample_2	Sample_2	Sample_2	Sample_2	Sample_2
..	..	..	..	..	..	..	..	..	..
Sample_n	Sample_n	Sample_n	Sample_n	Sample_n	Sample_n	Sample_n	Sample_n	Sample_n	Sample_n

5.7.2.1.1 Proposed Methodology

Figure 5.33 (page no 236) elaboratively discusses the various steps involved in the generation of synthesized dataset using various transformation operations. The input topographic sheet is suitably pre-processed, followed by localization of text features using a deep network model based on ROI. The process starts with localization, followed by segmentation to identify the individual digits. The segmented digits are then suitably labelled. The labelled digits are then pre-processed for resizing, background subtraction, and grayscale conversion. Thereafter, transformations are performed, including smoothing, sharpening, rotation, noise elimination, logarithmic, power, exponential, and gamma operations.

Text resembling numeric data was searched for, and vertical segmentation was performed to segment the individual digits. The individual digits were then resized to 28x28 pixels to ensure uniformity of representation. The images were assigned tags resembling the value depicted by the image. Irrelevant information was selectively removed using the global thresholding technique for ease of processing. The pixel averaging technique was used for converting all the images to grey. Smoothening of the dataset has been performed using 3x3 and 5x5 masks. Sharpening of the dataset has been performed using the Laplacian filter. Rotation has been performed in degrees. Gaussian noise was induced to further alter the dataset. Logarithmic transformation of the dataset was performed considering c=1 and c=3. A power transformation of the dataset was performed by empirically considering c=1 with γ =1.2 and c=2 with γ =0.3. An exponential transformation of the dataset was performed by empirically considering c=1 with α=0.5 and c=2 with α=0.5. Gamma transformation of the dataset was performed by empirically considering γ=2 and γ=0.5. For accessing the dataset, commence by downloading and extracting the contents of the “Topographic Number Dataset.zip” file from the Mendeley data link. The details of various transformation technique realized are highlighted in table 5.8 (page no 236).

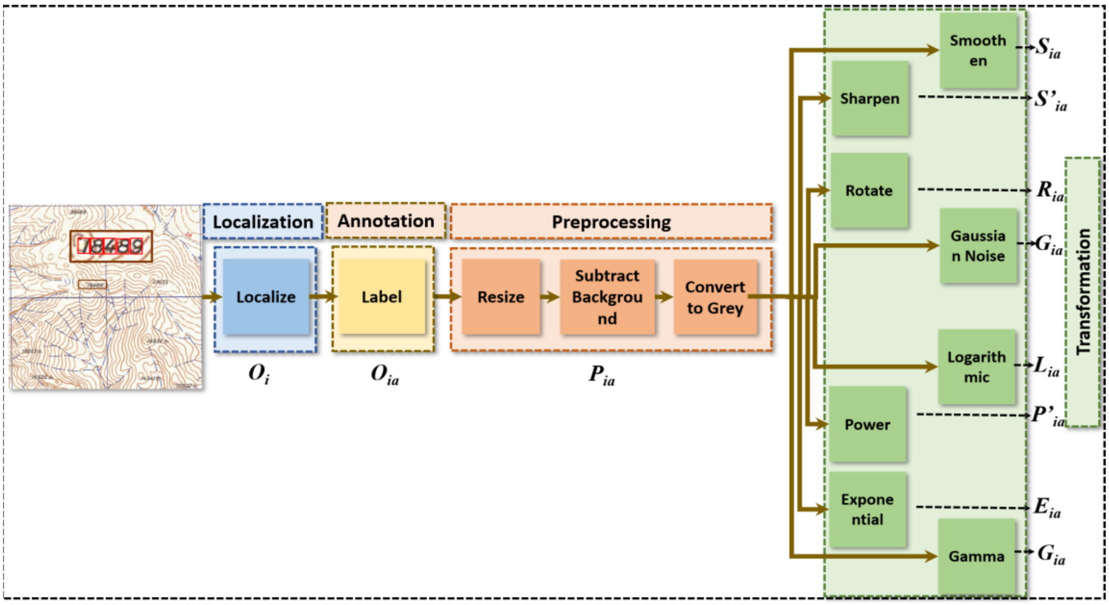


Figure 5.33: A schematic representation of various steps involved in the process of generating the augmented referenced dataset from the localized numeric values in the sample topographic sheets.

Table 5.8: Description of the transformation steps

Step	Description	Steps Involved
a)	Prepare sample topographic sheet	Preprocessing
b)	Localize text using deep architecture	• Annotate the region of interest (ROI) • Training the model using Deep architecture like Faster RCNN

		- Detect text for all samples and extract the ROI
c)	Perform vertical segmentation of character sequences in the numeric data segments to extract individual digits	Segment individual digits in the localized numeric data
d)	Resize the segmented digits for facilitating operations	Create uniform representation of all individual digits by resizing it to 28x28 pixels
e)	Label each digit logically (file extension: *.png)	Suitably tag the images with the corresponding unique numeric labels resembling the value
f)	Apply background subtraction to extract only feature on interest	Remove irrelevant information using global thresholding technique
g)	Convert all image dataset to gray scale image	Using pixel averaging technique
h)	Apply following transformation [17]:	

i **Smoothening Operation**

Resultant pixel value=($\sum x_i.w_i$)/ n, where x_i is the image element and w_i is the coefficient value of the mask and n is the total number of coefficient values in mask.

$$\mathbf{g(x,y) = \frac{1}{mn}\sum_{(s,t)\in S_{xy}} f(s,t)}$$

Description of the mask

- 3x3 sized mask

1	1	1
1	1	1
1	1	1

- 5x5 sized mask

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

…… (Equation 5.3)

where,

- g(x, y) is the resultant image
- f(s, t) is input image
- m x n is the size of the mask
- s x t is the size of the image

ii Sharpening operation using Laplacian filter

$$g(x,y)=5f(x,y)-f(x+1,y)+f(x-1,y)+f(x,y+1)+ f(x,y-1)$$

… (Equation 5.4)

where,

- g(x, y) is the resultant image
- f(x, y) is input image

Description of the mask

0	-1	0
-1	A+4	-1
0	-1	0

A is variable dictating the degree of sharpness

iii Rotation

In order to rotate any point present in the image, the generic equation used in implementation is:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ x } \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Where, (x, y) is a significant pixel location, , (x’, y’) is a rotated pixel

Where $\theta=0^0$, 45^0,90^0, 135^0, 180^0, 225^0,270^0, 315^0

Description of the mask

cos(θ)	-sin(θ)	0
sin(θ)	cos(θ)	0
0	0	1

location, and θ is the angle for rotation.

Using this strategy, 8 different datasets were generated.

iv Transformation by inducing Gaussian Noise

Initially, a gaussian noise is induced and removed using smoothening filter, particularly, arithmetic mean filter. The function for gaussian noise with random variable, z, is:

$$\mathbf{p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(z-\mu)^2/2\sigma^2}}$$

…..… **(Equation 5.5)**

where z represents gray level, μ is the mean value of z and σ is the standard deviation.

v Logarithmic Transformation

To transform the given dataset, the following transformation equation was applied to every pixel of every images.

$$\mathbf{s = c \log(1 + r)}$$

By empirically considering c=1 and c=3, two set of datasets was generated

	…..… **(Equation 5.6)** where c is the constant, and r≥0 is a pixel value. Using this strategy, 2 different datasets were generated.	
vi	**Power Transformation** To transform the given dataset, the following transformation equation was applied to every pixel of every images. $\mathbf{x = cr^{\gamma}}$ **….. (Equation 5.7)** where c is the constant, r≥0 is a pixel value, and γ is the variable. Using this strategy, 2 different datasets were generated.	By empirically considering c=1 with γ =1.2 and c=2 with γ =0.3, two set of datasets was generated
vii	**Exponential Transformation** To transform the given dataset, the following transformation equation was applied to every pixel of every images. $\mathbf{s=c*((1+ \alpha)r -1)}$ ….. **(Equation 5.8)**	By empirically considering c=1 with α=0.5 and c=2 with α=0.5, two set of datasets was generated

	where c is the constant, $r \geq 0$ is a pixel value, and α is a variable. Using this strategy, 2 different datasets were generated.	
viii	**Gamma Transformation** To transform the given dataset, the following transformation equation was applied to every pixel of every images. **$x = r^{\gamma}$….. (Equation 5.9)** where $r \geq 0$ is a pixel value and γ is the variable. Using this strategy, 8 different datasets were generated.	By empirically considering γ=2 and γ=0.5, two set of datasets was generated

This initiative successfully generates a comprehensive dataset consisting of two subsets. The first subset has 1322 original images recorded from various topographic sheets, and the second dataset has 26440 images created through various transformations resulting from smoothing, sharpening, rotation, noise elimination, logarithmic, power, exponential, and gamma operations. Table 5.9 (page no 242) represents the original image and the synthesized images obtained through the various transformations.

Table 5.9: Description of original and synthesized data

Dataset	Type	Amount	Description
P_{ia}	Original	1322	Obtained through segmentation of localized numeric data in the topographic sheet
S_{ia}	Synthesized	1322	Obtained through application of Smoothening Operation on the original dataset with a window of 3x3
S_{ia}	Synthesized	1322	Obtained through application of Smoothening Operation on the original dataset with a window of 5x5
S'_{ia}	Synthesized	1322	Obtained through application of Sharpening Operation on the original dataset
R_{ia}	Synthesized	1322	Obtained through application of Rotation Operation on the original dataset with $\theta=0^0$
R_{ia}	Synthesized	1322	Obtained through application of Rotation Operation on the original dataset with $\theta=45^0$
R_{ia}	Synthesized	1322	Obtained through application of Rotation Operation on the original dataset with $\theta=90^0$
R_{ia}	Synthesized	1322	Obtained through application of Rotation Operation on the original dataset with $\theta=135^0$
R_{ia}	Synthesized	1322	Obtained through application of Rotation Operation on the original dataset with $\theta=180^0$
R_{ia}	Synthesized	1322	Obtained through application of Rotation Operation on the original dataset with $\theta=225^0$

R_{ia}	Synthesized	1322	Obtained through application of Rotation Operation on the original dataset with $\theta=270^0$
R_{ia}	Synthesized	1322	Obtained through application of Rotation Operation on the original dataset with $\theta=315^0$
G_{ia}	Synthesized	1322	Obtained through application of Gaussian Noise on the original dataset
L_{ia}	Synthesized	1322	Obtained through application of Logarithmic Transformation on the original dataset with $c=1$
L_{ia}	Synthesized	1322	Obtained through application of Logarithmic Transformation on the original dataset with $c=3$
P'_{ia}	Synthesized	1322	Obtained through application of Power Transformation on the original dataset with $c=1$ and γ =1.2
P'_{ia}	Synthesized	1322	Obtained through application of Power Transformation on the original dataset with $c=2$ and $\gamma=0.5$
E_{ia}	Synthesized	1322	Obtained through application of Exponential Transformation on the original dataset with $c=1$ and $\alpha=0.5$
E_{ia}	Synthesized	1322	Obtained through application of Exponential Transformation on the original dataset with $c=2$ and $\alpha=0.5$
G_{ia}	Synthesized	1322	Obtained through application of Gamma Transformation on the original dataset with $\gamma=2$

G_{ia}	Synthesized	1322	Obtained through application of Gamma Transformation on the original dataset with $\gamma=0.5$

where,

Dataset	Description
P_{ia}	Original numeric data from topographic map
S_{ia}	Smoothening Operation on the original dataset with a window of 3x3
S_{ia}	Smoothening Operation on the original dataset with a window of 5x5
S'_{ia}	Sharpening Operation on the original dataset
R_{ia}	Rotation Operation on the original dataset with $\theta=0^0$
R_{ia}	Rotation Operation on the original dataset with $\theta=45^0$
R_{ia}	Rotation Operation on the original dataset with $\theta=90^0$
R_{ia}	Rotation Operation on the original dataset with $\theta=135^0$
R_{ia}	Rotation Operation on the original dataset with $\theta=180^0$
R_{ia}	Rotation Operation on the original dataset with $\theta=225^0$
R_{ia}	Rotation Operation on the original dataset with $\theta=270^0$
R_{ia}	Rotation Operation on the original dataset with $\theta=315^0$
G_{ia}	Gaussian Noise on the original dataset
L_{ia}	Logarithmic Transformation on the original dataset with c=1
L_{ia}	Logarithmic Transformation on the original dataset with c=3
P'_{ia}	Power Transformation on the original dataset with c=1 and γ =1.2
P'_{ia}	Power Transformation on the original dataset with c=2 and $\gamma=0.5$
E_{ia}	Exponential Transformation on the original dataset with c=1 and $\alpha=0.5$
E_{ia}	Exponential Transformation on the original dataset with c=2 and $\alpha=0.5$
G_{ia}	Gamma Transformation on the original dataset with $\gamma=2$
G_{ia}	Gamma Transformation on the original dataset with $\gamma=0.5$

The subset of the synthesized dataset is presented in figure 5.34 (page no 245). The complete dataset has been uploaded in Mendeley data and can be accessed using the direct link specified: (https://data.mendeley.com/preview/jxnfmkkpj3?a=c45ee09b-0586-4d4a-8b94-bb4f7b1214cc)

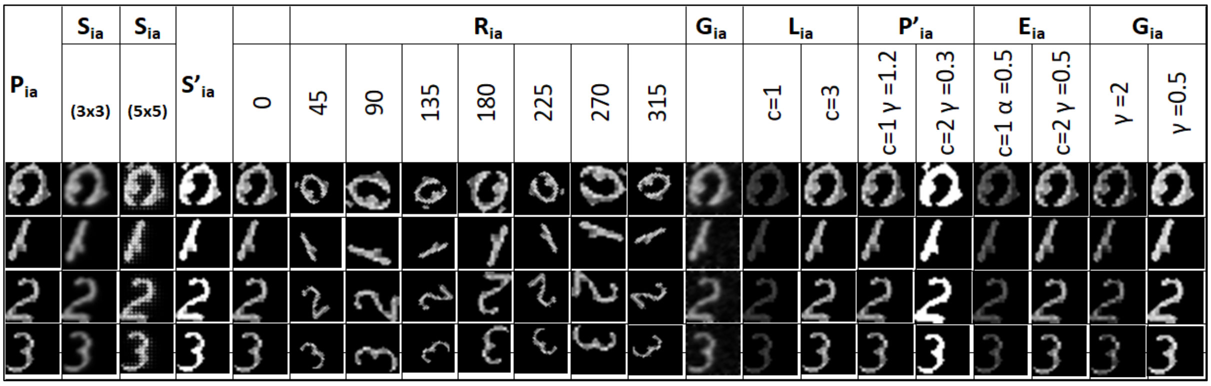


Figure 5.34: Subset of synthesised dataset.

5.7.2.2 Performance of the YOLOv8 models on augmented dataset

The YOLO architecture can train the model to recognize the elevation digits in one pass. Hence, it saves the time and cost. To recognize the elevation digits, three version of YOLOv8 algorithm namely, YOLOv8n, YOLOv8s and YOLOv8m have been considered.

5.7.2.2.1 Model Evaluation of Yolov8n

Summary for evaluation of Yolov8n is specified in table 5.10 (page no 245).

Table 5.10: Model Summary for Yolov8n

Attribute	Value
Total Layers	168
Total Parameters	3,007,598
Gradients	0
GFLOPs	8.1

Summary of evaluated results from Yolov8n is specified in table 5.10 (page no 245). Here,

$$\mathbf{F1=2.\frac{Precision*Recall}{Precision+Recall}} \text{.................. (Equation 5.10)}$$

$$\mathbf{Precision=\frac{TP}{TP+FP}} \text{.................. (Equation 5.11)}$$

$$\mathbf{Recall=\frac{TP}{TP+FN}} \text{.................. (Equation 5.12)}$$

The table 5.11 (page no 246) provides a detailed breakdown of the model's performance, including precision, recall, mAP (mean Average Precision) at various confidence thresholds (mAP@50 and mAP@50-95), and the number of images and instances processed for each class. These metrics offer a comprehensive evaluation of the model's ability to recognize characters in topographical maps across different classes and confidence levels.

Table 5.11: Evaluation Results for Yolov8n

Class	**Images**	**Instances**	**Precision (P)**	**Recall (R)**	**mAP@50**	**mAP@50-95**	**F1 Score**
All	387	387	0.957	0.932	0.98	0.979	0.944
0	387	61	0.959	0.967	0.98	0.98	0.963
1	387	72	0.922	0.98	0.982	0.982	0.95
2	387	47	0.953	0.872	0.977	0.977	0.911
3	387	29	0.965	0.95	0.983	0.969	0.957
4	387	28	0.95	0.964	0.991	0.991	0.957

5	387	33	1	0.907	0.976	0.976	0.951
6	387	22	0.889	0.909	0.981	0.981	0.899
7	387	35	0.971	0.955	0.992	0.992	0.963
8	387	32	1	0.904	0.985	0.985	0.941
9	387	28	0.962	0.916	0.958	0.958	0.938

Evaluation speed for Yolov8n is specified in table 5.12 (page no 247).

Table 5.12: Speed of Yolov8n

Attribute	**Value**
Pre-process	0.1 ms per image
Inference	0.7 ms per image
Loss Calculation	0.0 ms per image
Post-process	2.7 ms per image

The figure 5.35 (page no 247) represents confusion matrix for Yolov8n model.

Predicted

0	0.90			0.03			0.05			0.04	0.11
1	0.03	0.99	0.04					0.03			0.11
2			0.91					0.03		0.04	0.02
3			0.02	0.83		0.06					0.18
4					0.96						0.13
5			0.02	0.07		0.94					0.07
6	0.02				0.04		0.91		0.06	0.04	0.16
7								0.94			0.07
8				0.07			0.05		0.87		0.11
9	0.05	0.01							0.06	0.89	0.04
BG	0	1	2	3	4	5	6	7	8	9	BG

True

Figure 5.35: Confusion matrix for Yolov8n model

The F1 curve, P curve, R curve and PR curve for Yolov8n presented in figure 5.36-5.39 (page no 248).

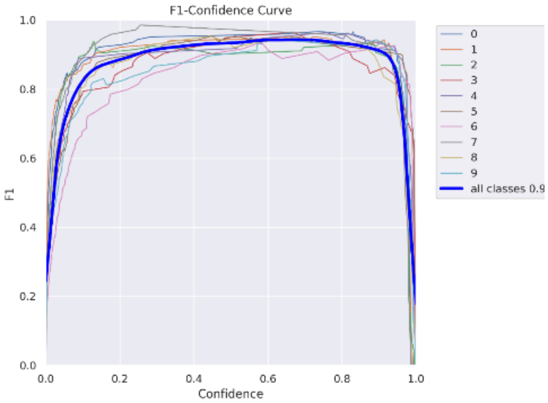


Figure 5.36: F1 Curve

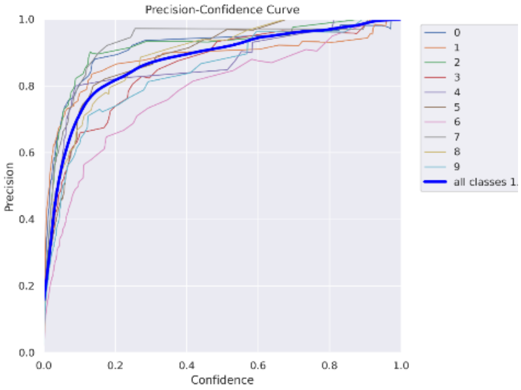


Figure 5.37: P Curve

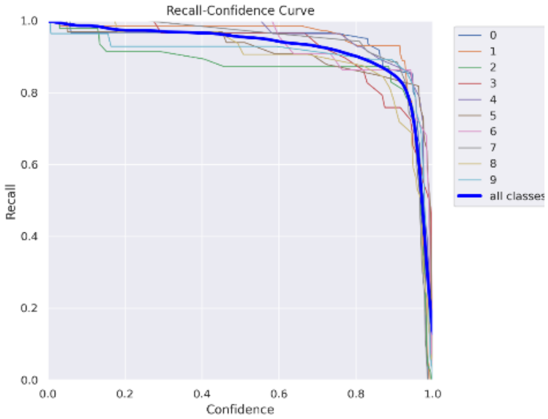


Figure 5.38: R Curve

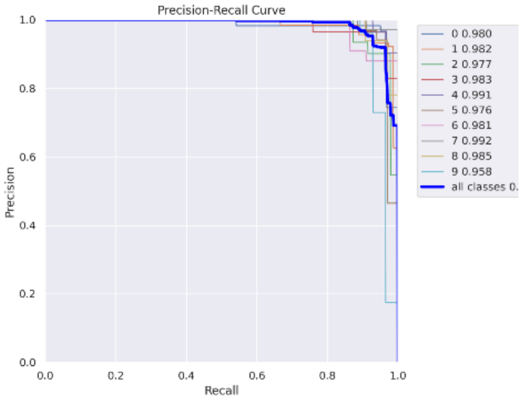


Figure 5.39: PR Curve

The different losses, precision, recall and mAP for Yolov8n presented in figure 5.40 (page no 248).

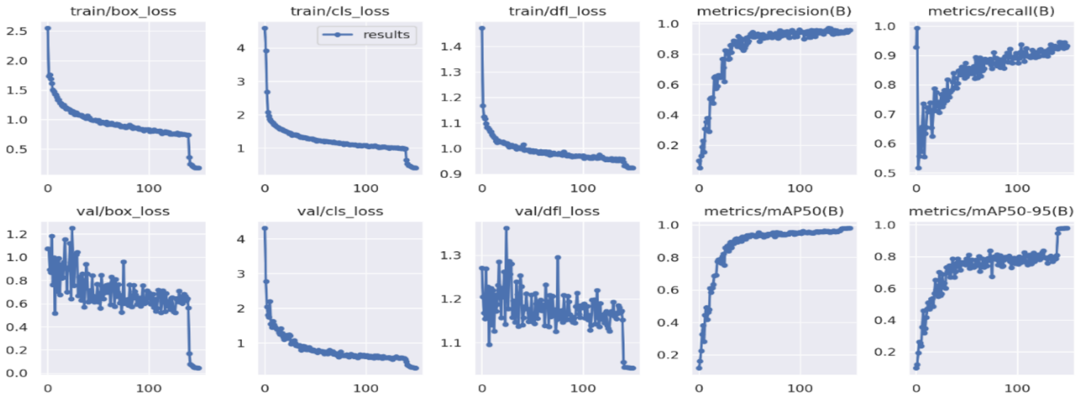


Figure 5.40: Losses, Precision, Recall and mAP for Yolov8n

5.7.2.2.2 Model Evaluation of Yolov8s

Model Summary for evaluation of Yolov8s is specified in table 5.13 (page no 249).

Table 5.13: Model Summary for Yolov8s

Attribute	Value
Total Layers	168
Total Parameters	11,129,454
Gradients	0
GFLOPs	28.5

The table 5.14 (page no 249) provides a detailed breakdown of the model's performance, including precision, recall, mAP (mean Average Precision) at various confidence thresholds (mAP@50 and mAP@50-95), and the number of images and instances processed for each class. These metrics offer a comprehensive evaluation of the model's ability to recognize characters in topographical maps across different classes and confidence levels. The F1 score, calculated from precision and recall, provides an additional measure of the model's overall performance for each class.

Table 5.14: Evaluation Results for Yolov8s

Class	Images	Instances	Precision (P)	Recall (R)	mAP@50	mAP@50-95	F1 Score
All	387	387	0.971	0.973	0.993	0.992	0.972
0	387	61	0.970	0.967	0.993	0.993	0.969
1	387	72	0.951	0.986	0.987	0.987	0.968
2	387	47	1.000	0.925	0.990	0.990	0.961
3	387	29	0.978	1.000	0.995	0.987	0.989
4	387	28	0.950	1.000	0.994	0.994	0.974
5	387	33	0.977	0.970	0.993	0.993	0.974

6	387	22	0.919	1.000	0.993	0.993	0.957
7	387	35	1.000	0.959	0.995	0.995	0.979
8	387	32	1.000	0.968	0.995	0.995	0.984
9	387	28	0.964	0.953	0.992	0.992	0.958

Evaluation speed for Yolov8s is specified in table 5.15 (page no 250).

Table 5.15: Speed of Yolov8s

Attribute	**Value**
Pre-process	0.0 ms per image
Inference	3.4 ms per image
Loss Calculation	0.0 ms per image
Post-process	0.5 ms per image

The figure 5.41 (page no 250) represents confusion matrix for Yolov8s model.

Predicted

0	0.95										0.24
1	0.03	0.99	0.06					0.03			0.10
2			0.89								0.19
3			0.02	1.00					0.03		0.05
4			0.02		1.00						
5						0.97					0.14
6	0.02					0.03	1.00		0.03		0.1
7								0.97			0.05
8									0.94		0.1
9	0.01									1.00	0.05
BG	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**BG**

True

Figure 5.41: Confusion matrix for Yolov8s model

The F1 curve, P curve, R curve and PR curve for Yolov8s presented in figure 5.42-5.45 (page no 251).

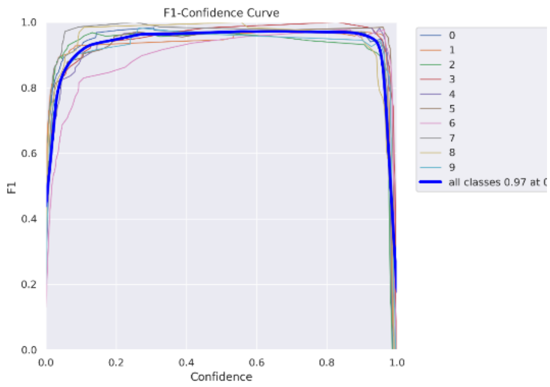


Figure 5.42: F1 Curve

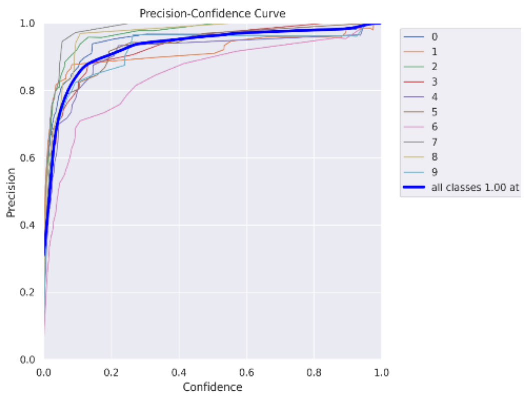


Figure 5.43: P Curve

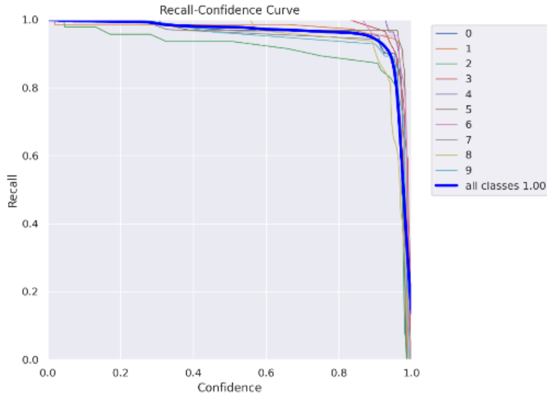


Figure 5.44: R Curve

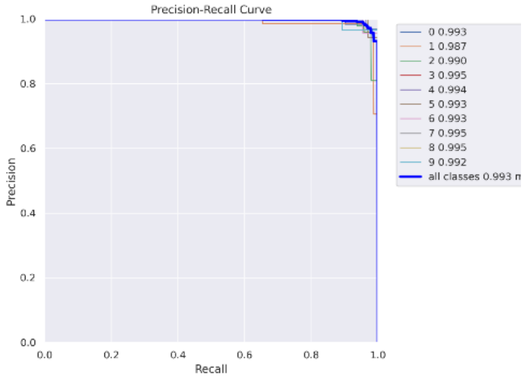


Figure 5.45: PR Curve

The different losses, precision, recall and mAP for Yolov8s presented in figure 5.46 (page no 251).

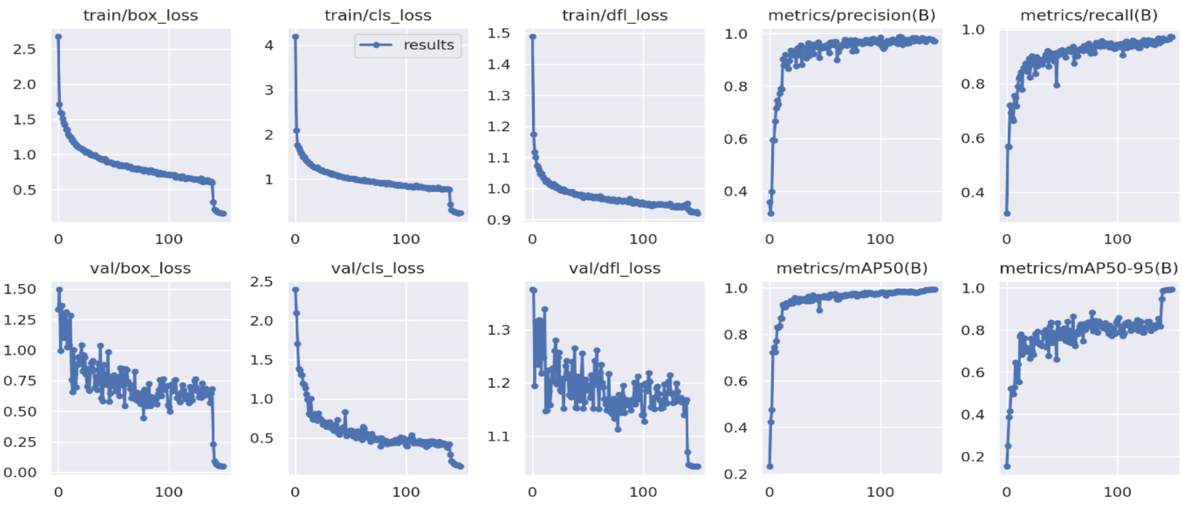


Figure 5.46: Losses, Precision, Recall and mAP for Yolov8s

5.7.2.2.3 Model Evaluation of Yolov8m

Model Summary for evaluation of Yolov8m is specified in table 5.16 (page no 252).

Table 5.16: Model Summary for Yolov8m

Attribute	Value
Total Layers	218
Total Parameters	25,845,550
Gradients	0
GFLOPs	78.7

The table 5.17 (page no 252) provides a detailed breakdown of the model's performance, including precision, recall, mAP (mean Average Precision) at various confidence thresholds (mAP@50 and mAP@50-95), and the number of images and instances processed for each class. These metrics offer a comprehensive evaluation of the model's ability to recognize characters in topographical maps across different classes and confidence levels, given the summary details of the model's architecture and performance.

Table 5.17: Evaluation Results for Yolov8m

Class	Images	Instances	Precision (P)	Recall (R)	mAP@50	mAP@50-95	F1 Score
All	387	387	0.983	0.972	0.992	0.991	0.978
0	387	61	0.962	1.000	0.993	0.991	0.980
1	387	72	0.959	0.975	0.991	0.987	0.967
2	387	47	1.000	0.898	0.993	0.993	0.946
3	387	29	0.995	0.966	0.989	0.986	0.980
4	387	28	0.953	1.000	0.995	0.992	0.976
5	387	33	1.000	0.980	0.995	0.995	0.990

6	387	22	0.985	1.000	0.995	0.995	0.992
7	387	35	1.000	0.971	0.983	0.983	0.986
8	387	32	0.988	1.000	0.995	0.995	0.994
9	387	28	0.990	0.929	0.993	0.993	0.959

Evaluation speed for Yolov8m is specified in table 5.18 (page no 253).

Table 5.18: Speed of Yolov8m

Attribute	Value
Pre-process	0.1 ms per image
Inference	1.9 ms per image
Loss Calculation	0.0 ms per image
Post-process	1.9 ms per image

The figure 5.47 (page no 253) represents confusion matrix for Yolov8m model.

Predicted

	0	1	2	3	4	5	6	7	8	9	BG
0	1.00			0.03						0.07	
1		0.99	0.02			0.03		0.03			0.47
2			0.94								0.05
3				0.97							
4			0.02		0.96						0.11
5			0.02			0.97					0.11
6					0.04		1.00				0.21
7								0.97			
8									1.00		0.05
9		0.01								0.93	
BG											

True

Figure 5.47: Confusion matrix for Yolov8m model

The F1 curve, P curve, R curve and PR curve for Yolov8m presented in figure 5.48-5.51 (page no 254).

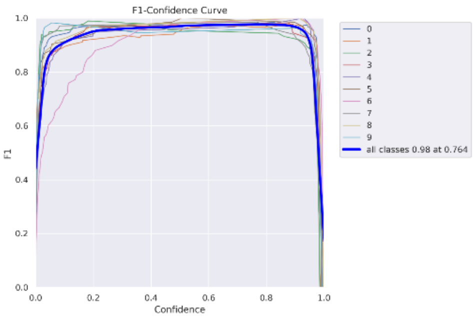


Figure 5.48: F1 Curve

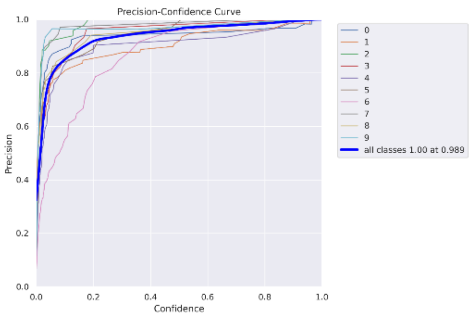


Figure 5.49: P Curve

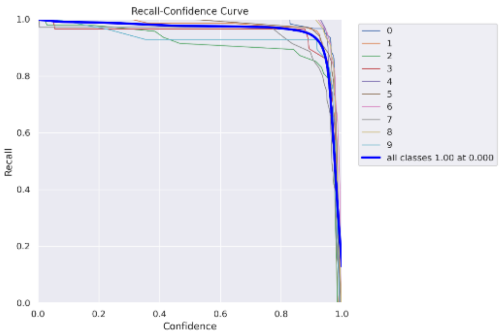


Figure 5.50: R Curve

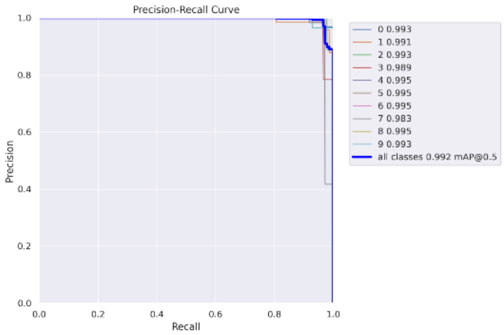


Figure 5.51: PR Curve

The different losses, precision, recall and mAP for Yolov8m presented in figure 5.52 (page no 254).

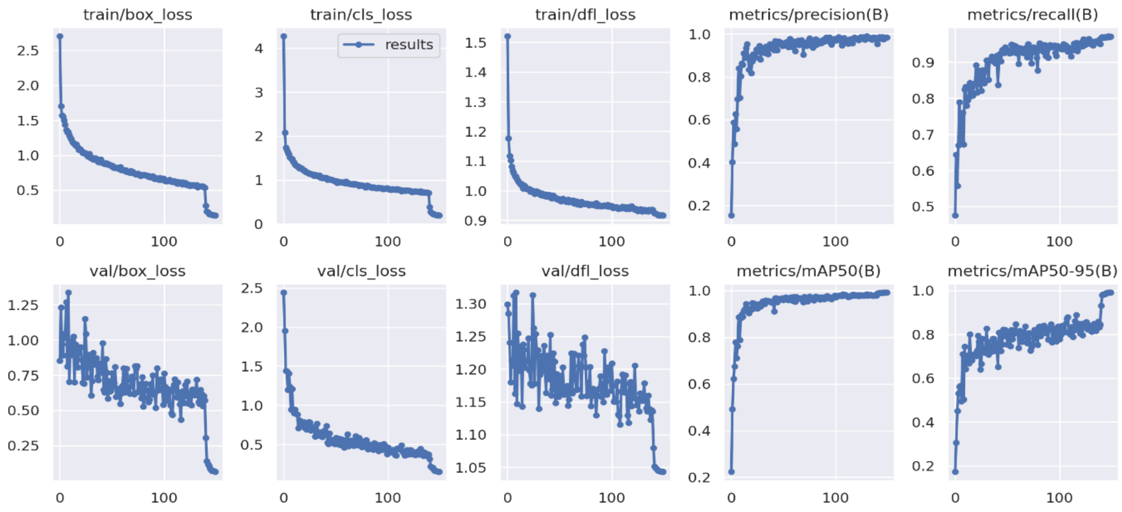


Figure 5.52: Losses, Precision, Recall and mAP for Yolov8m

5.7.2.2.4 Comparative analysis of Yolov8n, Yolov8s and Yolov8m

Table 5.19 (page no 255) provides a comparative analysis of the three YOLOv8 models in terms of their architecture, performance metrics, and speed during inference. The differences can be prominently observed in parameters, GFLOPs, precision, recall, mAP50, mAP50-95, and inference speed for each model. These metrics provides a valuable insight into how different model sizes impact character recognition in topographical maps.

Table 5.19: Comparative analysis of Yolov8n, Yolov8s and Yolov8m

Model	**Total Parameters**	**GFLOPs**	**Precision (all)**	**Recall (all)**	**mAP @50 (all)**	**mAP @50-95 (all)**	**Speed (Inference) (In ms/image)**
YOLOv8n	30,07,598	8.1	0.957	0.932	0.98	0.979	0.7
YOLOv8s	1,11,29,454	28.5	0.971	0.973	0.993	0.992	3.4
YOLOv8m	2,58,45,550	78.7	0.983	0.972	0.992	0.991	1.9

5.8 Limitations of the research initiative

The proposed methodologies implement localization of elevation values present in topographic sheet with the aid of limited dataset. In addition, the intersection of different morphological features dilutes the quality of output, hence, making it more challenging. Due to limited dataset, the accuracy rate achieved for localization is not appreciable. To recognize the evaluation values using supervised technique requires annotation of extracted features which in itself is a very time-consuming process.

5.9 Conclusion

Automated feature extraction from poor-quality topographic sheets poses quite a challenge as several relief features are often represented in an overlapped manner. The computational engine must be tuned to enable effective identification and further interpretation. In addition, the quality of the topographic sheet also plays a crucial role in the determination of the performance measure. On close analysis, it was observed that in a given representation, the elevation value was represented with a similar font and size but may have different orientation. In some instances, the elevation value intersected with other morphological features, which made the background segmentation process difficult. It may be noted that filling the holes with elevation values majorly contributed to retain the information during the erosion process, and a similar idea was used for the localization of elevation values. The results have been tested with a variety of sample inputs, and the approach was found to be effective. Localization is difficult in situations where the elevation data is encompassed by different geographic regions or intersects with other morphological features. It is also noted that the segmentation of a character with an intersecting elevation value with other morphological values and its recognition are also big challenges. This provides an area for future research initiatives in order to make a fully automatic computational system for the vectorization of topographic sheets and their representation.

Although the DEM can be constructed from many other sources, like satellite data, creating the DEM from a scanned topographic sheets will be more accurate and less expensive [257] [258]. In this research initiative, a novel approach based on morphological and image processing operations has been realized to identify the location (x, y) of elevation values present in a topographic sheet.

This work aims at locating text representing different relief features like, landmark name or elevation value, using the faster R-CNN technique. The model was trained using a variety of poor-quality samples of topographic sheets, which had in them many text features in different orientations, to check the robustness of the trained model. An accuracy of 72.42% was achieved using the faster R-CNN model. To check the performance of the results as reflected by the faster R-CNN model, a pre-trained R-CNN was trained using the same training set. An accuracy of 51.44% was obtained by R-CNN. This proves that the faster RCNN performs well for the localization of not only elevation values but also any text features present in the topographic sheet. The overall accuracy achieved was 85%.

With a detector and recognizer model trained and validated on datasets of topographic sheet, these can be further used to achieve the task of retrieving information from topographic sheet, thereby removing human intervention. In addition to the aforementioned technique, a faster RCNN with R-101-FPN as a backbone was also trained for text detection or localization. An accuracy of 79.6% against Intersection over Union (IoU) value >= 0.7 is achieved using Faster RCNN+R-101-FPN.

For text recognition, CRNN model was trained with localized elevation detail. An accuracy of 60% was achieved for the training set and 45% accuracy was achieved for validation set. One major issue with this trained model is the limited dataset. Hence, the volume of dataset was increased using various data augmentation techniques, like smoothing, sharpening, rotation, noise elimination, logarithmic, power, exponential, and gamma operations. The dataset which was originally 1322 was increased to 26440. Then, YOLOv8 models were trained using the augmented dataset to recognize the elevation values of topographic sheet. Three versions of YOLOv8 namely, YOLOv8n, YOLOv8s and YOLOv8m were trained and compared. YOLOv8n was found to be the lightest and smallest model during execution, with the lowest GFLOPs and total parameters. Its inference speed of 0.7 ms per image was found to be relatively rapid and demonstrated good recall and precision. YOLOv8s was found capable in successfully balancing performance and size. It performed better in terms of precision, recall, and mAP than YOLOv8n since it has larger total parameters and GFLOPs. At 3.4 ms per image, the inference speed was found to be relatively slower. With the greatest GFLOPs and total parameters, YOLOv8m is the largest and most accurate model. Out of the three models, YOLOv8m attains the highest precision, recall, and mean average precision (mAP); nonetheless, its inference speed was found to be 1.9 ms per image. YOLOv8n works well in situations where processing power is scarce. YOLOv8s strikes a decent mix of performance and size, making it a viable option for a general-purpose model. Although YOLOv8m has a slower inference speed and demands more computer resources, it gives the best accuracy. The application-specific requirements determine which model is best, taking into account things like accuracy, processing speed in real time, and computational capacity. YOLOv8m is regarded as the best model since it recognizes elevation digits on a topographical map, which is a non-time-critical activity.

There remains a big challenge of producing a robust, high-performance detector and especially a recognizer network. The current recognizer model is limited to just English alphabet and needs to be expanded to numbers and some non-English alphabet characters commonly seen on topographic sheets of Indian region. The recognizer model has low validation as well as training accuracy which could be attributed to the low-quality images in the recognition dataset. Data is driving these models hence having data that is engineered to correctly represent what a model must learn and in enough numbers is important and almost necessary. Though work on generalizing over small dataset is in research and also probably shown in [259], currently state of the art deep learning models performs well on large amount of data. Specific to topographic sheets, engineering a model for images containing texts with line strides of contours passing over and around could be heavily challenging. Wrong representations of characters could be learned that would suffer on testing. The next most plausible step would be to create handcrafted image processing techniques to counter wrong representations and increase receptiveness of data to be learned. Fu et al [260] de-noises images with pepper like noise using CNNs. This inspires to research and work on removing strikes and unwanted lines from images using some variation of neural networks. Dataset for recognition was limited to just words in English alphabet but can be broadened to numbers. Detecting texts not horizontally inclined with non-horizontal bounding box is another sought out issues [261].

Chapter 6

Contour Generation and 3D Modelling

6.1 Introduction

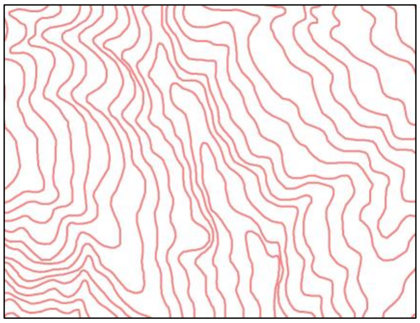
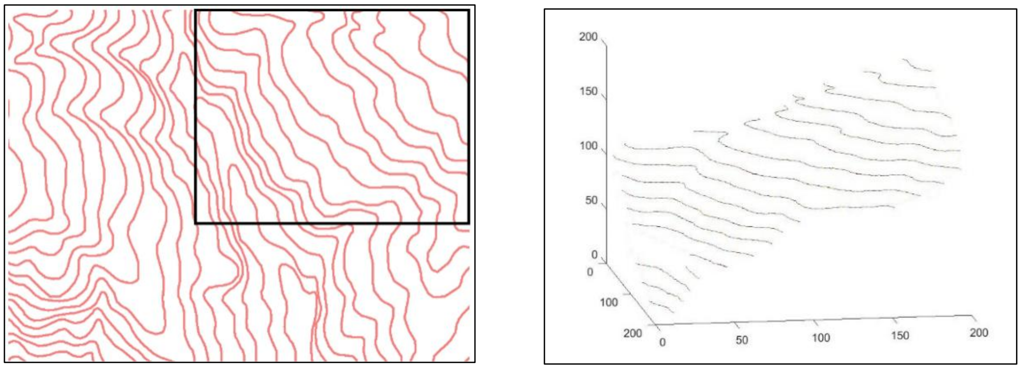


Figure 6.1: Sample Contour Map

Contour, also known as isoline or isopleth, are virtual lines (as shown in figure 6.1 (page no 260)) used to join points in the landscape that are located at the same elevation from a given reference point (for example, sea level). These are non-intersecting and may take the form of a line or a polygon. Attributed with numeric values representing its elevation, these lines are primarily used for representing a morphological aspect of landforms and terrain information and are also widely used for creating elevation models. In the topographic sheets, these lines are represented at a certain vertical distance value known as the contour intervals. Depending upon the scale at which the topographic sheet is built, the size of the contour interval may vary. This implies that if the scale of the map is considerably small, the contour interval should be greater and vice versa.

It has been often observed that the variability in the contour interval has a tremendous impact on the quality of the DEM created using the contours. DEM is a 3D computer graphics model of an object of interest. In context of studies related to GIS, DEM refers to the 3D model of a landscape; figure 6.2 b) (page no 261) represents the 3D model of the highlighted portion of the contour map in figure 6.2 (a) (page no 261). Such a 3D model finds its application in planning civil engineering projects, comprehensive terrain analysis, cadastral mapping, suitability mapping for the establishment of communication networks, and many more. DEMs built using contours with greater contour intervals generally have steeper curves and sharper inclinations. Reliance on such an imprecise DEM model may result in the generation of a low-confidence scientific basis for formulating a solution model for critical terrain-related problems.



(a) (b)

Figure 6.2: a) Portioned Sample Contour Map b) 3D model of Portioned Sample Contour Map

In addition, such an elevation model may correlate badly with the actual ground reality. Figure 6.3 (page no 263) represents a sample contour map. In figure 6.4 (page no 263), a portion of the contours has been digitized. Figure 6.5 (page no 263) represents a set of cross-sectional points in the digitized contours. Figure 6.6 (page no 263) represents an elevation model generated by plotting the cross-sectional points at certain intervals. Here, it is observed that the curve generated is steep, and in this situation, it may not be able to correlate with the actual landscape of the geographical area. Ideally, it would be best to use contour maps with lower contour intervals as a basis for generating elevation models, but in situations where the same is not available or difficult to procure, maps with larger contour intervals may be digitally processed with the help of knowledge-based computation programs to incorporate integrity-preserving features for elevating the quality of elevation maps. Figure 6.7 (page no 263) represents the incorporation of an integrity-preserving contour into the contour map using knowledge-based computing. Figure 6.8 (page no 263) represents a set of cross-sectional points in the existing and generated contours. Figure 6.9 (page no 263) represents an elevation model generated by plotting the cross-sectional points at certain intervals. Here, it is observed that the curve generated is much smoother and correlates better with the actual landscape of the geographical area, as represented in figure 6.10 (page no 263).

6.2 Objectives of the proposed work

The objectives of this research initiative are as follows:

a) Identify all the structural operators prominently describing the characteristics of the contour lines

b) Understand the significance of structural operators in relation to its relative position with respect to certain reference points

c) Build strategies for traversal and interpolation while preserving the structural integrity of the contour lines

d) Efficiently connect the interpolated elements to form continuous contour elements

e) Associate elevation details with contours and elevate the same in the 3D plane to create the digital elevation model and its assessment

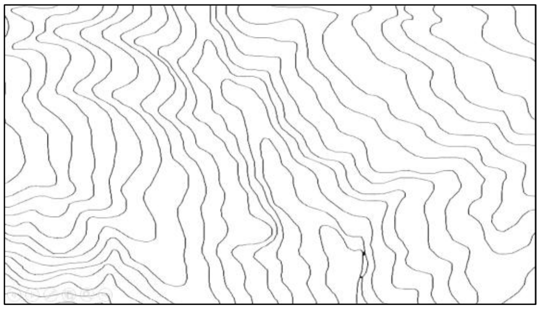


Figure 6.3: Sample Contour Map

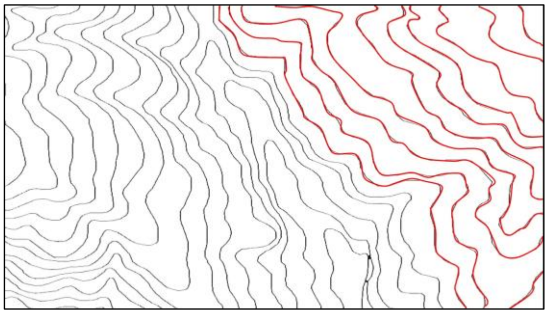


Figure 6.4: Digitized Contour Map

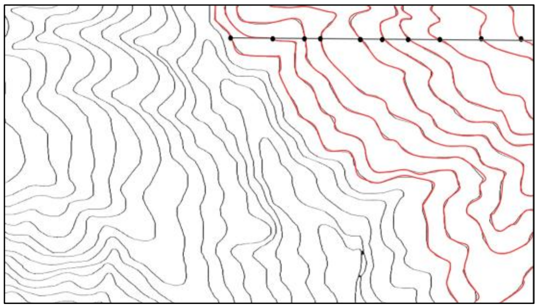


Figure 6.5: Selection of cross-sectional points

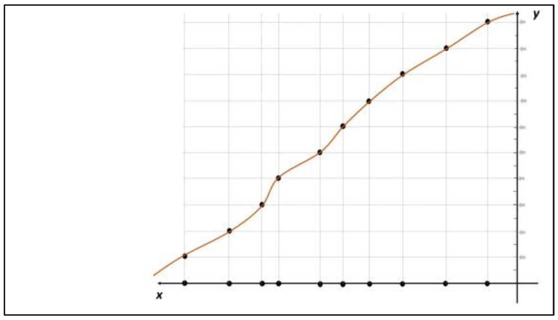


Figure 6.6: Elevation model using cross-sectional points

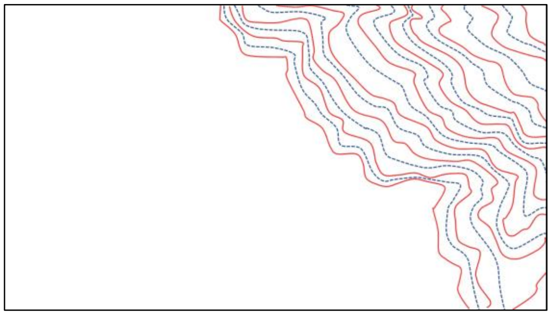


Figure 6.7: Incorporation of integrity preserving contours

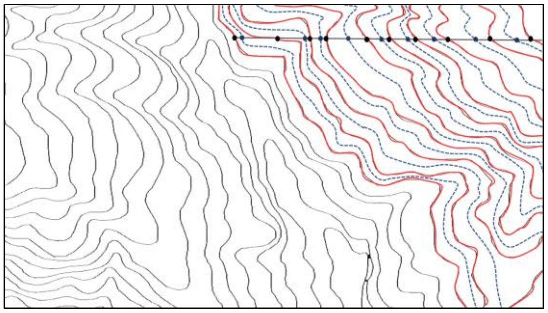


Figure 6.8: Selection of cross-sectional points

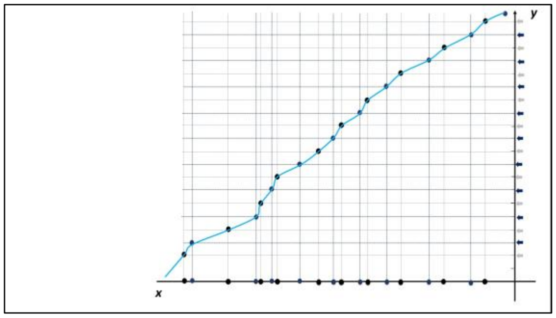


Figure 6.9: Elevation model using cross-sectional

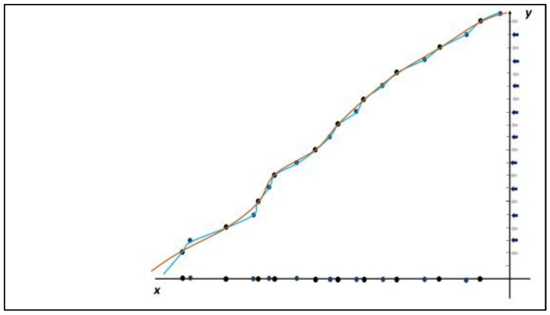


Figure 6.10: Comparison of two elevation models

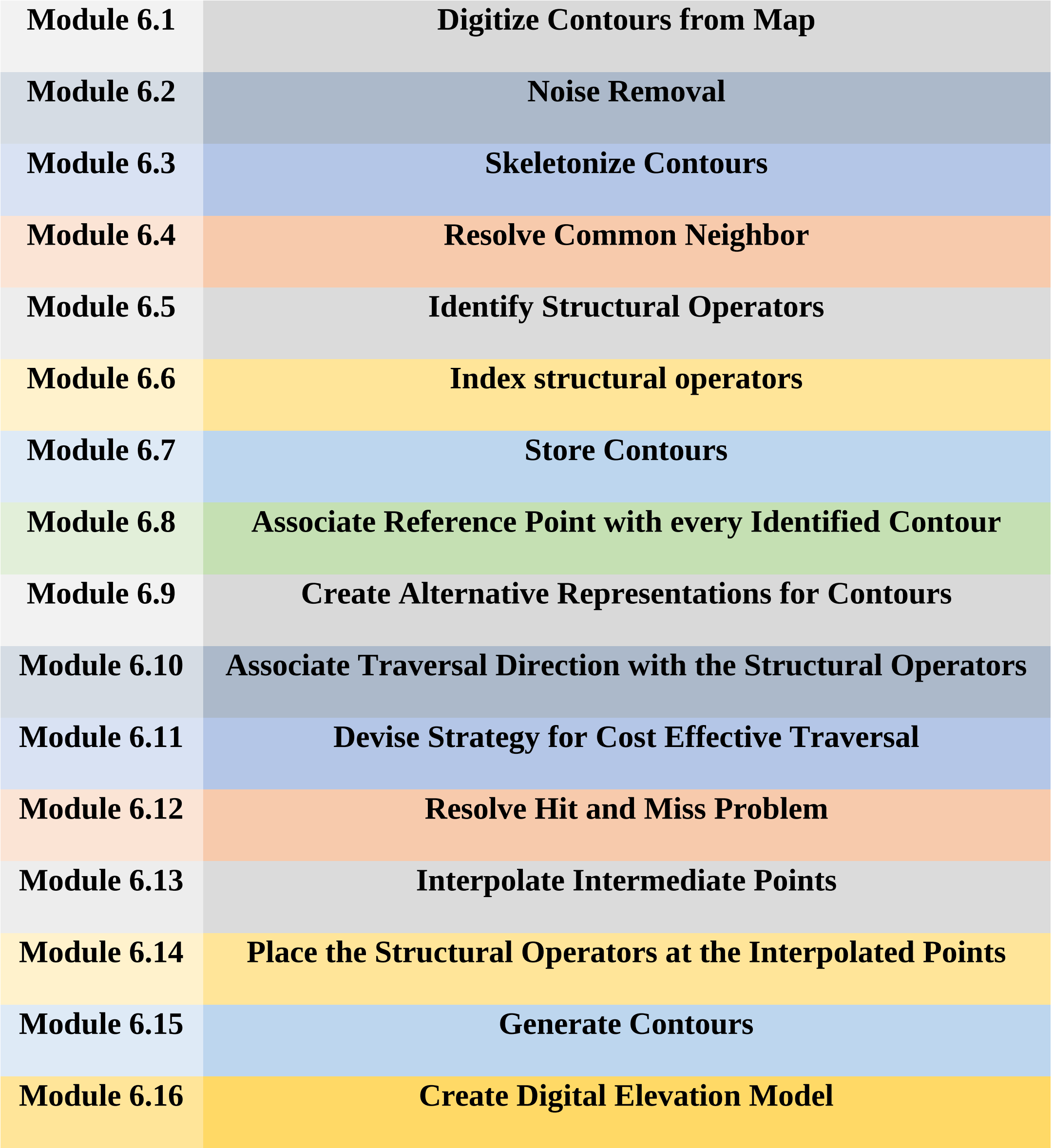
6.3 Applications of the proposed research initiative

The major drawback of contour maps is the missing information about earth surface between contours. Due to this inadequate information, it is difficult to perceive areas of interest and use existing contours information effectively in varied application areas. The generation of contours between existing contours provides more detailed information about earth landscape. Hence, all applications of contours as mentioned in Chapter 4 can be used more effectively.

The DEM derived from interpolated contour map provides better visualization with more adequate information about earth's natural surface, thereby enhancing the use in variety of applications. The DEM is mainly used for infrastructural development during urban planning and architectural design. It is used during geological survey to study and analyze geological features like landforms, rock formation and aids in mineral exploration and hazard assessment. It is also commonly used in hydroelectric projects for assessing dam locations and reservoir capacity for planning release of water. In the field of telecommunication, 3D maps are used to identify optimal locations for communications towers and antennas. It has a potential use in disaster management for planning during natural hazards like flood, landslide, earthquake, etc. The other areas of application are environmental impact assessments due to construction and development, assessment of agricultural land, forest, wildlife, etc.

6.4 Proposed Methodology

The research initiatives propose the following sequence of steps, presented in figure 6.11 (page no 265), for the attainment of the aforementioned set of objectives:



Module 6.1	**Digitize Contours from Map**
Module 6.2	**Noise Removal**
Module 6.3	**Skeletonize Contours**
Module 6.4	**Resolve Common Neighbor**
Module 6.5	**Identify Structural Operators**
Module 6.6	**Index structural operators**
Module 6.7	**Store Contours**
Module 6.8	**Associate Reference Point with every Identified Contour**
Module 6.9	**Create Alternative Representations for Contours**
Module 6.10	**Associate Traversal Direction with the Structural Operators**
Module 6.11	**Devise Strategy for Cost Effective Traversal**
Module 6.12	**Resolve Hit and Miss Problem**
Module 6.13	**Interpolate Intermediate Points**
Module 6.14	**Place the Structural Operators at the Interpolated Points**
Module 6.15	**Generate Contours**
Module 6.16	**Create Digital Elevation Model**

Figure 6.11: Framework for proposed research initiative

a) Digitize Contours from Map

In a typical topographic sheet, contours are represented using a distinctive color code for ease of feature presentation and to facilitate visual interpretation. Research initiatives relying on the use of contours for drawing valuable morphological inferences mandate the digitization of these features into a distinct thematic layer. This can be achieved by realizing an intensity-based image segmentation module. The effectiveness of the segmentation process depends heavily on the precision with which the range of intensity values representing the contours is established. Although these processes are not computationally intensive, as they have a greater reliance on the set of reference values, they must be carefully executed as discussed in algorithm 6.1. In addition, the quality of the results of the subsequent steps depends on the outcome of the digitization process. To manage the computation time, the sample topographic sheet was converted from the 3D-RGB model to the HSV color model. Likewise, the range of intensity values taken into account for segmentation was 0.0 to 0.3 for hue (H) and 0.0 to 0.7 for value (V).

Algorithm 6.1: Algo_ Digitize_Contour

Consider	Img	*Dataset*
	i_r,i_c	*row size, column size*
	Contour	*Dataset*
	b_1 b_2	*bound for threshold representing contour lines*

```
Start    temp:=rgb to gray (img)
         i_r,i_c:=size(temp)
         for i =1 to i_r do
             for j=1 to i_c do
                 if temp (i, j) ≥ $b_1$&& temp (i, j) ≤ $b_2$
                     contour (i,j)=1;
```

```
                    else
                        contour (i,j)=0;
                    End if
                End for
End         End for
```

Algorithm 6.1: Contour line extraction using thresholding operation

Figure 6.12 a) (page no 267) represents the sample contour map taken into consideration. Figure 6.12 b) (page no 267) represents the digitized sample contour map.

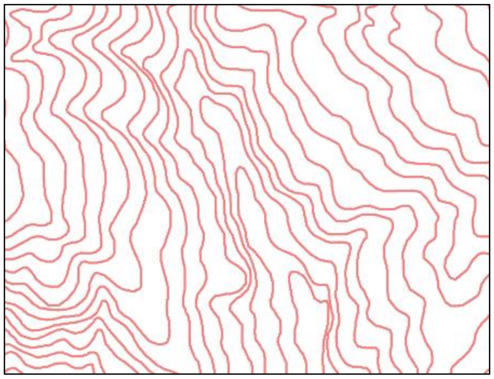


Figure 6.12 a): Contour Map

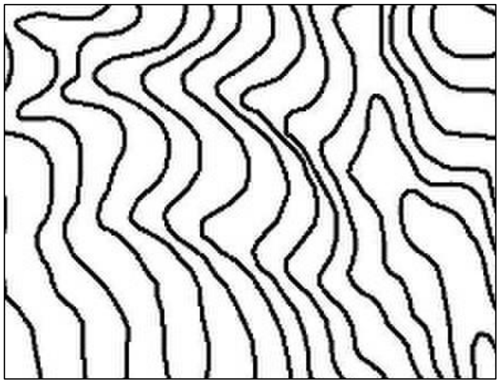


Figure 6.12 b): Segmented Contour Map

b) Noise Removal

Very often, it is observed that segmented images, in addition to significant datasets, also express noises (e.g., salt and pepper). In simple terms, noise may be perceived as non-contextual data elements that have to be removed using purposeful spatial processing. This can be achieved by using standard, well-accepted filtering techniques or by designing a simple contextual module. Here, a median filter was used to remove salt and pepper noises as detailed in algorithm 6.2. However, it is also to be remembered that these morphology filters are highly sensitive to errors. The median filter proves effective in the case of single-point errors, but its effectiveness tremendously decreases in the presence of patches [262].

Algorithm 6.2: Algo_ Remove_Noise

```
Consider  Contour       Dataset
          i_r,i_c       row size, column size
          Filter        processing variable
          filter_w      filter  width
          filter_h      filter height
          i,j,k,l,ex,ey temporary variables
Start          i_r,i_c:=size(cont)
               ex:= floor(filter_w /2)
               ey:= floor(filter_h /2)
               for i:=ex to i_r-ex do
                    for j:=ey to i_c-ey  do
                         k: = 0
                         for l:=0 to 3 do
                            for m:= 0 to 3 do
                                 filter[k] := cont[i + l - ex][j + m - ey]
                                 k:= i + 1
                        End for
                    End for
                    sort entries in filter[]
                    contour[i][j] := filter[filter_w * filter_h / 2]
               End for
End
```

Algorithm 6.2: Removal of noise

Figure 6.13 a) (page no 269) represents the sample contour map with salt and pepper noise. Figure 6.13 b) (page no 269) represents the cleaned contour map free from salt and pepper noise.

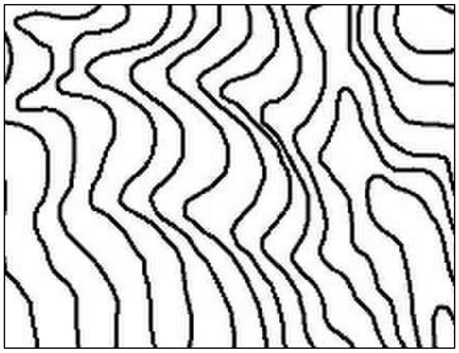


Figure 6.13 a): Before removal of noise

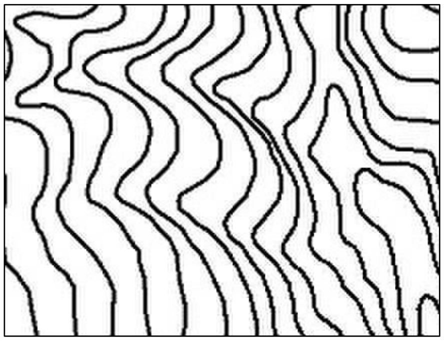


Figure 6.13 b): After Removal of noise

c) Skeletonize Contours

The complexity involved in the morphological processing of a feature reduces exponentially by condensing the dimensional information content related to the feature. This can be achieved through thinning, which enables the generation of features with single-pixel width. The proposed research initiative relies on the Zhang-Suen Thinning Algorithm [263] for creating a skeletonized (single-pixel) representation of the contours in the digitized contour map, as shown in figures 6.13 a) and b) (page no 269).

Here, the ROI in the image is considered to have a pixel value of "1," whereas the background's pixel value is "0." It is executed over 2 passes, applied to a given image, in which a contour point is represented by a value of "1" and out of 8 neighbors, at least one should have a value of "0."

Table 6.1: 3*3 Window

P_9	***P_2***	***P_3***
P_8	***P_1***	***P_4***
P_7	***P_6***	***P_5***

On implementation of the same using a 3x3 mask as shown in table 6.1 (page no 270), the 8 neighbor pixels are stored in a linear array. The idea is to identify the contour pixel that can be removed with minimum or no influence. Consider P_1 in table 6.1(page no 270) as the pixel under consideration.

In pass 1, a pixel becomes a candidate for removal, provided all of the following conditions are satisfied:

a) $2 \leq N(P_1) \leq 6$ (Count of non-zero neighbours of P_1 should be in the range of 2 to 6)

b) $S(P_1) = 1$ (Count of 0-1 transition in the ordered sequence should be exactly 1)

c) $P_2 * P_4 * P_6 = 0$ (Any one of P_2 or P_4 or P_6 should be zero)

d) $P_4 * P_6 * P_8 = 0$ (Any one of P_4 or P_6 or P_8 should be zero)

Similarly, in pass 2, a pixel becomes candidate for deletion, provided all of the following conditions are satisfied:

a) $2 \leq N(P_1) \leq 6$. (Count of non-zero neighbours of P_1 should be in the range of 2 to 6)

b) $S(P_1) = 1$ (Count of 0-1 transition in the ordered sequence should be exactly 1)

c) $P_2 * P_4 * P_8 = 0$ (Any one of P_2 or P_4 or P_6 should be zero)

d) $P_2 * P_6 * P_8 = 0$ (Any one of P_4 or P_6 or P_8 should be zero)

It is to be noted that the conditions a) and b) remain the same and Pass 1 followed by Pass 2 is used iteratively until a singly connected ROI is obtained as detailed in algorithm 6.3.

Algorithm 6.3: Algo_ Thinning

```
Consider  contour    Dataset
          row,col    row size, column size
          i,j        Variables

Start        row,col:=size(contour)
          for i=2 to row-1 do
              for j= 2 to col-1 do
                  check for conditions stated for pass 1 to determine whether pixel
                  becomes a candidate for removal
              End for
          End for
          for i=2 to row-1 do
               for j= 2 to col-1 do
                  check for conditions stated for pass 2 to determine whether pixel
                  becomes a candidate for deletion
               End for
          End for
End       repeat until no new pixel is marked for deletion.
```

Algorithm 6.3: Thinning of contour lines

Figure 6.14 a) (page no 272) represents the cleaned contour map that needs to be skeletonized using thinning morphological operators. Figure 6.14 b) (page no 272) represents the thinned contour map.

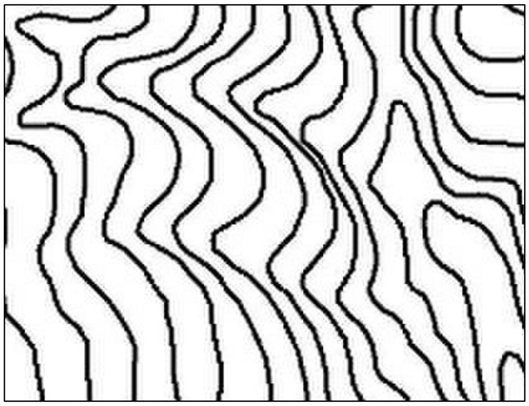


Figure 6.14 a): Before Thinning

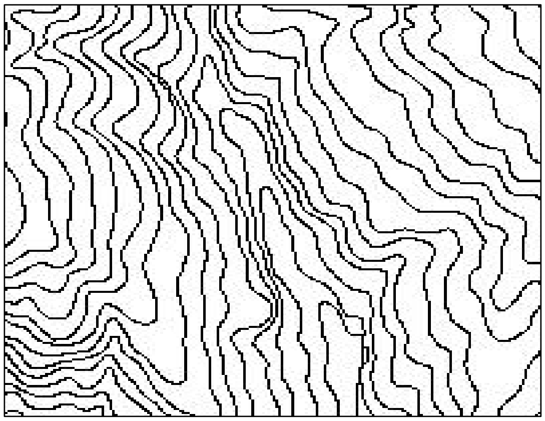


Figure 6.14 b): After Thinning

d) Resolve Common Neighbor

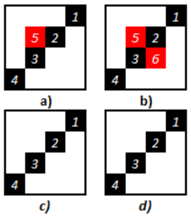


Figure 6.15: Common neighbours

On scrutiny of the processed sample contour map in some exceptional instances, it was found that there exist some coordinates with common neighbors. For example, in figure 6.15 a) (page no 272), pixels 2 and 3 have a common neighbor pixel 5, whereas in figure 6.15 b) (page no 272), pixels 2 and 3 have common neighbors pixel 5 and 6. Although such structural organizations pose no hindrances, they would increase the number of structural orientations and inherently increase the complexity of computational time. Eradicating such coordinates would ease computation and help in constricting the number of structural operators. Figure 6.15 c) & d) (page no 272) represents the desired pixel organization.

For any significant pixel with two neighbors P and Q, the m-Connectivity is defined as in equation 6.1:

$$N_4(P) \cap N_4(Q) \neq \text{NULL} \quad \text{………………. (Equation 6.1)}$$

where,

- $N_4(P)$ represents four neighborhood of significant pixel P (top, down, left, and right pixel)
- $N_4(Q)$ represents four neighborhood of significant pixel Q (top, down, left, and right pixel)
- i.e., there should not be any common N_4 elements between pixel P and Q.

Algorithm 6.4: Algo_ Resolve Neighbor

```
Consider  contour    Dataset
          i_r,i_c    row size and column size
          tmp        window of size 3x3

Start     for i=2 to i_r-1 do
              for j=2 to i_c-1 do
                 For any significant pixel with two neighbors P and Q ensure $N_4(P) \cap N_4(Q)$=
                 NULL where $N_4(P)$ & $N_4(Q)$ represents four neighborhood of significant pixel
End           End for
          End for
          save and display contour
```

Algorithm 6.4: Removal of common neighborhood for ensuring m-connectivity

Figures 6.15 a) and b) (page no 272) represent the contour maps where possible instances of m-connectivity need to be resolved. Figures 6.15 c) and d) (page no 272) represent the contour map that ensures m-connectivity achieved with the help of algorithm 6.4.

e) Identify Structural Operators

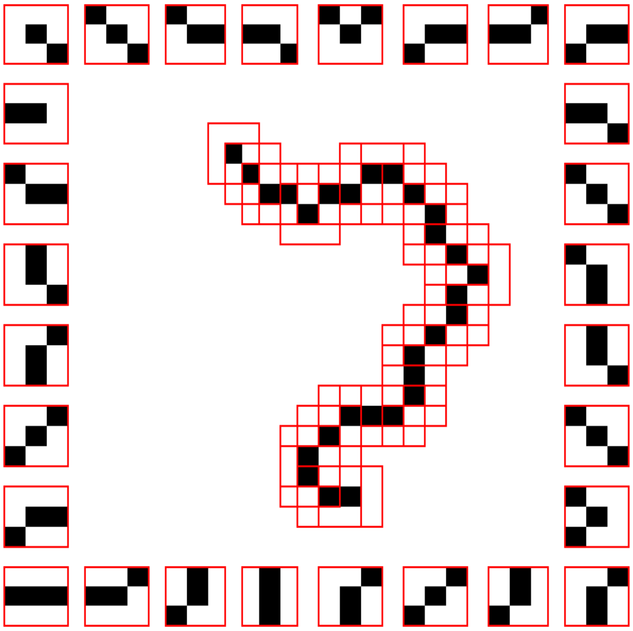


Figure 6.16: Structural Operators of 3x3

In the context to the proposed research initiative, a structural operator refers to a possible orientation of a pixel in a matrix of predefined order; here, 3x3 was considered the ideal size for realizing the structural operator. It is possible to have variable-sized structural operators, and it was observed that increasing the size of the structural operators would increase the pixel combinations, leading to a larger number of structural operators to handle and consequently greater complexity. Figure 6.16 (page no 274) represents a small sample segment of a contour, with 3x3 as the size of the operator. 28 such orientations were extracted, centered at every significant pixel. It may also be possible that some of the operators are repeated as a consequence of the contour alignment. In order to find all possible structural operators that may be important in the context of contours, the proposed research project looked at nine sample reference images and used an unsupervised search process to find all unique 3x3 orientations centered at the significant pixel location. Figure 6.17 (page no 275) highlights 24 unique structural operators determined by the search process from the set of sample reference images taken into consideration as detailed in algorithm 6.5.

Algorithm 6.5: Algo_ Identify Structure Operator

```
Consider   contour      Dataset
Start      for i=1 to n  do
               for  j=2 to row-1 do
                   for  k=2 to col-1 do
                      extract every unique 3x3 orientation centered at a significant position
                             and store the same for reference
           End for
               End for
           End for
End        save and display count and operator
```

Algorithm 6.5: Extraction of all possible morphological operators

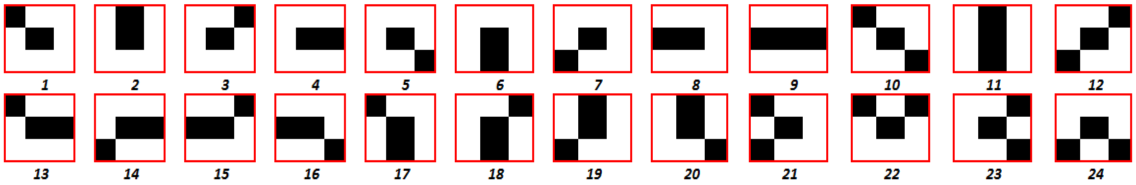


Figure 6.17: Structural operators of size 3x3 with numeric tags 1-24 [230]

f) Index Structural Operators

It is quite essential to assign unique identity tags to every structural operator identified by the unsupervised search process for ease of onward operations.

Algorithm 6.6: Algo_ Assign Tag to Structure Operator

```
Consider   Contour          Dataset
           i, j, k, count   variables, count track of the number of operator
           Operator         3D variable of (3x3) mask x n operators
Start      for i=0 to count do
```

```
              assign tag (i+1) to operator (i)
End      End for
```

Algorithm 6.6: Assignment of tag to morphological operators

Here, the operators were assigned numeric tags from 1 to 24. The structural operators highlighted in figure 6.16 (page no 274) are represented in figure 6.17 (page no 275) with numeric tags with the help of algorithm 6.6. Every structural operator is given one of these numeric tags to make it easier to link the direction of traversal based in its morphological orientation in relation to certain feature-specific reference points.

g) Store Contours

The elementary details of each and every contour present in the sample reference map are to be recorded in a suitable data structure for facilitating operations. The elementary details include contour id, type of contour (line or polygon), (x, y) coordinates of the contour elements, description of the structural operators along with the coordinates, direction associated with the structural operators along the coordinates, coordinate count, elevation details, and contour reference.

Algorithm 6.7: Algo_ Generate Attribute

```
Consider contour   dataset
Start    for every detected contour do
              store contour details
End      End for
```

Algorithm 6.7: Extraction of contour attributes

Contour details include contour id, contour type (point, line, or polygon), elevation value of the contour, centroid value, coordinate trail of the contour, coordinate count, direction of traversal, and count of contour instances, achieved through algorithm 6.7.

h) Associate Reference Point with every Identified Contour

The interpolation operation mandates the determination of two endpoints, or two end coordinates; thereafter, desired portioning of the distance between these two points would enable placement of the interpolated points. A suitable distance matrix, such as the Euclidian distance, may be employed for determining the distance. Among all the inherent tasks involved, the task of deciding where to search for the cooperating coordinate for interpolation from a known coordinate point requires serious attention and consideration. In addition, the same is to be achieved with reliance to certain well-established references. In this research initiative, the centroids of the various aspects are taken as reference points.

The proposed research initiative aims at generating interpolated contours in between two types of contour elements, which are lines and enclosures. With regards to interpolation in between lines, the centroid of the line and the centroid of the entire data set are taken into consideration for deciding on the direction of traversal, whereas with regards to interpolation in between enclosures, the coordinates of the enclosure and the centroid of the individual enclosure are taken into account. The detailed process for interpolation is explained in algorithm 6.8.

Algorithm 6.8: Algo_ Contour Reference Point

```
Consider    contour    Dataset
            cont_cent
Start       for every detected contour do
                store $cont\_cent_i = \sum cont\_x_i / n$ , $\sum cont\_y_i / n$,  n coordinate instances
End         End for
```

Algorithm 6.8: Associate Reference Point with every Identified Contour using mean

i) Create Alternative Representations for Contours

To facilitate easy interpolation, the significant pixels are suitably replaced by numeric tags representing the structural orientation listed in figure 6.17 (page no 275). This transient step is implemented to simplify the traversal and eventually the interpolation process. Here, figure 6.18 (page no 278) is a representation of figure 6.16 (page no 274) with numeric tags with the help of algorithm 6.9.

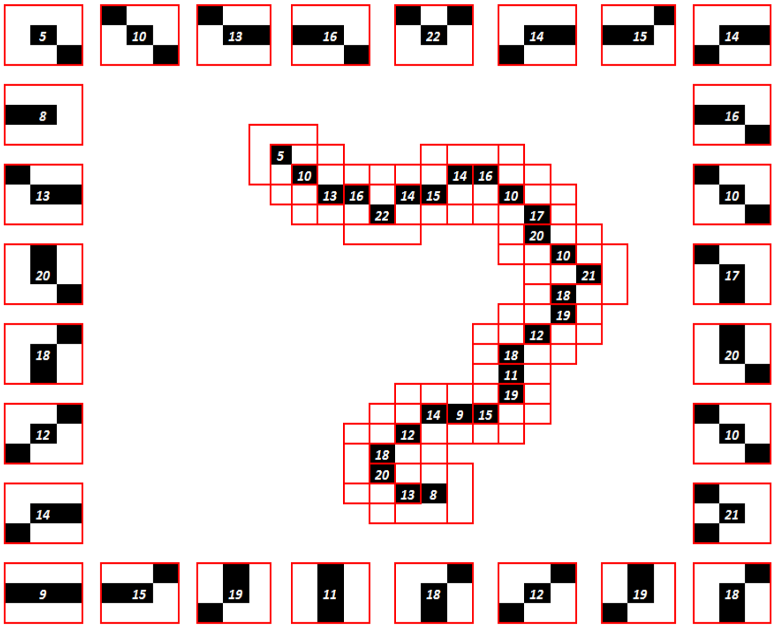


Figure 6.18: Structural operators with numeric tags

Algorithm 6.9: Algo_Alternative Representations

```
Consider    contour    Dataset
------------------------------------------------------------
Start       for every detected contour do
                for every detected structural operator do
                    replace every  center pixel of 3x3 window representing the structural
                    operator with the numeric tag associated with it
                End for
End         End for
```

Algorithm 6.9: Assignment morphological operator identity in contours

j) Associate Traversal Direction with the Structural Operators

The detection of cooperating coordinates for interpolation mandates the association of the direction of traversal with each and every identified structural operator listed in figure 6.18 (page no 278) based on its positioning relative to the respective reference point.

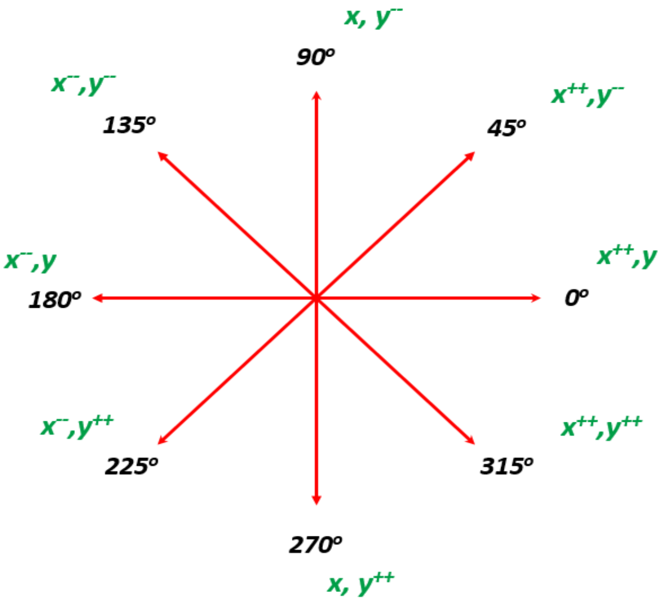


Figure 6.19: Direction of traversal.

With regards to a given position, identifiably, there are eight possible directions for traversal, i.e., along0^{o}, 45 ^{o}, 90 ^{o}, 135 ^{o}, 180 ^{o}, 225 ^{o}, 270 ^{o} and 315 ^{o} respectively, as represented in figure 6.19 (page no 279) and table 6.2 (page no 280).

i) Traversal in case of lines:

Here, the centroid of the entire image and the centroid of the line are taken as the reference points for determining the direction of traversal for each and every structural operator represented in figure 6.17 (page no 275) that exists in the contour line.

Table 6.2: Operations along the various directions [230]

Sl	Direction	x-coordinate	y-coordinate
1	0^{o}	++	No Op
2	45 ^{o}	++	--
3	90 ^{o}	No Op	--
4	135 ^{o}	--	--
5	180 ^{o}	--	No Op
6	225 ^{o}	--	++
7	270 ^{o}	No Op	++
8	315 ^{o}	++	++

Consider,

- X_c, Y_c is the centroid of the image
- X_{cl}, Y_{cl} is the centroid of the line

There may be two distinct relationship that may be established in between X_c, Y_c and X_{cl}, Y_{cl} to decide on the direction of traversal with the structural operators based on whether $Y_c \geq Y_{cl}$ ***or*** $Y_c < Y_{cl}$.

Case 1: When $Y_c \geq Y_{cl}$

The direction of traversal is represented in figure 6.20 (page no 281) and table 6.3 (page no 281).

Table 6.3: Assignment of direction with the operators when $Y_c \geq Y_{cl}$ [230]

Sl. No.	Operation	Direction
1	1	45 °
2	2	0° or 180 °
3	3	135 °
4	4	90 °
5	5	45 °
6	6	0° or 180 °
7	7	135 °
8	8	90 °
9	9	90 °
10	10	45 °
11	11	0° or 180 °
12	12	135 °
13	13	45 °
14	14	135 °
15	15	135 °
16	16	45 °
17	17	45 °
18	18	135 °
19	19	135 °
20	20	45 °
21	21	0° or 180 °
22	22	90 °
23	23	0° or 180 °
24	24	90 °

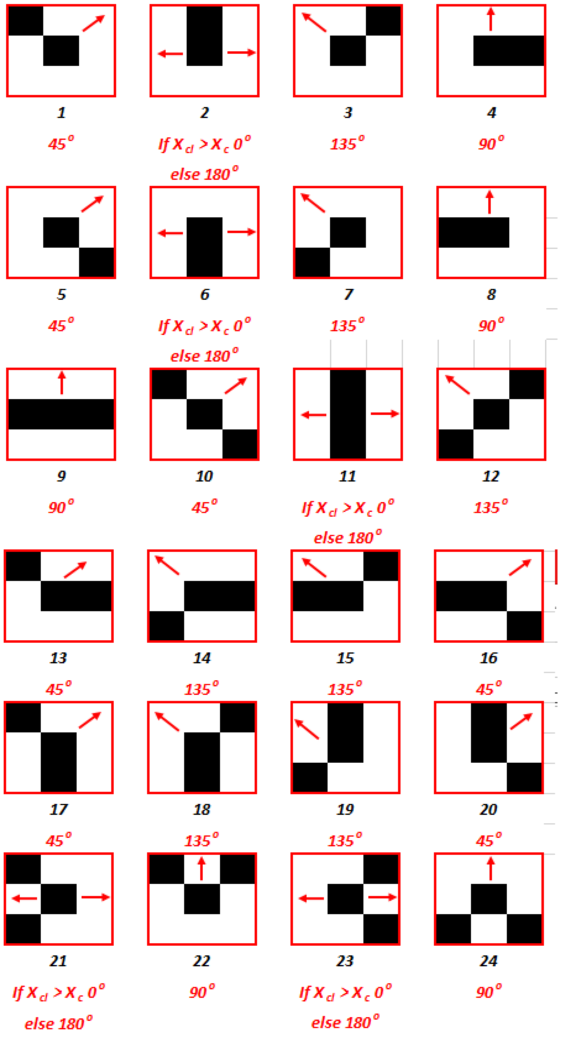


Figure 6.20: Association of direction with the structural operators when $Y_c \geq Y_{cl}$ [230]

Case 2: When $Y_c < Y_{cl}$

The direction of traversal is represented in figure 6.21 (page no 282) and table 6.4 (page no 282).

Table 6.4: Assignment of direction with the operators when $Y_c < Y_{cl}$ [230]

Sl.	Operation	Direction
1	1	225°
2	2	0° or 180 °
3	3	315 °
4	4	270 °
5	5	225°
6	6	0° or 180 °
7	7	315 °
8	8	270 °
9	9	270 °
10	10	225°
11	11	0° or 180 °
12	12	315 °
13	13	225°
14	14	135 °
15	15	315 °
16	16	225°
17	17	225°
18	18	315 °
19	19	315 °
20	20	225°
21	21	0° or 180 °
22	22	270 °
23	23	0° or 180 °
24	24	270 °

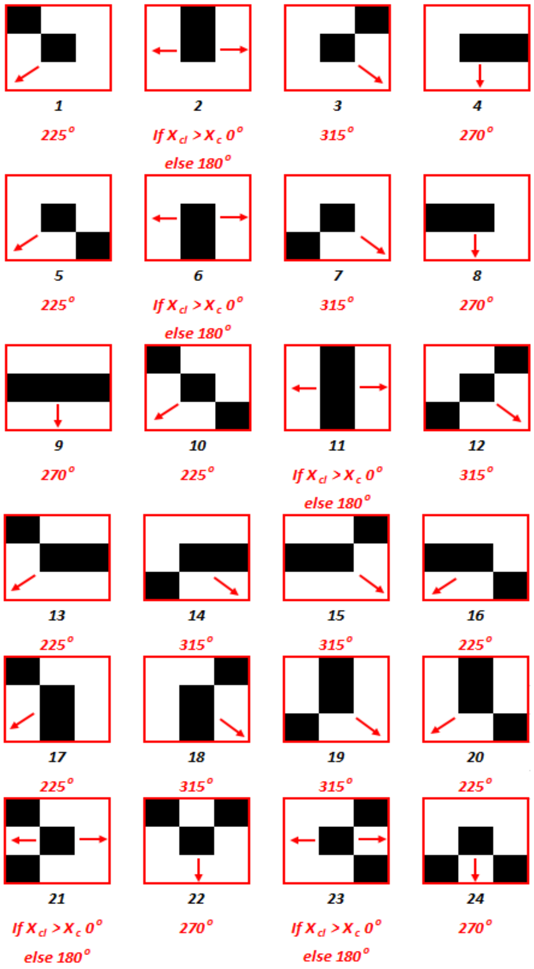

1
225°
2
If $X_{cl} > X_c$ 0° else 180°
3
315°
4
270°
5
225°
6
If $X_{cl} > X_c$ 0° else 180°
7
315°
8
270°
9
270°
10
225°
11
If $X_{cl} > X_c$ 0° else 180°
12
315°
13
225°
14
315°
15
315°
16
225°
17
225°
18
315°
19
315°
20
225°
21
If $X_{cl} > X_c$ 0° else 180°
22
270°
23
If $X_{cl} > X_c$ 0° else 180°
24
270°

Figure 6.21: Association of direction with the structural operators when $Y_c < Y_{cl}$ [230]

ii) Traversal in case of enclosures

Here, the centroid of the individual enclosure and the coordinate positioning of the structural operator in the particular enclosure are taken into consideration for determining the direction of traversal to be associated with the particular structural operator represented in figure 6.17 (page no 275).

Consider,

- X_{ce}, Y_{ce} is the centroid of the enclosure
- X, Y coordinate position of the structural operator

There may be two distinct relationship that may be established in between X, $Y$and X_{ce}, $Y_{ce}$to decide on the direction of traversal with the structural operators based on whether $Y_{ce} \geq Y$or $Y_{ce} < Y$.

Case 3: When $Y_{ce} \geq Y$

The direction of traversal is represented in figure 6.22 (page no 284) and table 6.5 (page no 284).

Table 6.5: Assignment of direction with the operators when $Y_{ce} \geq Y$ [230]

Sl. No.	Operation	Direction
1	1	45^{o}
2	2	0^{o} or 180^{o}
3	3	135^{o}
4	4	90^{o}
5	5	45^{o}
6	6	0^{o} or 180^{o}
7	7	135^{o}
8	8	90^{o}
9	9	90^{o}
10	10	45^{o}
11	11	0^{o} or 180^{o}
12	12	135^{o}
13	13	45^{o}
14	14	135^{o}
15	15	135^{o}
16	16	45^{o}
17	17	45^{o}
18	18	135^{o}
19	19	135^{o}
20	20	45^{o}
21	21	0^{o} or 180^{o}
22	22	90^{o}
23	23	0^{o} or 180^{o}
24	24	90^{o}

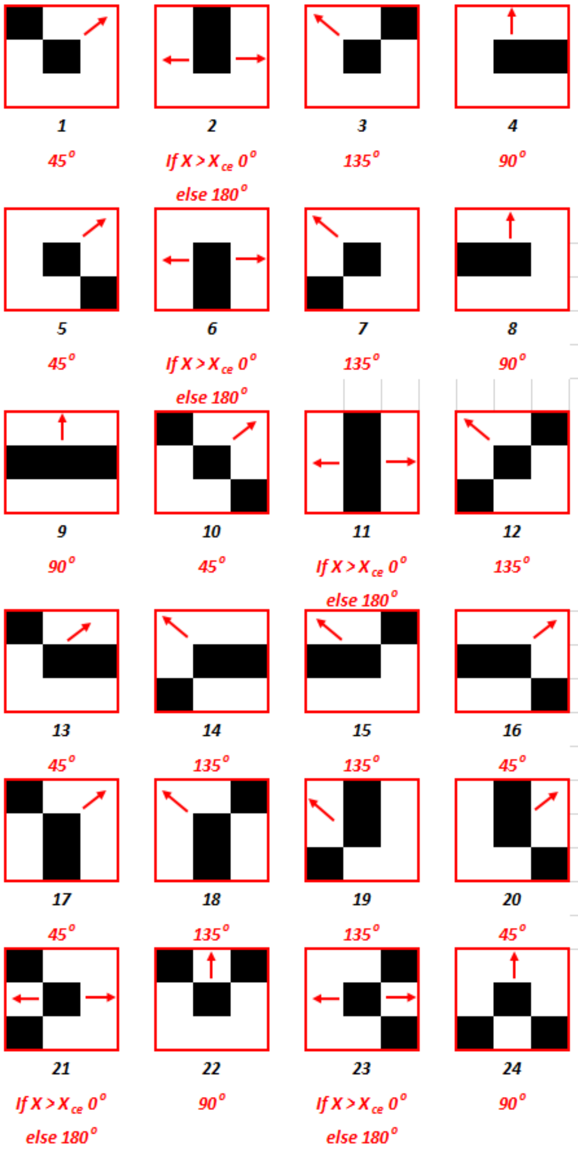


Figure 6.22: Association of direction with the structural operators when $Y \geq Y_{cl}$ [230]

Case 4: When $\boldsymbol{Y_{ce} < Y}$

The direction of traversal is represented in figure 6.23 (page no 285) and table 6.6 (page no 285).

Table 6.6: Assignment of direction with the operators when $Y_{ce} < Y$ [230]

Sl.	Operation	Direction
1	1	225^{o}
2	2	0^{o} or 180^{o}
3	3	315^{o}
4	4	270^{o}
5	5	225^{o}
6	6	0^{o} or 180^{o}
7	7	315^{o}
8	8	270^{o}
9	9	270^{o}
10	10	225^{o}
11	11	0^{o} or 180^{o}
12	12	315^{o}
13	13	225^{o}
14	14	135^{o}
15	15	315^{o}
16	16	225^{o}
17	17	225^{o}
18	18	315^{o}
19	19	315^{o}
20	20	225^{o}
21	21	0^{o} or 180^{o}
22	22	270^{o}
23	23	0^{o} or 180^{o}
24	24	270^{o}

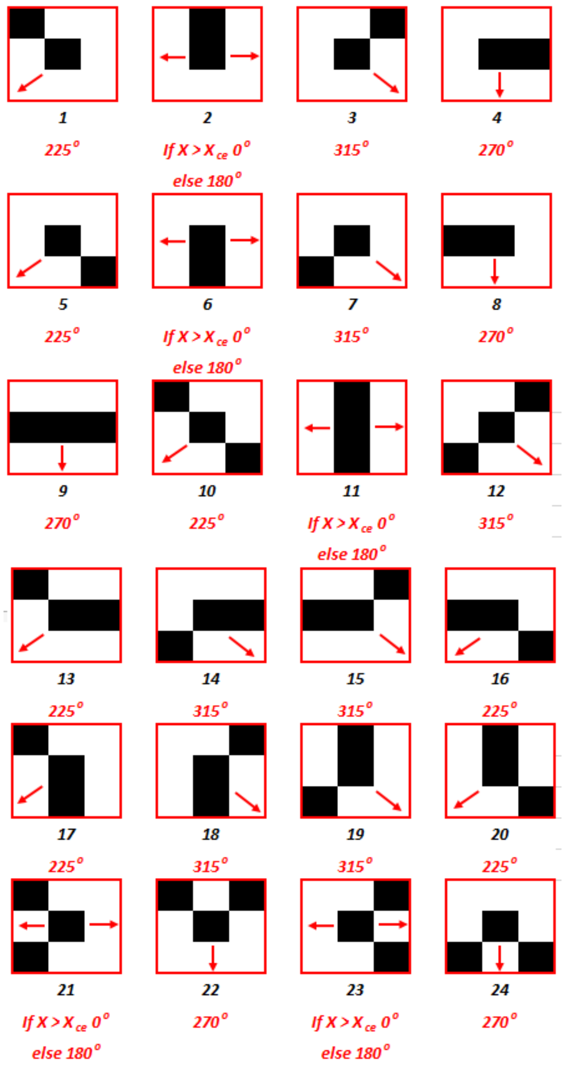


Figure 6.23: Association of direction with the structural operators when $\boldsymbol{Y_c < Y_{cl}}$ [230]

To enable the traversal module to effectively perform the desired operations, the knowledge regarding the traversal directions associated with the structural operators specified in tables 6.2–6.6 needs to be stored for future program references as detailed in algorithm 6.10.

Algorithm 6.10: Algo_Associate Traversal Direction

```
Consider  contour    Dataset
Start     for every detected contour do
              for every detected structural operator do
                  based on type of contour (line or enclosure), structural operators and Table 6.3, 6.4, 6.5 and 6.6 associate direction with structural operators
              End for
End       End for
```

Algorithm 6.10: Assignment of directional information for matched contour points

k) Devise Strategy for Cost Effective Traversal

Consider the sample contour line shown in figure 6.24 (page no 287), where with each significant coordinate a numeric tag is associated, representing the structural operator. Now, based on the criteria stated in serial j above, the direction for traversal is to be associated.

On closely analyzing the numeric tags associated with contour lines, it was observed that there might be instances wherein there might be multiple occurrences of the same numeric tag in sequence. In such situations, the direction may be associated with the numeric tags only when there is a transition from one numeric tag to another. As in the case of figure 6.24 (page no 287), there are multiple instances of 11, so direction may be associated with the structural operators only when there is a transition in the numeric tag, as represented in figure 6.25 (page no 287), with the help of algorithm 6.11.

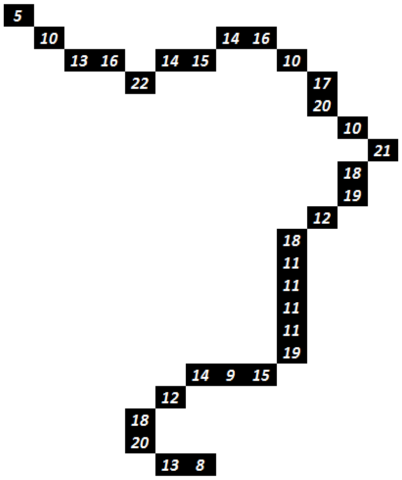


Figure 6.24: Sample contour line with numeric tags

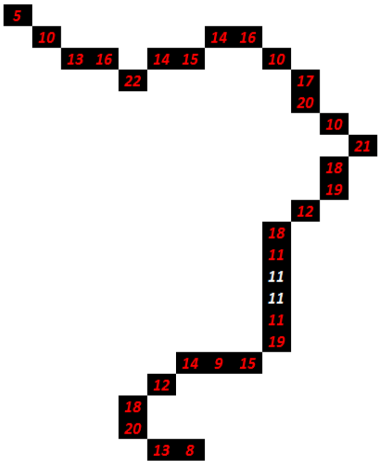


Figure 6.25: Selection of numeric tags for assignment of direction

Algorithm 6.11: Algo_Associate Traversal Direction

```
Consider   contour   Dataset

Start      for every detected contour do
               for every detected structural operator do
                   if there exist two neighbors with same numeric tag as that of
                   the significant pixel
                       continue
                   else
                       assign direction for traversal
                 End if
```

```
                    End for
End             End for
```

Algorithm 6.11: Optimization of traversal operation

l) Resolve Miss Problem

The objective of traveling along any of the 8 identified directions listed in table 6.2 (page no 280) is to hit any significant pixel in the adjacent contour. In doing so, the distance between the contours may be suitably portioned to place interpolated points.

It was observed that the probability of a hit in the case of traversal along with directions 0^o, 90^o, 180^o and 270^o is 1 whereas the probability of a hit in the case of traversal along with directions 45^o, 135^o, 225^o and 315^o is 0.5. In the situation where the probability is only 0.5 the traversal may result in the selection of inappropriate coordinate points, leading to disturbances in the morphological structure of the generated contour lines, as explained below.

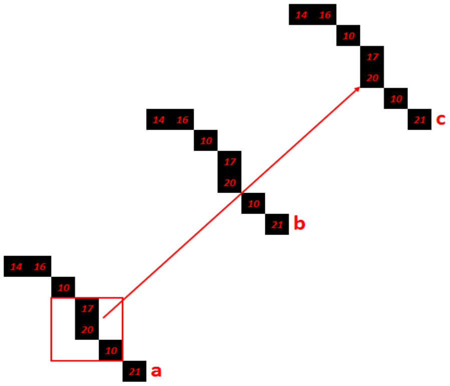


Figure 6.26: Problem of miss [230]

As represented in figure 6.26 (page no 288), the traversal from the structural operator centered at 20 in contour 'a' should have hit any of the coordinates in contour 'b' but due to the unfavourable orientation of the coordinates in contour 'b', none of it could be hit. On further traversal, one of the

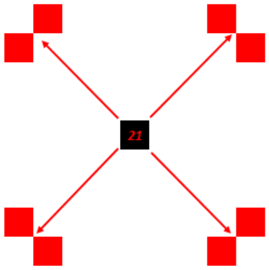


Figure 6.27: Search space [230]

coordinates of contour ‘c’ is hit. Thus, the generation of interpolated points taking into account coordinates from contours ‘a’ and ‘b’ would disturb the morphological orientation of the generated contour. To avoid such a situation, special masks may be conceived as presented in figure 6.27 (page no 288) to ensure that the probability of a hit is 1 as discussed in algorithm 6.12.

Table 6.7 (page no 289) represents the locations to search for, in situation, if a coordinate is indexed with x, y and the traversal mandated is along the diagonals.

Table 6.7: Locations to search for if coordinate is indexed with (x, y)

Sl	Direction	Coordinate to search	Sl	Direction	Coordinate to search
1	45^o	x+1, y and x, y-1	5	0^o	x+1, y
2	135^o	x-1, y and x, y-1	6	90^o	x, y-1
3	225^o	x-1, y and x, y+1	7	180^o	x-1, y
4	315^o	x+1, y and x, y+1	8	270^o	x, y+1

Algorithm 6.12: Algo_Resolve Miss

```
Consider   contour   Dataset
Start      for every detected contour do
               for every detected structural operator do
           if there traversal is along diagonal
                         look for the locations listed in table 1, 2, 3 & 4
                   else
                         look for the locations listed in table 5, 6, 7 & 8
                   End if
               End for
End          End for
```

Algorithm 6.12: Resolve hit and miss problem by morphological operator traversal

m) Interpolate Intermediate Points

On a successful hit traveling along any of the 8 identified directions, coordinate values of interpolation are to be generated in between the two cooperating coordinates. This is achieved by applying the following formula:

$$\mathbf{x_r = (x_i+x_j)/2} \text{ ………………. } \textbf{(Equation 6.2)}$$

$$\mathbf{y_r = (y_i+y_j)/2} \text{ ………………. } \textbf{(Equation 6.3)}$$

Let x_i, y_i and x_i, y_i be the two cooperating coordinate and x_i, y_i be the interpolated coordinate.

Figure 6.28 a) (page no 290) represents input sample contour map, figure 6.28 b) (page no 290) represents generation of interpolated points, and figure 6.28 c) (page no 290) represents superimposition of figure 6.28 a) and figure 6.28 b).

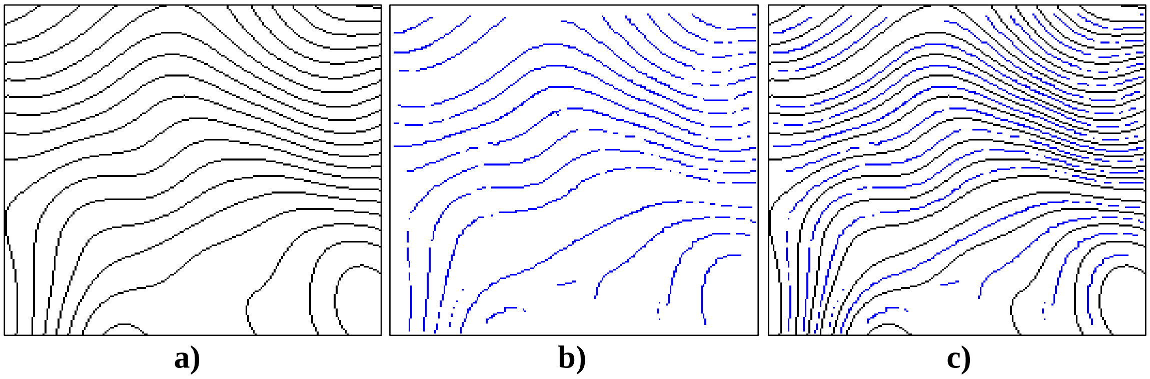


Figure 6.28: a) Represents sample contour map b) Represents interpolated points c) Sample contour map combined with the interpolated points.

n) Place the Structural Operators at the Interpolated Points

On determining the interpolated coordinate, the structural operator resembling the source contour is suitably placed at the interpolated coordinate wherever found suitable, and the process is repeated for all the directional traversals possible for a given contour (line or enclosure).

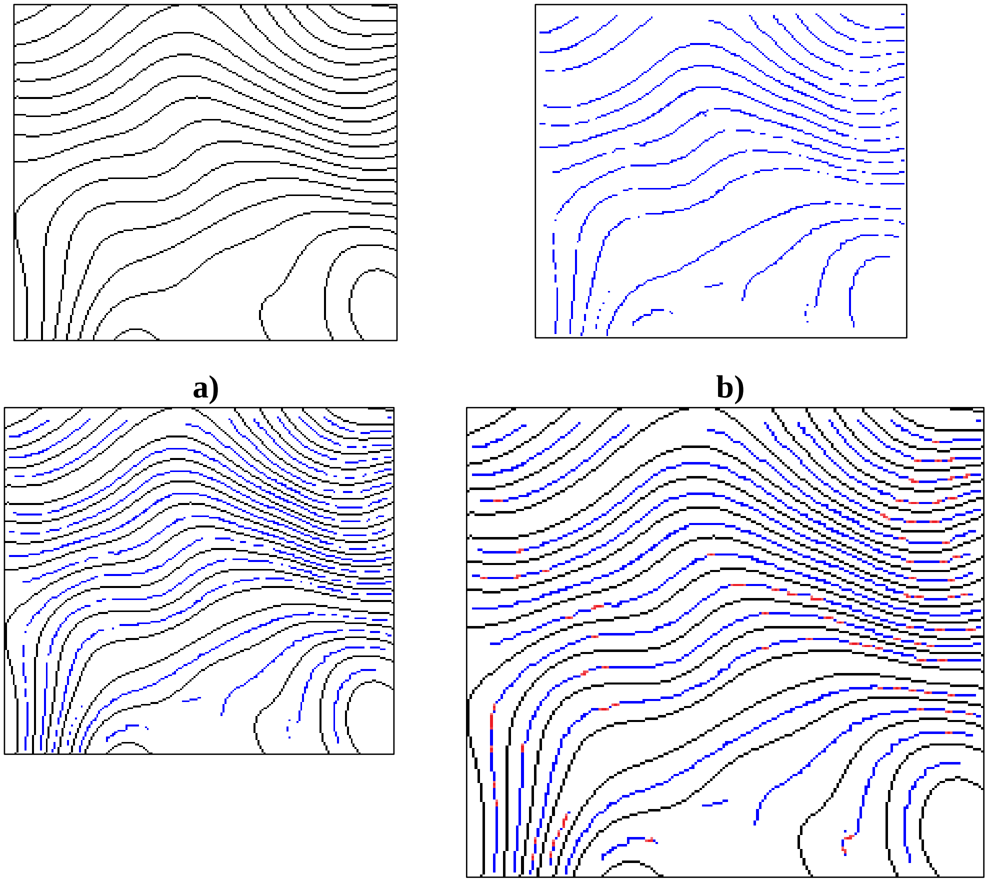
o) Generate Contours

The structural operators placed in between the contour lines are then connected to build a continuous contour (line or enclosure). To ensure preservation of morphological characteristics, a gradient-based approach has been pursued. To ensure continuity of the broken contour lines, the following sequence of steps has been applied:

- Identify all the terminal points (Terminal point is location on contour line, in which, number of neighbor is exactly one)
- Determine gradient for every terminal points
- Determine optimal pair based on minimal distance between terminal points and the positive-negative gradient of the respective terminal points
- Apply Straight line drawing algorithm (Bresenhem's Line Drawing) between optimal pairs

The reconnection between two terminal points of a broken contour is realized using the straight light drawing algorithm. The adopted techniques work on the principle of gradient calculation and then determine an optimal pair for reconnection, considering positive and negative gradients. Figure 6.29 d) (page no 292) highlights the ability of the proposed technique to establish contour continuity (highlighted with the help of red pixels). Although found effective, there are many inherent challenges faced by the technique, such as:

- The reconnection is not possible if the technique fails to identify the optimal pair.
- In situations where the breakage is severe, the technique fails to either identify the optimal pair or gets wrongly connected to another contour line.
- In situations where the inter-contour distance is very short, the chances of getting wrongly connected to another contour line greatly increase.

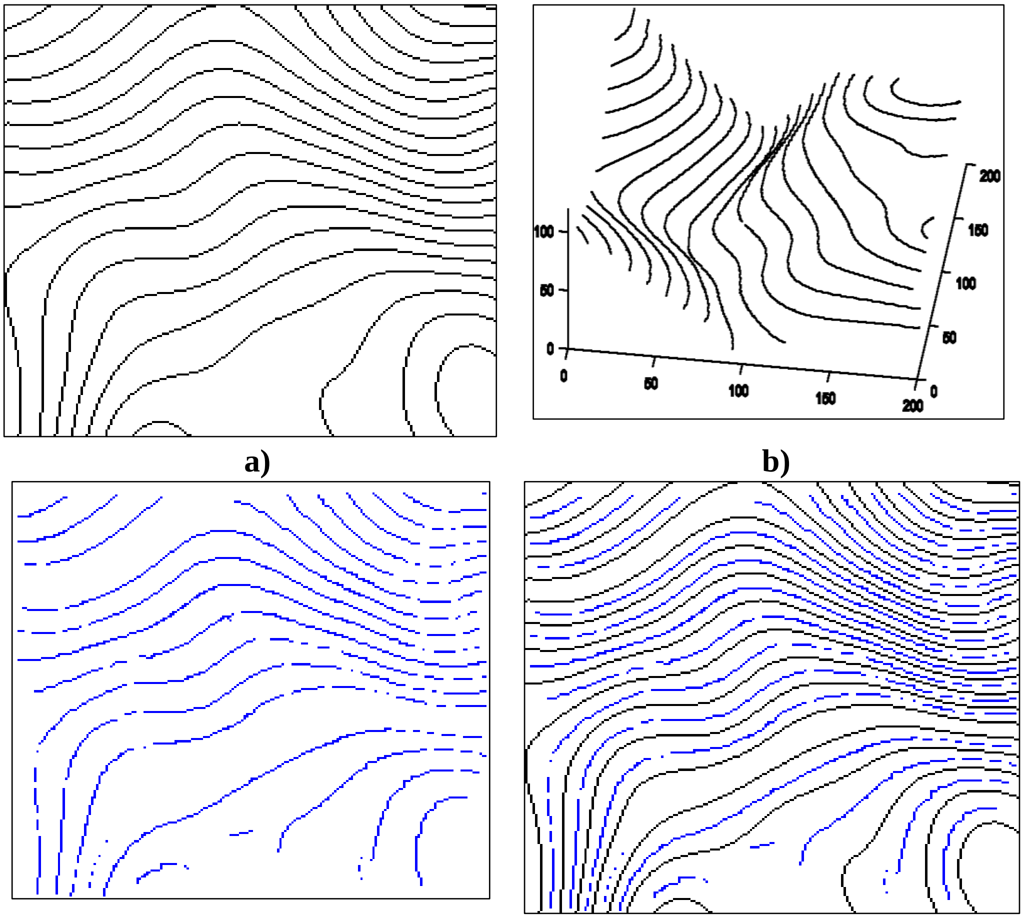


a) b) c) d)

Figure 6.29: a) Sample contour map b) Interpolated points c) Sample contour map combined with the interpolated points d) Creation of continuous contours

p) Create Digital Elevation Model

On successfully achieving steps a)–o), the respective elevation value may be associated with each of the identified contours. Subsequently, the contours may be elevated in the z-plane to have a graphical view of the 3D elevation model, as represented in figure 6.30 f) (page no 294).

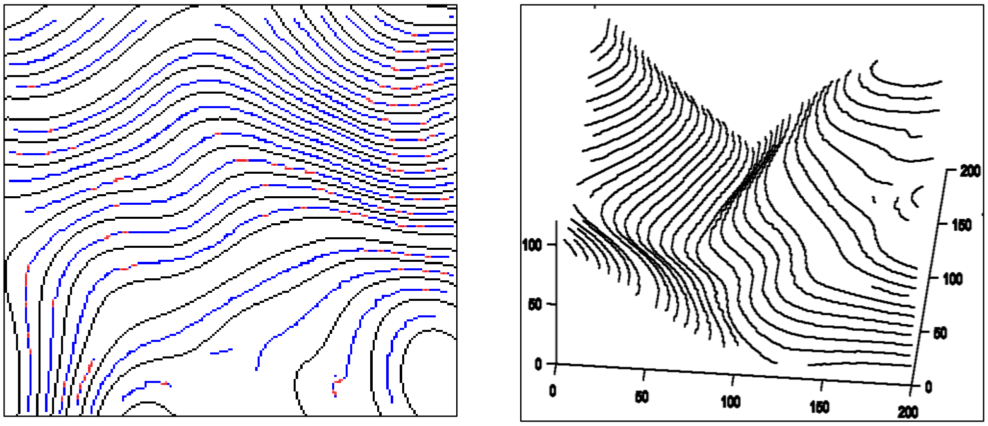


a)

b)

c)

d)



e) **f)**

Figure 6.30: a) Sample contour map b) Elevation model for sample contour map c) Interpolated points d) Sample contour map combined with the interpolated points e) Creation of continuous contours f) Elevation model for the refined contour map

6.5 Considerations, Constraints, Development Environment

6.5.1 Considerations

The input reference map for the proposed work was taken from the University of Texas, Austin. The scale of representation for the map was 1:250000. A sizable portion of a 200 x 200 subset consisting of six images covering a varied landscape was created from the reference map and set as input. The sample input image type is BMP (bitmap).

6.5.2 Constraints

The following are the constraints to implement the research initiatives:

- The feature of interest should hold the properties of connected components.
- The sample image must be a binarized image, free from noise (particularly salt-and-pepper noise).

- The feature of interest was skeletonized to create a single-pixel-width representation for ease of complexity management and to ensure m-connectivity between any two pixels in the neighborhood.
- The proposed methodology is currently capable of efficiently handling datasets of size 200 x 200 and can be further extended to work with variable-sized data.
- Any breakage in refined contours, including interpolated contours, must be connected using the reconnection technique.

6.5.3 Development Environment

The description of the development environment for the proposed research initiative is as detailed below:

Experimental Configuration	**Description**	**Criteria for Selection**
• Processor	11th Gen Intel(R) Core i7-1165G7 @ 2.80GHz	• Facilitates faster execution of programming code
• RAM	16.0 GB (15.8 GB usable)	• Sustain applications requirements
• Operating System	Windows, 64-bit	• User friendly interface • Compatibility with application
• Development Software	MATLAB (2020a)	• Easy translation of concepts to executables • Debugging ease

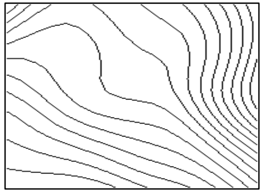
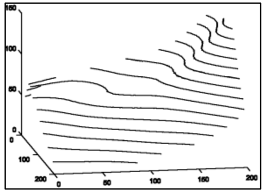
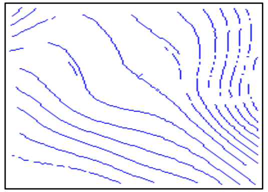
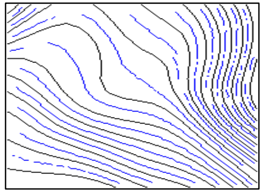
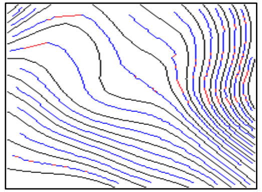
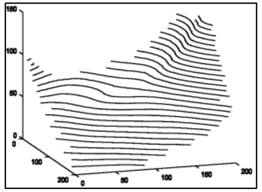
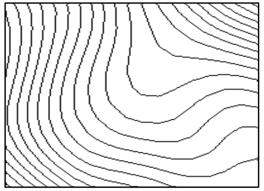
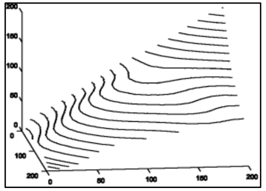
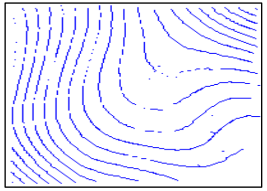
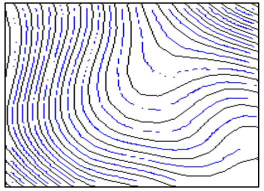
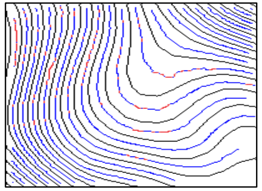
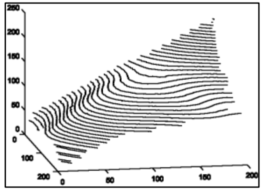
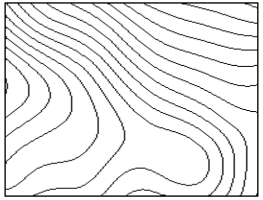
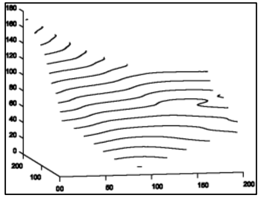
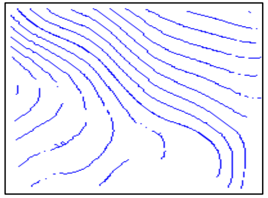
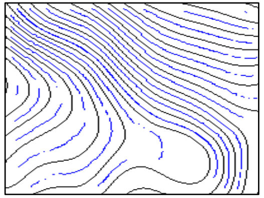
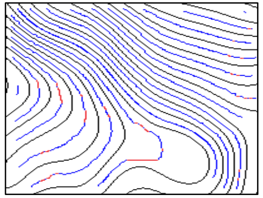
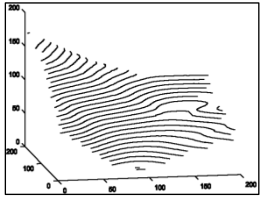
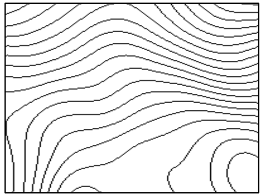
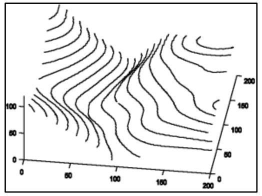
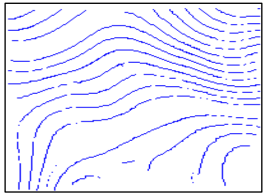
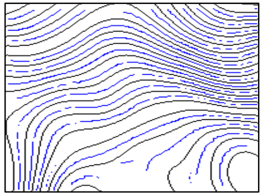
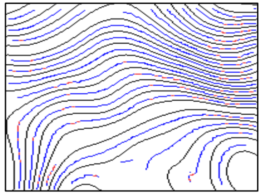
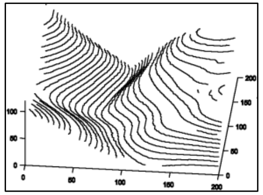
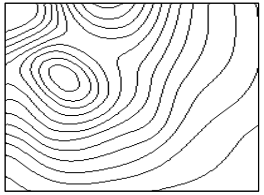
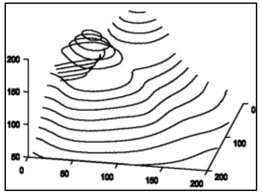
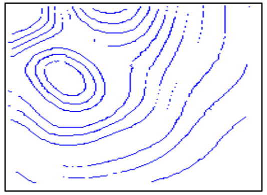
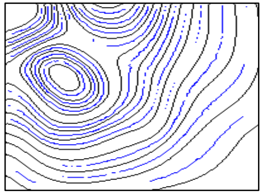
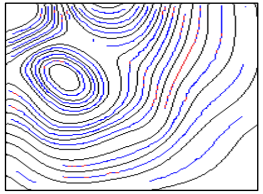
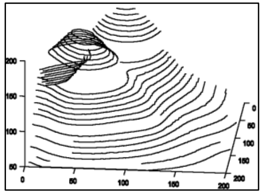
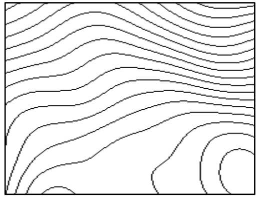
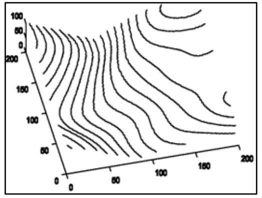
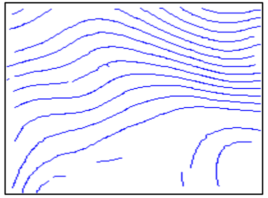
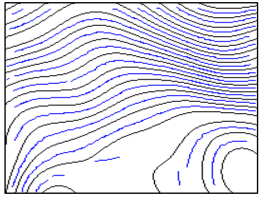
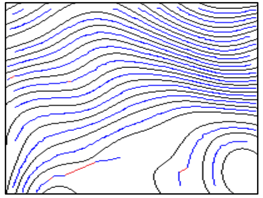
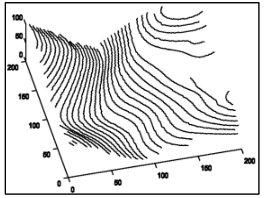
- Scalable
- Dynamic 3D Visualization of Features

6.6 Result and Discussion

The proposed methodology was tested with a variety of sample images selected from the different contour maps, six of which are represented in column A of table 6.8 (page no 298). Column B represents the corresponding elevation model, assuming a set of assigned elevation values. It is evident that the elevation model expresses greater slope and sharpness due to a larger inter-contour distance. The proposed work successfully interpolates necessary points in between the existing contour lines based on directional traversal associated with various identified structural operators, considering certain crucial reference points. The overall efficiency of the directional traversal depends on its ability to identify the desired ends of interpolation. In situations where the strategy fails, the efficiency reduces to 50%; therefore, an advanced mask has been developed to ensure the probability of a miss is 0. Column B in table 6.8 (page no 298) highlights the achievements made by the interpolation strategy deployed, where points are automatically generated, completely relying on the knowledge of the existing contour points and the conceived directional traversal. The interpolated points are represented with the help of blue-colored pixels. It can be observed that the points generated are not continuous in nature due to the fact that the steps were iterated only in situations where there was a transition in the slope of the contour to save computation time. Further, to ensure continuity of the generated contour points, they were connected using the straight line drawing algorithm by determining optimal pairs. Although the strategy proves effective, it is only suitable in situations where the inter-point distance is relatively small. In addition, as these lines are not able to retain the symmetry of the existing contours, their resemblance to the same may not be as desired. The strategy is also not suitable if the interpoint distance is very large.

Column C in table 6.8 (page no 298) represents the superimposition of the actual contour map with the interpolated points. Column D in table 6.8 (page no 298) represents the interpolated points as continuous contours obtained after reconnection. Eventually, column E in table 6.8 (page no 298) represents the refined elevation models.

Table 6.8: Sample Test Cases and results obtained

Sample	Input Sample (A)	DEM of (A)	Interpolated Points (B)	(A+B)	Contour Continuity	Refined DEM of (A+B)
1						
2						
3						
4						
5						
6						

6.7 Limitations of the research initiative

This research initiative transforms the 2D contours along with interpolated contours in 3D space. This transformation is limited to only contour features of topographic sheet. However, the transformation is not applicable to other geo-morphological features like rivers, etc. which may further extended to man-made features like roads, railways. Also, the color intensity assigned to different contours of varied altitude is constant. The variation in intensity of contour as per altitude provides better understanding and visibility of earth landscape. The other major limitation of research initiative is to validate the outcome, i.e., DEM. The best possible measure to assess the quality of outcome is with ground truth value which is extremely time-consuming. The other means assessment of DEM is through digitizer and the outcome totally relies on expertise of the digitizer.

6.8 Conclusion

The proposed research initiative is an effort directed towards enabling the generation of an elevation model that closely resembles the landscape. To attain the same, sixteen identifiable modules were conceived. A simple intensity-based segmentation technique was deployed to digitize the contours from the sample reference image. The digitized reference map was then refined using a median filter for the removal of salt and pepper noise. The contours in the digitized map were then skeletonized to create a single-pixel-width representation. This was done to reduce computation overhead. Common neighbors in the skeletonized dataset were resolved by identifying all m-connectivity. This tremendously helped in limiting the number of structural operators or morphological orientations to be managed. A collection of sample images was pre-processed using an unsupervised approach for identifying all possible unique 3x3 structural operators. A set of 24 structural operators was identified and appropriately stored in the repository. These structural operators were then indexed for future reference. Subsequently, the sample dataset was also processed to extract various attributes related to the existing contours about which the new set of contours were to be generated. In the case of lines, the centroid of the dataset and the lines were taken into reference for associating direction with the structural operators. Depending on the relative placement of the line centroid with regards to the centroid of the dataset, two distinct sets of angular movements were identified. Likewise, in the case of enclosures, the centroid of the enclosure and the coordinate position of the enclosures were taken into reference for associating direction with the structural operators. Depending on the relative placement of the enclosure's coordinates and its centroid, two distinct sets of angular movements were identified. A cost-effective traversal mechanism was pursued, reducing the computational load. To prevent imprecise interpolation due to lag in directional traversal, a task-suitable mask was designed. After identifying the appropriate endpoints of interpolation, the coordinates of the interpolated points were then generated. Structural operators resembling the source contour were then placed at the interpolated points to maintain structural or morphological resemblance. The structural operators were then connected using a gradient-based approach for generating intermediate contours. The contours were then elevated in the z-plane by associating elevation details.

Chapter 7

Summary and Conclusion

This research initiative was motivated to overcome the shortcomings of the existing manual and semi-automatic approaches, deployed for digitization of geomorphological features crucial to GIS based applications by leveraging the capabilities of advanced computational support and realization of fully automated knowledge based computational processes. Here, some of the notable morphological features considered are rivers, contours and associated annotations. The work is further advanced to generate purpose specific attributes to all these identifiable features. These attributes are crucial to the studies related to these features.

In pursuit of the research goals, the various objectives were drawn as specified below:

a) Conception and realization of fully automated computational process for extracting river pattern and generation of associated attributes.

b) Conception and realization of fully automated computational process for extraction of contours lines, its refinement and generation of associated attributes.

c) Conception and realization of fully automated computational process for text localization, recognition, and mapping.

d) Transformation of the features to 3D space for effective visualization

Upon successful attainment of the aforementioned research objectives with desired level of precision and computational accuracy, the outcome of the research initiative is summarized as follows:

7.1 Extraction of River Pattern, its Refinement and Attribute Generation

This module of the research work implements a fully automated knowledge-based computational process for extracting river network from TS, generating associated attributes, reducing human intervention, and consequently lowering effort, time, and cost requirements, thereby elevating the confidence of the research outcome.

Here, color segmentation was performed to digitize the river network from the TS. The digitized river network was suitably preprocessed in order to b skeletonized, eliminate inherent noise, and resolve m-connectivity. An effective spiral traversal mechanism was deployed for efficient identification of terminal streams. The refined river network was qualitatively and quantitatively analysed with regards to eight different river ordering concepts, namely, Classic Stream Order, Strahler Stream Order, Horton Stream Order, Shreve Stream Order, Scheidegger Stream Order, Order by Path Length, Consistent Stream Order, and Cumulative Stream Order. In due course of analysis, the following ten quantitative parameters, namely stream order, stream number, bifurcation ratio, streams participating in bifurcation ratio, weighted mean bifurcation ratio, stream length, mean stream length, streams participating in length ratio, weighted mean length ratio, and length of main channel were derived for each of the ordering techniques mentioned above. The proficiency of the implemented process was tested using selected river samples, and it was successfully able to accurately realize the motives of all the conceived aspects related to the modules.

A qualitative comparative assessment of all of these algorithms was performed, highlighting their various aspects, such as the technique used, the number of passes required, the computational cost, their advantages and disadvantages, and the need for human intervention. The obtained results can be further combined with aspects like the DEM and satellite imagery for simulating hydrological processes, studying morphological changes, and effective disaster management.

7.2 Extraction of Contour Lines, its Refinement and Attribute Generation

This module of the research work realizes a comprehensive, fully automated computational process directed towards digitizing contour lines, its effective preprocessing, and maintaining its continuity. The contour lines were digitized using color segmentation, and need-based filters were deployed for skeletonization, eliminating noise and resolving m-connectivity. A suitable modification to the Bezier curve technique was conceived to ensure connectivity between broken segments of a contour line, determining optimal pairs based on the sign of the gradient. The efficiency of the process was tested on 24 samples with 95 breakages, and an accuracy of 86.31% was achieved with an error rate of 13.68%. The process may be further generalized by making suitable adaptations to handle other linear morphological features like rivers and transportation networks.

7.3 Extraction of Associated Text, its Refinement and Attribute Generation

This module of the research work was motivated by the localization of annotations present in the TS and, thereafter, pursue knowledge-guided multilayer model for recognition of the localized text.

The aforementioned motives were successfully achieved through two distinct sub-modules dedicated to localization and recognition. Localization was performed leveraging the processing capabilities of color segmentation techniques, preprocessing operations, and morphological operators. The capability of the implemented model was tested by randomly selecting 30 samples of topographic sheets with 100 features of interest. The model successfully detected 85 features out of 100, delivering an accuracy of 85%.

In addition, a multi-layer model was conceived for extending the localization capabilities, relying on the strength of CNN-based deep learning architectures such as R-CNN, Fast-RCNN, and Faster-RCNN. The two models were tested with 15 different unknown samples of TS with 15-20 features of interest aligned along different directions. The trained RCNN model was able to localize with an accuracy of 51.44%, whereas the faster RCNN model delivered an accuracy of 72.42%. The accuracy is expected to increase with an increase in the training dataset. This implies that the faster R-CNN can identify both elevation values and text features on topographic sheets with greater efficiency. Further, the accuracy of the localization was improved to 79.6% (against Intersection over Union (IoU) value >= 0.7) using Faster RCNN+R-101-FPN.

For text recognition, an accuracy of 60% was achieved for the training set, and 45% accuracy was achieved for the validation set. The current recognizer model is limited to just english alphabet and can be further enhanced to identify numbers and special symbols. The accuracy is compromised due to the low quality of the images in the recognition dataset. The overall accuracy of the models listed above is greatly influenced by the relative positioning of the feature of interest with regard to other morphological features contained in the TS. The greater the disturbance to the feature of interest by other overlapping or intersecting features, the lower will be the accuracy, as it would greatly affect both testing as well as validation. The problem intensifies in situations where non-significant representations are wrongly pursued for training the models.

7.4 Generation of Digital Elevation Model from Refined and Generated Contour Lines

In this module, a sincere effort has been made to create a refined elevation model from existing and automatically interpolated contour lines. The same was achieved through the neatly conceived sixteen distinguishable sub-modules intended for segmentation, preprocessing, morphological analysis, directional analysis, interpolation, generation, and 3D modelling. Here, 14 different samples adequately representing possible contour orientations were considered as input to an unsupervised search operation directed at identifying all unique 3x3 orientations centered on a significant pixel. The process successfully identifies 24 such unique orientations (operators) contained in the sample image. To effectively contain the number of operators, the input samples were thinned to a single-pixel-width representation with all instances m-connectivity. With every identified operator, the direction of traversal was decided based on the angular orientation of the operators and designated reference points. Consequently, the operators were suitably located at the point of interpolation and were purposefully connected to create interpolated contour lines using a gradient-based method. The actual and interpolated contour lines were then projected in a 3D space to create the desired elevation model by incorporating elevation details. The overall accuracy of this module depends on the effectiveness of all the other modules discussed in section 7.1-7.3.

7.5 Limitation of Research Work

The following represents the limitations of the research initiative:

a) The robustness and accuracy of the conceived models may be further enhanced and elevated by incorporating complex scenarios with larger dimensions in the training set.

b) The efficiency and effectiveness of the conceived models greatly vary with the quality, tone, texture, and scale of TS. Advancement may be planned towards the conception of comprehensive generalizable scale invariant models capable of efficiently handling varying datasets.

c) The segmentation process inadequately handles poor-quality TS. It fails to produce the desired outcome in situations where various features intersect with each other, resulting in breakage. Furthermore, in situations where there is a close resemblance between the intensity values of different features, the segmentation process inappropriately inducts unnecessary values into the outcome.

d) In situations where breakages are observed in contour lines, the use of an inappropriate end point selection approach may result in the establishment of the wrong connections. In addition, if the breakage is unexpectedly large, the end point selection approach may fail to deliver the expected result. Further, contours generated using such an approach fail to preserve the morphological characteristics of the contours due to uncertainty of contour orientation, an inadequate number of control points, and inherent computational complexity.

e) The effectiveness of the models greatly depends on the refining ability of the preprocessing operations. These operations should be appropriately tied to the need to elevate the quality of representation.

f) All the models implemented mandate that the features of interest be represented with the help of a single pixel width for computational ease. Therefore, the skeletonization technique adopted should be able to preserve the morphological resemblance of the features of interest for effective processing.

g) All the models implemented mandated resolution of m-connectivity before the onset of feature engineering operations.

h) The overall ability of the model designed for interpolation of contour lines depends on its ability to identify the structural operators, its relative position based on a certain reference point, its ability to associate the direction of traversal with structural operators, and its ability to resolve hit-and-miss problems.

i) The overall effectiveness of all the implemented models greatly relies on the ability of the traversal technique used and the data structures designed to store the details of features.

j) Due to a lack of adequate annotated training data resembling text in the TS, the accuracy of localization achieved is considerably low, which adversely effects the recognition process. A research initiative may be planned towards automating the same for building up a comprehensive dataset.

k) Inappropriate recognition of the elevation value may distort the 3D model obtained after projection.

7.6 Future Directions

The following represents the future directions of the research initiative:

a) The model conceptualized for handing the river network may be carefully advanced to work with satellite imaginary.

b) Reconnection techniques preserving the morphological orientation of contours may be conceived for handling complex breakages.

c) An advanced multilayer model may be pursued to enhance the accuracy of localization and recognition by enhancing the training dataset.

d) The 3D projection may be used to visualize the DTM and DSM by overlapping satellite imaginary on top of them.

References

[1] India and Pakistan AMS Topographic Maps, [Online], “India and Pakistan AMS Topographic Maps,” [Online]. Available: https://legacy.lib.utexas.edu/maps/ams/india/. [Accessed 6 April 2021].

[2] “Intro to Vector Data in R,” [Online]. Available: https://www.neonscience.org/vector-data-series. [Accessed 6 April 2019].

[3] “Integrated Regional Water Management Program,” [Online]. Available: https://inyo-monowater.org/resources/giswhat/layers/. [Accessed 6 April 2019].

[4] “Drawn Lines Volcanoes,” [Online]. Available: https://art4clip.com/explore/Drawn%20lines%20volcano/. [Accessed 6 April 2019].

[5] “Contour Lines and Topo Maps,” [Online]. Available: https://www.greenbelly.co/pages/contour-lines. [Accessed 6 April 2019].

[6] C. M. Epstein, “APPLICATION OF ROSGEN ANALYSIS TO THE NEW JERSEY PINE BARRENS,” *Journal of the American Water Resources Association,* vol. 38, p. 69–78, February 2002.

[7] M. I. C. H. A. E. L. CHURCH, “Geomorphic thresholds in riverine landscapes,” *Freshwater Biology,* vol. 47, p. 541–557, April 2002.

[8] J. R. Desloges and M. A. Church, “WANDERING GRAVEL-BED RIVERS,” *The Canadian Geographer/Le Géographe canadien,* vol. 33, p. 360–364, December 1989.

[9] A. D. Knighton and G. C. Nanson, “Anastomosis and the continuum of channel pattern,” *Earth Surface Processes and Landforms,* vol. 18, p. 613–625, November 1993.

[10] G. M. Kondolf, H. Piégay, L. Schmitt and D. R. Montgomery, *Geomorphic classification of rivers and streams,* Wiley, 2016, p. 133–158.

[11] F. B. Lotspeich, “WATERSHEDS AS THE BASIC ECOSYSTEM: THIS CONCEPTUAL FRAMEWORK PROVIDES A BASIS FOR A NATURAL CLASSIFICATION SYSTEM,” *Journal of the American Water Resources Association,* vol. 16, p. 581–586, August 1980.

[12] D. R. Montgomery and E. Foufoula-Georgiou, “Channel network source representation using digital elevation models,” *Water Resources Research,* vol. 29, p. 3925–3934, December 1993.

[13] S. J. Paustian, “A channel type users guide for the Tongass National Forest, Southeast Alaska,” 2010.

[14] OSGeo (June, 2022) (Online) Available at:, " https://svn.osgeo.org/grass/grass-addons/grass7/raster/r.stream.order/ r.stream.order. html," [Accessed on: 19 February 2022].

[15] A. N. Strahler, "Quantitative analysis of watershed geomorphology," *Transactions, American Geophysical Union,* vol. 38, p. 913, 1957.

[16] R. Chorley, "Horton, RE 1945: Erosional development of streams and their drainage basins: hydrophysical approach to quantitative morphology. Bulletin of the Geological Society of America 56, 2 75-3 70," *Progress in Physical Geography,* vol. 19, p. 533–554, 1995.

[17] R. L. Shreve, "Infinite Topologically Random Channel Networks," *The Journal of Geology,* vol. 75, p. 178–186, March 1967.

[18] R. L. Shreve, "Statistical Law of Stream Numbers," *The Journal of Geology,* vol. 74, p. 17–37, January 1966.

[19] A. E. Scheidegger, "The algebra of stream-order numbers," *United States Geological Survey Professional Paper,* vol. 525, p. 187–189, 1965.

[20] A. E. Scheidegger, "Horton's law of stream order numbers and a temperature-analog in river nets," *Water Resources Research,* vol. 4, p. 167–171, February 1968.

[21] O. Buttner, J. W. Jawitz and D. Borchardt, "Ecological status of river networks: stream order-dependent impacts of agricultural and urban pressures across ecoregions," *Environmental Research Letters,* vol. 15, p. 1040b3, October 2020.

[22] Geographic Information Technology Training Alliance, (April, 2016), (Online) Available at:, "http://www.gitta.info/Accessibiliti/en/html/StructPropNetw_learningObject5.html," [Accessed on: 19 February 2022].

[23] K. W. Cummins, "Structure and Function of Stream Ecosystems," *BioScience,* vol. 24, p. 631–641, November 1974.

[24] F. M. Henderson, "Stability of Alluvial Channels," *Transactions of the American Society of Civil Engineers,* vol. 128, p. 657–686, January 1963.

[25] K. Pareta and U. Pareta, "Quantitative Morphometric Analysis of a Watershed of Yamuna Basin, India using ASTER (DEM) Data and GIS," *International journal of Geomatics and Geosciences,* vol. 2, pp. 248-269, 2011.

[26] T. Bryndal, "The River Systems in Small Catchments in the Context of the Horton's and Schumm's Laws – Implication for Hydrological Modelling. The Case Study of the Polish Carpathians," *Quaestiones Geographicae,* vol. 34, p. 85–98, March 2015.

[27] M. J. Winterbourn, “The river continuum concept — reply to Barmuta and Lake,” *New Zealand Journal of Marine and Freshwater Research,* vol. 16, p. 229–231, June 1982.

[28] C. E. Cushing, C. D. McIntire, K. W. Cummins, G. W. Minshall, R. C. Petersen and J. R. Sedell, “Relationships among chemical, physical, and biological indices along river continua based on multivariate analyses,” *Archiv für Hydrobiologie,* vol. 98, p. 317–326, 1983.

[29] D. R. Montgomery, “PROCESS DOMAINS AND THE RWER CONTINTUUM,” *Journal of the American Water Resources Association,* vol. 35, p. 397–410, April 1999.

[30] R. L. Vannote, G. W. Minshall, K. W. Cummins, J. R. Sedell and C. E. Cushing, “The River Continuum Concept,” *Canadian Journal of Fisheries and Aquatic Sciences,* vol. 37, p. 130–137, January 1980.

[31] W. N. McDavitt, Application of a stream channel classification in a mountainous setting, Utah State University, 2004.

[32] T. Y. Barila, R. D. Williams and J. R. Stauffer, “The Influence of Stream Order and Selected Stream Bed Parameters on Fish Diversity in Raystown Branch, Susquehanna River Drainage, Pennsylvania,” *The Journal of Applied Ecology,* vol. 18, p. 125, April 1981.

[33] E. E. Agency., European waters: assessment of status and pressures 2018., Publications Office, 2018.

[34] W. F. Directive, “Common implementation strategy for the water framework directive (2000/60/EC),” *Guidance document,* vol. 7, 2003.

[35] A. Lorenz, C. K. Feld and D. Hering, “Typology of streams in Germany based on benthic invertebrates: Ecoregions, zonation, geology and substrate,” *Limnologica,* vol. 34, p. 379–389, December 2004.

[36] S. Birk, W. Bonne, A. Borja, S. Brucet, A. Courrat, S. Poikane, A. Solimini, W. van de Bund, N. Zampoukas and D. Hering, “Three hundred ways to assess Europe\textquotesingles surface waters: An almost complete overview of biological methods to implement the Water Framework Directive,” *Ecological Indicators,* vol. 18, p. 31–41, July 2012.

[37] M. Kuemmerlen, P. Reichert, R. Siber and N. Schuwirth, “Ecological assessment of river networks: From reach to catchment scale,” *Science of The Total Environment,* vol. 650, p. 1613–1627, February 2019.

[38] G. M. A. T. H. I. A. S. KONDOLF, M. W. SMELTZER and S. F. RAILSBACK, “Design and Performance of a Channel Reconstruction Project in a Coastal California Gravel-Bed Stream,” *Environmental Management,* vol. 28, p. 761–776, December 2001.

[39] K. L. E. M. E. N. T. TOCKNER, M. A. R. T. I. N. PUSCH, D. I. E. T. R. I. C. H. BORCHARDT and M. S. LORANG, “Multiple stressors in coupled river-floodplain ecosystems,” *Freshwater Biology,* vol. 55, p. 135–151, January 2010.

[40] D. Hering, L. Carvalho, C. Argillier, M. Beklioglu, A. Borja, A. C. Cardoso, H. Duel, T. Ferreira, L. Globevnik, J. Hanganu, S. Hellsten, E. Jeppesen, V. Kodeš, A. L. Solheim, T. Nõges, S. Ormerod, Y. Panagopoulos, S. Schmutz, M. Venohr and S. Birk, “Managing aquatic ecosystems and water resources under multiple stress — An introduction to the MARS project,” *Science of The Total Environment,* Vols. 503-504, p. 10–21, January 2015.

[41] R. J. Naiman, “Biotic stream classification,” *River ecology and management: Lessons from the Pacific coastal ecoregion,* p. 97–119, 1998.

[42] J. A. Priess, J. Hauck, R. Haines-Young, R. Alkemade, M. Mandryk, C. J. Veerkamp, G. Bela, P. Berry, R. Dunford, P. Harrison, H. Keune, M. Kok, L. Kopperoinen, T. Lazarova, J. Maes, G. Pataki, E. Preda, C. Schleyer, A. Vadineanu and G. Zulian, “New EU-Level Scenarios on the Future of Ecosystem Services,” in *Atlas of Ecosystem Services*, Springer International Publishing, 2019, p. 135–140.

[43] Y. Fang, S. Ceola, K. Paik, G. McGrath, P. S. C. Rao, A. Montanari and J. W. Jawitz, “Globally Universal Fractal Pattern of Human Settlements in River Networks,” *Earth\textquotesingles Future,* vol. 6, p. 1134–1145, August 2018.

[44] S. Yang, O. Büttner, J. W. Jawitz, R. Kumar, P. S. C. Rao and D. Borchardt, “Spatial Organization of Human Population and Wastewater Treatment Plants in Urbanized River Basins,” *Water Resources Research,* vol. 55, p. 6138–6152, July 2019.

[45] C. G. Jager and D. Borchardt, “Longitudinal patterns and response lengths of algae in riverine ecosystems: A model analysis emphasising benthic-pelagic interactions,” *Journal of Theoretical Biology,* vol. 442, p. 66–78, April 2018.

[46] R. Dupas, B. W. Abbott, C. Minaudo and O. Fovet, “Distribution of Landscape Units Within Catchments Influences Nutrient Export Dynamics,” *Frontiers in Environmental Science,* vol. 7, April 2019.

[47] J. S. Bridge, “The interaction between channel geometry, water flow, sediment transport and deposition in braided rivers,” *Geological Society, London, Special Publications,* vol. 75, p. 13–71, January 1993.

[48] J. M. Buffington, R. D. Woodsmith, D. B. Booth and D. R. Montgomery, “Fluvial processes in Puget Sound rivers and the Pacific Northwest,” *Restoration of Puget Sound Rivers,* p. 46–78, 2003.

[49] W. B. Dade, “Grain size, sediment transport and alluvial channel pattern,” *Geomorphology,* vol. 35, p. 119–126, October 2000.

[50] G. C. Poole, J. A. Stanford, S. W. Running, C. A. Frissell, W. W. Woessner and B. K. Ellis, “A patch hierarchy approach to modeling surface and subsurface hydrology in

complex flood-plain environments," *Earth Surface Processes and Landforms,* vol. 29, p. 1259–1274, September 2004.

[51] J. W. Powell, Exploration of the Colorado River of the West and its Tributaries, 1875.

[52] M. H. Wimmer, M. Hollaus, G. Blöschl, A. Buttinger-Kreuzhuber, J. Komma, J. Waser and N. Pfeifer, "Processing of nationwide topographic data for ensuring consistent river network representation," *Journal of Hydrology X,* vol. 13, p. 100106, December 2021.

[53] M. Muthusamy, M. R. Casado, D. Butler and P. Leinster, "Understanding the effects of Digital Elevation Model resolution in urban fluvial flood modelling," *Journal of Hydrology,* vol. 596, p. 126088, May 2021.

[54] Q.-Y. Ma, A.-B. Li and P. Wang, "Automatic detection of river capture based on planform pattern and \upchi{}{-}{p}lot of the stream network," *Geomorphology,* vol. 425, p. 108587, March 2023.

[55] P.-C. Huang and K. T. Lee, "Influence of topographic features and stream network structure on the spatial distribution of hydrological response," *Journal of Hydrology,* vol. 603, p. 126856, December 2021.

[56] N. Gasnier, L. Denis, R. Fjortoft, F. Liege and F. Tupin, "Narrow River Extraction From SAR Images Using Exogenous Information," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing,* vol. 14, p. 5720–5734, 2021.

[57] E. W. Lane, "Study of the shape of channels formed by natural streams flowing in erodible material, A," 1957.

[58] R. D. Harmel, C. T. Haan and R. C. Dutnell, "EVALUATION OF ROSGEN'S STREAMBANK EROSION POTENTIAL ASSESSMENT IN NORTHEAST OKLAHOMA," *Journal of the American Water Resources Association,* vol. 35, p. 113–121, February 1999.

[59] R. Kellerhals, M. Church and D. I. Bray, "Classification and Analysis of River Processes," *Journal of the Hydraulics Division,* vol. 102, p. 813–829, July 1976.

[60] T. J. Beechie, M. Liermann, M. M. Pollock, S. Baker and J. Davies, "Channel pattern and river-floodplain dynamics in forested mountain river systems," *Geomorphology,* vol. 78, p. 124–141, August 2006.

[61] G. J. Brierley and K. A. Fryirs, Eds., Geomorphology and River Management, Wiley, 2004.

[62] D. L. Rosgen, "A classification of natural rivers," *CATENA,* vol. 22, p. 169–199, June 1994.

[63] C. N. Goodwin, "Fluvial classification: Neanderthal necessity or needless normalcy," *Wildland hydrology,* vol. 229236, 1999.

[64] Z. X. Dai, C. M. Li, P. D. Wu and Y. Yin, “AN AUTOMATED BOTTOM UP HYDROLOGIC CODING SYSTEM FOR DENDRITIC RIVER SYSTEM,” *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences,* Vols. XLII-4/W16, p. 169–175, October 2019.

[65] W. Rieger, “Automated river line and catchment area extraction from dem data,” *International Archives of Photogrammetry and Remote Sensing,* vol. 29, p. 642–642, 1993.

[66] Y. Xue, C. Qin, B. Wu, D. Li and X. Fu, “Automatic Extraction of Mountain River Surface and Width Based on Multisource High-Resolution Satellite Images,” *Remote Sensing,* vol. 14, p. 2370, May 2022.

[67] C. Ciaburri, M. Kiehnle-Benitez, A. Sheta and M. Braik, “Automatic extraction of rivers from satellite images using image processing techniques,” *ACCENTS Transactions on Image Processing and Computer Vision,* vol. 6, p. 32–41, May 2020.

[68] F. H. Dawson, D. D. Hornby and J. Hilton, “A method for the automated extraction of environmental variables to help the classification of rivers in Britain,” *Aquatic Conservation: Marine and Freshwater Ecosystems,* vol. 12, p. 391–403, 2002.

[69] M. P. Pradhan, M. K. Ghose and Y. R. Kharka, “Automatic association of Strahler’s order and attributes with the drainage system,” *International Journal of Advanced Computer Science and Applications,* vol. 3, 2012.

[70] P. Venkatachalam, B. K. Mohan, A. Kotwal, V. Mishra, V. Muthuramakrishnan and M. Pandya, “Automatic delineation of watersheds for hydrological applications,” in *22nd Asian conference on remote sensing*, 2001.

[71] D. G. Tarboton, “Terrain analysis using digital elevation models in hydrology,” in *23rd ESRI international users conference, San Diego, California*, 2003.

[72] R. Storey and S. Wadhwa, An assessment of the lengths of permanent, intermittent and ephemeral streams in the Auckland region, Auckland Regional Council, 2009.

[73] A. Sen and T. Gokgoz, “Clustering Approaches for Hydrographic Generalization,” in *Proceedings of GIS Ostrava 2012—Surface Models for Geoscience, Ostrava*, 2012.

[74] C. Li, X. Liu, W. Wu and Z. Hao, “A Reconstruction Method for Broken Contour Lines Based on Similar Contours,” *ISPRS International Journal of Geo-Information,* vol. 8, p. 8, December 2018.

[75] S. Gul and M. F. Khan, “Automatic Extraction of Contour Lines from Topographic Maps,” in *2010 International Conference on Digital Image Computing: Techniques and Applications*, 2010.

[76] T. Y. Zhang and C. Y. Suen, “A fast parallel algorithm for thinning digital patterns,” *Communications of the ACM,* vol. 27, p. 236–239, March 1984.

[77] N. Dmitriev, “Complex method of reconstruction of contour lines,” in *PROCEEDINGS OF THE 45TH INTERNATIONAL CONFERENCE ON APPLICATION OF MATHEMATICS IN ENGINEERING AND ECONOMICS (AMEE'19)*, 2019.

[78] E. Hancer, R. Samet and D. Karaboga, “A hybrid method to the reconstruction of contour lines from scanned topographic maps,” in *2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE)*, 2014.

[79] F. Wang, P. Liu, Y. Yang, H. Wei and X. An, “A Novel Method for Reconstructing Broken Contour Lines Extracted from Scanned Topographic Maps,” *Proceedings of the ICA,* vol. 1, p. 1–7, May 2018.

[80] M. P. a. G. M. a. R. P. S. a. M. N. Pradhan, “Knowledge based contour line reconnection techniques,” *International Journal of Computer Applications,* vol. 65, no. 9, 2013.

[81] B. Xu, J. Chen and M. Yao, “Identification of Contour Lines from Average-Quality Scanned Topographic Maps,” *Mathematical Problems in Engineering,* vol. 2016, p. 1–14, 2016.

[82] A. Khotanzad and E. Zink, “Contour line and geographic feature extraction from USGS color topographical paper maps,” *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 25, p. 18–31, January 2003.

[83] L. M. San, S. M. Yatim, N. A. M. Sheriff and N. I. bin Nik Ismail, “Extracting contour lines from scanned topographic maps,” in *Proceedings. International Conference on Computer Graphics, Imaging and Visualization, 2004. CGIV 2004.*.

[84] D.-j. Xin, X. Zhou and H. Zheng, “Contour Line Extraction from Paper-based Topographic Maps,” 2006.

[85] J. R. Parker, Algorithms for image processing and computer vision, Wiley Computer Pub., 2011, p. 480.

[86] S. Oka, A. Garg and K. Varghese, “Vectorization of contour lines from scanned topographic maps,” *Automation in Construction,* vol. 22, p. 192–202, March 2012.

[87] K. I. Joy, “Breshenham’s algorithm,” *Visualization and Graphics Research Group, Department of Computer Science, University of California, Davis,* p. 1–15, 1999.

[88] D. Amirkhani and A. Bastanfard, “An objective method to evaluate exemplar-based inpainted images quality using Jaccard index,” *Multimedia Tools and Applications,* April 2021.

[89] H. Mansourifar and A. Bastanfard, “Fast Conic Spline Data Fitting of Noise-Free Data Points,” in *2011 Eighth International Conference Computer Graphics, Imaging and Visualization*, 2011.

[90] H. Mansourifar, M. M. Dehshibi and A. Bastanfard, "Shoulder Point Detection: A Fast Geometric Data Fitting Algorithm," in *2011 International Conference on Cyberworlds*, 2011.

[91] H. Mansourifar and A. Bastanfard, "A Novel Practical Approach for Weight Manipulation of Conic Splines," in *2011 Eighth International Conference Computer Graphics, Imaging and Visualization*, 2011.

[92] H. Li, J. Liu and X. Zhou, "Intelligent Map Reader: A Framework for Topographic Map Understanding With Deep Learning and Gazetteer," *IEEE Access,* vol. 6, p. 25363–25376, 2018.

[93] K. Pokonieczny and S. Borkowska, "Using artificial neural network for labelling polygon features in topographic maps," *GeoScape,* vol. 13, p. 125–131, December 2019.

[94] A. Pezeshk and R. L. Tutwiler, "Automatic Feature Extraction and Text Recognition From Scanned Topographic Maps," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 49, p. 5047–5063, December 2011.

[95] N. G. Ganpatro and J. K. Ghosh, "Information extraction from topographic map using colour and shape analysis," *Sadhana,* vol. 39, p. 1095–1117, August 2014.

[96] S. Kang, J. Lee, K. Bong, C. Kim, Y. Kim and H.-J. Yoo, "Low-Power Scalable 3-D Face Frontalization Processor for CNN-Based Face Recognition in Mobile Devices," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems,* vol. 8, p. 873–883, December 2018.

[97] X.-X. Niu and C. Y. Suen, "A novel hybrid CNN–SVM classifier for recognizing handwritten digits," *Pattern Recognition,* vol. 45, p. 1318–1325, April 2012.

[98] Z. Cao, T. Simon, S.-E. Wei and Y. Sheikh, "Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields," November 2016.

[99] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning," *Nature,* vol. 521, p. 436–444, May 2015.

[100] Z.-Q. Zhao, P. Zheng, S.-T. Xu and X. Wu, "Object Detection With Deep Learning: A Review," *IEEE Transactions on Neural Networks and Learning Systems,* vol. 30, p. 3212–3232, November 2019.

[101] J. Yang and J. Li, "Application of deep convolution neural network," in *2017 14th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, 2017.

[102] J. Yim, J. Ju, H. Jung and J. Kim, "Image Classification Using Convolutional Neural Networks With Multi-stage Feature," in *Advances in Intelligent Systems and Computing*, Springer International Publishing, 2015, p. 587–594.

[103] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn and A. Zisserman, “The Pascal Visual Object Classes (VOC) Challenge,” *International Journal of Computer Vision,* vol. 88, p. 303–338, September 2009.

[104] G. E. Hinton, S. Osindero and Y.-W. Teh, “A Fast Learning Algorithm for Deep Belief Nets,” *Neural Computation,* vol. 18, p. 1527–1554, July 2006.

[105] O. K. Bretschneider T, “Content-based image retrieval,” *Encyclopedia of Data Ware Housing Mining,* vol. 4, no. 34, 2010.

[106] R. Chauhan, K. K. Ghanshala and R. C. Joshi, “Convolutional Neural Network (CNN) for Image Detection and Recognition,” in *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)*, 2018.

[107] C.-L. Liu, K. Nakashima, H. Sako and H. Fujisawa, “Handwritten digit recognition: benchmarking of state-of-the-art techniques,” *Pattern Recognition,* vol. 36, p. 2271–2285, October 2003.

[108] K. E. A. van de Sande, T. Gevers and C. G. M. Snoek, “Evaluating Color Descriptors for Object and Scene Recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 32, p. 1582–1596, September 2010.

[109] J. Imran and P. Kumar, “Human action recognition using RGB-D sensor and deep convolutional neural networks,” in *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2016.

[110] J. Redmon and A. Angelova, “Real-Time Grasp Detection Using Convolutional Neural Networks,” December 2014.

[111] N. Jmour, S. Zayen and A. Abdelkrim, “Convolutional neural networks for image classification,” in *2018 International Conference on Advanced Systems and Electric Technologies (IC_ASET)*, 2018.

[112] J. Du, “Understanding of Object Detection Based on CNN Family and YOLO,” *Journal of Physics: Conference Series,* vol. 1004, p. 012029, April 2018.

[113] H. Gao, B. Cheng, J. Wang, K. Li, J. Zhao and D. Li, “Object Classification Using CNN-Based Fusion of Vision and LIDAR in Autonomous Vehicle Environment,” *IEEE Transactions on Industrial Informatics,* vol. 14, p. 4224–4231, September 2018.

[114] M. Cheon, W. Lee, C. Yoon and M. Park, “Vision-Based Vehicle Detection System With Consideration of the Detecting Location,” *IEEE Transactions on Intelligent Transportation Systems,* vol. 13, p. 1243–1252, September 2012.

[115] H. Rashid, N. Zafar, M. J. Iqbal, H. Dawood and H. Dawood, “Single Image Dehazing using CNN,” *Procedia Computer Science,* vol. 147, p. 124–130, 2019.

[116] E. Hsu, T. Mertens, S. Paris, S. Avidan and F. Durand, “Light mixture estimation for spatially varying white balance,” *ACM Transactions on Graphics,* vol. 27, p. 1–7, August 2008.

[117] P. Wang, X. Zhang and Y. Hao, “A Method Combining CNN and ELM for Feature Extraction and Classification of SAR Image,” *Journal of Sensors,* vol. 2019, p. 1–8, November 2019.

[118] R. Girshick, J. Donahue, T. Darrell and J. Malik, “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation,” in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[119] S. Ren, K. He, R. Girshick, X. Zhang and J. Sun, “Object Detection Networks on Convolutional Feature Maps,” April 2015.

[120] Y. Ren, C. Zhu and S. Xiao, “Object Detection Based on Fast/Faster RCNN Employing Fully Convolutional Architectures,” *Mathematical Problems in Engineering,* vol. 2018, p. 1–7, 2018.

[121] S. Ren, K. He, R. Girshick and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 39, p. 1137–1149, June 2017.

[122] K. He, X. Zhang, S. Ren and J. Sun, “Deep Residual Learning for Image Recognition,” December 2015.

[123] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, “Going Deeper with Convolutions,” September 2014.

[124] K.-H. Kim, S. Hong, B. Roh, Y. Cheon and M. Park, “PVANET: Deep but Lightweight Neural Networks for Real-time Object Detection,” August 2016.

[125] A. Krizhevsky, I. Sutskever and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Communications of the ACM,* vol. 60, p. 84–90, May 2017.

[126] R. Girshick, “Fast R-CNN,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

[127] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” September 2014.

[128] C. Cao, B. Wang, W. Zhang, X. Zeng, X. Yan, Z. Feng, Y. Liu and Z. Wu, “An Improved Faster R-CNN for Small Object Detection,” *IEEE Access,* vol. 7, p. 106838–106846, 2019.

[129] D. Alamsyah and M. Fachrurrozi, “Faster R-CNN with Inception V2 for Fingertip Detection in Homogenous Background Image,” *Journal of Physics: Conference Series,* vol. 1196, p. 012017, March 2019.

[130] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[131] M. J. Alam and M. Chowdhury, “Detection of fingertips based on the combination of color information and circle detection,” in *2013 IEEE 8th International Conference on Industrial and Information Systems*, 2013.

[132] D. Alamsyah and M. I. Fanany, “Particle filter for 3D fingertips tracking from color and depth images with occlusion handling,” in *2013 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, 2013.

[133] M. Rachmadi and D. Alamsyah, “Estimasi Citra Kedalaman Dengan Conditional Random Field (CRF) dan Structured Support Vector Machine (SSVM),” *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi),* vol. 1, p. 198–203, December 2017.

[134] Y. Kawazoe, K. Shimamoto, R. Yamaguchi, Y. Shintani-Domoto, H. Uozaki, M. Fukayama and K. Ohe, “Faster R-CNN-Based Glomerular Detection in Multistained Human Whole Slide Images,” *Journal of Imaging,* vol. 4, p. 91, July 2018.

[135] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[136] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu and A. C. Berg, “SSD: Single Shot MultiBox Detector,” in *Computer Vision – ECCV 2016*, Springer International Publishing, 2016, p. 21–37.

[137] H. Nguyen, “Improving Faster R-CNN Framework for Fast Vehicle Detection,” *Mathematical Problems in Engineering,* vol. 2019, p. 1–11, November 2019.

[138] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto and H. Adam, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” April 2017.

[139] N. Bodla, B. Singh, R. Chellappa and L. S. Davis, “Soft-NMS — Improving Object Detection with One Line of Code,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017.

[140] X. Hu, X. Xu, Y. Xiao, H. Chen, S. He, J. Qin and P.-A. Heng, “SINet: A Scale-Insensitive Convolutional Neural Network for Fast Vehicle Detection,” *IEEE Transactions on Intelligent Transportation Systems,* vol. 20, p. 1010–1019, March 2019.

[141] S. Shivajirao, R. Hantach, S. B. Abbes and P. Calvez, “Mask R-CNN End-to-End Text Detection and Recognition,” in *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, 2019.

[142] N. Gandhewar, S. R. Tandan and R. Miri, “Deep Learning based Framework for Text Detection,” in *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, 2021.

[143] A. N. J. Raj, C. Junmin, R. Nersisson, V. G. V. Mahesh and Z. Zhuang, “Bilingual text detection from natural scene images using faster R-CNN and extended histogram of oriented gradients,” *Pattern Analysis and Applications,* vol. 25, p. 1001–1013, April 2022.

[144] L. Yang, D. Ergu, Y. Cai, F. Liu and B. Ma, “A review of natural scene text detection methods,” *Procedia Computer Science,* vol. 199, p. 1458–1465, 2022.

[145] X.-Y. Ding, X.-Q. Liu, X. Luo and X.-S. Xu, “DOC: Text Recognition via Dual Adaptation and Clustering,” *IEEE Transactions on Multimedia,* p. 1–11, 2023.

[146] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus and Y. LeCun, “Overfeat: Integrated recognition, localization and detection using convolutional networks,” *arXiv preprint arXiv:1312.6229,* 2013.

[147] R. Girshick, J. Donahue, T. Darrell and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014.

[148] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers and A. W. M. Smeulders, “Selective Search for Object Recognition,” *International Journal of Computer Vision,* vol. 104, p. 154–171, April 2013.

[149] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[150] J. Redmon and A. Farhadi, “YOLO9000: better, faster, stronger,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[151] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” *arXiv preprint arXiv:1804.02767,* 2018.

[152] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama and others, “Speed/accuracy trade-offs for modern convolutional object detectors,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[153] C. Szegedy, S. Ioffe, V. Vanhoucke and A. Alemi, “Inception-v4, inception-resnet and the impact of residual connections on learning,” in *Proceedings of the AAAI conference on artificial intelligence*, 2017.

[154] T. Wang, D. J. Wu, A. Coates and A. Y. Ng, “End-to-end text recognition with convolutional neural networks,” in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, 2012.

[155] A. Bissacco, M. Cummins, Y. Netzer and H. Neven, "PhotoOCR: Reading Text in Uncontrolled Conditions," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013.

[156] M. Jaderberg, K. Simonyan, A. Vedaldi and A. Zisserman, "Reading Text in the Wild with Convolutional Neural Networks," *International Journal of Computer Vision,* vol. 116, p. 1–20, May 2015.

[157] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke and J. Schmidhuber, "A Novel Connectionist System for Unconstrained Handwriting Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 31, p. 855–868, May 2009.

[158] B. Su and S. Lu, "Accurate Scene Text Recognition Based on Recurrent Neural Network," in *Computer Vision – ACCV 2014*, Springer International Publishing, 2015, p. 35–48.

[159] B. Shi, X. Bai and C. Yao, "An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 39, p. 2298–2304, November 2017.

[160] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation,* vol. 9, p. 1735–1780, November 1997.

[161] F. A. Gers, N. N. Schraudolph and J. Schmidhuber, "Learning precise timing with LSTM recurrent networks," *Journal of machine learning research,* vol. 3, p. 115–143, 2002.

[162] A. Graves, A.-r. Mohamed and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013.

[163] Y. Bengio, P. Simard and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks,* vol. 5, p. 157–166, March 1994.

[164] R. Girshick, "Fast R-CNN," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015.

[165] J. Mesa-Mingorance, E. Chicaiza, X. Buenaño, J. Cai, A. Rodríguez-Pascual and F. Ariza-López, "Analysis of Users and Uses of DEMs in Spain," *ISPRS International Journal of Geo-Information,* vol. 6, p. 406, December 2017.

[166] P. Stroeven, K. Li, N. L. B. Le, H. He and M. Stroeven, "Capabilities for property assessment on different levels of the micro-structure of DEM-simulated cementitious materials," *Construction and Building Materials,* vol. 88, p. 105–117, July 2015.

[167] S. Saksena and V. Merwade, "Incorporating the effect of DEM resolution and accuracy for improved flood inundation mapping," *Journal of Hydrology,* vol. 530, p. 180–194, November 2015.

[168] C. Gomez, Y. Hayakawa and H. Obanawa, "A study of Japanese landscapes using structure from motion derived DSMs and DEMs based on historical aerial photographs: New opportunities for vegetation monitoring and diachronic geomorphology," *Geomorphology,* vol. 242, p. 11–20, August 2015.

[169] I. S. Kweon and T. Kanade, "Extracting Topographic Terrain Features from Elevation Maps," *CVGIP: Image Understanding,* vol. 59, p. 171–182, March 1994.

[170] K. Kraus and N. Pfeifer, "Determination of terrain models in wooded areas with airborne laser scanner data," *ISPRS Journal of Photogrammetry and Remote Sensing,* vol. 53, p. 193–203, August 1998.

[171] C. J. V. Westen, "3.10 Remote Sensing and GIS for Natural Hazards Assessment and Disaster Risk Management," in *Treatise on Geomorphology*, Elsevier, 2013, p. 259–298.

[172] L. Kumar, A. K. Skidmore and E. Knowles, "Modelling topographic variation in solar radiation in a GIS environment," *International Journal of Geographical Information Science,* vol. 11, p. 475–497, July 1997.

[173] Q. Zhou and Y. Chen, "Generalization of DEM for terrain analysis using a compound method," *ISPRS Journal of Photogrammetry and Remote Sensing,* vol. 66, p. 38–45, January 2011.

[174] M. V. Alba-Fernández, F. J. Ariza-López and M. D. Jiménez-Gamero, "A New Approach to the Quality Control of Slope and Aspect Classes Derived from Digital Elevation Models," *Remote Sensing,* vol. 13, p. 2069, May 2021.

[175] S. Wise, "Assessing the quality for hydrological applications of digital elevation models derived from contours," *Hydrological processes,* vol. 14, p. 1909–1929, 2000.

[176] P. F. Fisher and N. J. Tate, "Causes and consequences of error in digital elevation models," *Progress in Physical Geography: Earth and Environment,* vol. 30, p. 467–489, August 2006.

[177] S. Frischknecht, E. Kanani and A. Carosio, "A raster-based approach for the automatic interpretation of topographic maps," *International Archives of Photogrammetry and Remote Sensing,* vol. 32, p. 523–530, 1998.

[178] A. Khotanzad and E. Zink, "Color paper map segmentation using eigenvector line-fitting," in *Proceeding of Southwest Symposium on Image Analysis and Interpretation*.

[179] H. Yan, "Color map image segmentation using optimized nearest neighbor classifiers," in *Proceedings of 2nd International Conference on Document Analysis and Recognition (ICDAR \textquotesingle93)*.

[180] G. Aumann, H. Ebner and L. Tang, "Automatic derivation of skeleton lines from digitized contours," *ISPRS Journal of Photogrammetry and Remote Sensing,* vol. 46, p. 259–268, October 1991.

[181] Z. Xian and L. Xiaobing, "Improved DDA line drawing anti-aliasing algorithm based on embedded graphics system," in *2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE)*, 2010.

[182] P. G. McCrea and P. W. Baker, "On Digital Differential Analyzer (DDA) Circle Generation for Computer Graphics," *IEEE Transactions on Computers,* Vols. C-24, p. 1109–1110, November 1975.

[183] M. U. Khan, M. R. Beg and M. Z. Khan, "Improved Line Drawing Algorithm: An Approach and Proposal," in *Proceedings of the International Conference on Advances in Computer Science and Electronics Engineering*, 2012.

[184] D. Wang, X. Chen, H. Toh and F. Yang, "A Simple Anti-Aliasing Method for Straight Line Drawing based on DSP Platform," in *2006 1ST IEEE Conference on Industrial Electronics and Applications*, 2006.

[185] X. Wu, "An efficient antialiasing technique," *ACM SIGGRAPH Computer Graphics,* vol. 25, p. 143–152, July 1991.

[186] J. Wassenberg, "Fast, High-Quality Line Antialiasing by Prefiltering with an Optimal Cubic Polynomial," in *2010 Fourth Pacific-Rim Symposium on Image and Video Technology*, 2010.

[187] K. K. Gorowara, "A problem on Bezier curves and Bezier surfaces," in *IEEE Conference on Aerospace and Electronics*.

[188] L. Yurong and D. Fuguo, "A Novel Pixel Line Based Algorithm for Line Generation," *International Journal of Digital Content Technology and its Applications,* vol. 4, p. 197–202, October 2010.

[189] P. Axelsson, "Interactive 3D extension of 2D map data using mono images," *International Archives of Photogrammetry and Remote Sensing,* vol. 32, p. 145–150, 1997.

[190] F. J. Ariza-Lopez and E. G. Chicaiza Mora, "DEMs: An Approach to Users and Uses from the Quality Perspective," *International Jounral of Spatial Data Infrastructure Research,* p. 131–171, May 2018.

[191] A. Casado, B. Hortobágyi and E. Roussel, "Historic reconstruction of reservoir topography using contour line interpolation and structure from motion

photogrammetry," *International Journal of Geographical Information Science,* vol. 32, p. 2427–2446, September 2018.

[192] M. A. Fonstad, J. T. Dietrich, B. C. Courville, J. L. Jensen and P. E. Carbonneau, "Topographic structure from motion: a new development in photogrammetric measurement," *Earth Surface Processes and Landforms,* vol. 38, p. 421–430, January 2013.

[193] M. R. James and S. Robson, "Straightforward reconstruction of 3D surfaces and topography with a camera: Accuracy and geoscience application," *Journal of Geophysical Research: Earth Surface,* vol. 117, p. n/a–n/a, August 2012.

[194] M. J. Westoby, J. Brasington, N. F. Glasser, M. J. Hambrey and J. M. Reynolds, "`Structure-from-Motion' photogrammetry: A low-cost, effective tool for geoscience applications," *Geomorphology,* vol. 179, p. 300–314, December 2012.

[195] A. Goudie, Geomorphological techniques, Routledge, 1994, p. 570.

[196] J. L. Carrivick, M. W. Smith and D. J. Quincey, Structure from Motion in the Geosciences, Wiley, 2016.

[197] I. K. Grain, "Computer interpolation and contouring of two-dimensional data: A review," *Geoexploration,* vol. 8, p. 71–86, August 1970.

[198] M.-L. Duplaquet and E. Cubero-Castan, "Updating cartographic models by Spot images interpretation," in *SPIE Proceedings*, 1994.

[199] W. Newton, C. Gurney, D. R. Sloggett and I. J. Dowman, "Approach to the automated identification of forests and forest change in remotely sensed images," in *SPIE Proceedings*, 1994.

[200] B. Plietker, "Semiautomatic revision of street objects in ATKIS database DLM 25/1," *International Archives of Photogrammetry and Remote Sensing,* vol. 30, p. 311–317, 1994.

[201] H. Yamada, K. Yamamoto and K. Hosokawa, "Directional mathematical morphology and reformalized Hough transformation for the analysis of topographic maps," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 15, p. 380–387, April 1993.

[202] K. J. Goodson and P. H. Lewis, "A knowledge based line recognition system," *Pattern Recognition Letters,* vol. 11, p. 295–304, April 1990.

[203] E. P. Baltsavias and M. Hahn, "Integration of image analysis and GIS," in *Joint EARSeL/ISPRS Workshop" Fusion of Sensor Data, Knowledge Sources and Algorithms for Extraction and Classification of Topographic Objects"*, 1999.

[204] P. Lohmann, A. Koch and M. Schaeffer, “Approaches to the filtering of laser scanner data,” *International Archives of Photogrammetry and Remote Sensing,* vol. 33, p. 540–547, 2000.

[205] P. Lohmann, “Segmentation and filtering of laser scanner digital surface models,” *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences,* vol. 34, p. 311–316, 2002.

[206] J. L. Mesa-Mingorance and F. J. Ariza-López, “Accuracy Assessment of Digital Elevation Models (DEMs): A Critical Review of Practices of the Past Three Decades,” *Remote Sensing,* vol. 12, p. 2630, August 2020.

[207] G. D. Malaperdas and V. V. Panagiotidis, “The aspects of Aspect: Understanding land exposure and its part in geographic information systems analysis,” *Energy & Environment,* vol. 29, p. 1022–1037, 2018.

[208] T. Masataka and S. Ryousuke, *Contour Line Interpolation by using Buffering Method,* 1995.

[209] A. L. Clarke, A. Gruen and J. C. Loon, “Application of contour data for generating high fidelity grid digital elevation models.,” in *5 th Int. Symp. Comp. Assisted Cartography & Int. Soc. Photogrammetry and Remote Sensing Commission IV*, 1982.

[210] H. Taud, J.-F. Parrot and R. Alvarez, “DEM generation by contour line dilation,” *Computers & Geosciences,* vol. 25, p. 775–783, August 1999.

[211] Q. Wang and R. Ward, “A contour-preserving image interpolation method,” in *Proceedings 2003 International Conference on Image Processing (Cat. No.03CH37429).*

[212] D. Watson, Contouring: a guide to the analysis and display of spatial data, Elsevier, 2013.

[213] C. Gold and M. Dakowicz, “Digital elevation models from contour lines: Improving Accuracy by incorporating skeleton points and slope information,” *GIM International,* vol. 17, pp. 56-59, 2003.

[214] Z. W. Yu and H. Q. Tan, “Regionalized interpolation: A new approach to surface map reconstruction,” *Terra Nostra,* vol. 3, p. 513–517, 2002.

[215] G. L. Heritage, D. J. Milan, A. R. G. Large and I. C. Fuller, “Influence of survey strategy and interpolation model on DEM quality,” *Geomorphology,* vol. 112, p. 334–344, November 2009.

[216] A. Bouillon, M. Bernard, P. Gigord, A. Orsoni, V. Rudowski and A. Baudoin, “SPOT 5 HRS geometric performances: Using block adjustment as a key issue to improve quality of DEM generation,” *ISPRS Journal of Photogrammetry and Remote Sensing,* vol. 60, p. 134–146, May 2006.

[217] M. B. Gousie and W. R. Franklin, “Constructing a DEM from grid-based data by computing intermediate contours,” in *Proceedings of the 11th ACM international symposium on Advances in geographic information systems*, 2003.

[218] F. Huang, D. Liu, X. Tan, J. Wang, Y. Chen and B. He, “Explorations of the implementation of a parallel IDW interpolation algorithm in a Linux cluster-based parallel GIS,” *Computers & Geosciences,* vol. 37, p. 426–434, April 2011.

[219] H. B. N. Hynes, “The stream and its valley,” *SIL Proceedings, 1922-2010,* vol. 19, p. 1–15, October 1975.

[220] A. D. Ward, S. W. Trimble, S. R. Burckhard and J. G. Lyon, Environmental Hydrology, CRC Press, 2015.

[221] J. M. Omernik, “THE MISUSE OF HYDROLOGIC UNIT MAPS FOR EXTRAPOLATION, REPORTING, AND ECOSYSTEM MANAGEMENT1,” *JAWRA Journal of the American Water Resources Association,* vol. 39, p. 563–573, June 2003.

[222] Bhuiyan, M. Bhuiyan and Hiromitsu, “An accurate method for finding the control points of bezier curves,” January 1997.

[223] E. V. Giusti and W. J. Schneider, The distribution of branches in river networks, US Government Printing Office, 1965.

[224] W. Lichtner, “Automatic Digitization of Conventional Maps and Cartographic Pattern Recognition”.

[225] T. Liu, Q. Miao, P. Xu, J. Song and Y. Quan, “Color topographical map segmentation Algorithm based on linear element features,” *Multimedia Tools and Applications,* vol. 75, p. 5417–5438, February 2015.

[226] R. Samet and Ç. Varol, “AN IMPLEMENTATION OF AUTOMATIC CONTOUR LINE EXTRACTION FROM SCANNED DIGITAL TOPOGRAPHIC MAPS,” 2010.

[227] Y. Jiang, H. Wang, Y. Cai and B. Fu, “Salt and Pepper Noise Removal Method Based on the Edge-Adaptive Total Variation Model,” *Frontiers in Applied Mathematics and Statistics,* vol. 8, June 2022.

[228] E. R. Davies and A. P. N. Plummer, “Thinning algorithms: A critique and a new methodology,” *Pattern Recognition,* vol. 14, p. 53–63, January 1981.

[229] T. Pavlidis, “A flexible parallel thinning algorithm,” 1981.

[230] M. P. Pradhan and M. K. Ghose, “Improvised interpolation of contour lines using spider weaving approach,” *International Journal on Computer Science and Engineering,* vol. 5, p. 782, 2013.

[231] K. L. Bowden and J. R. Wallis, “Effect of stream-ordering technique on Horton's laws of drainage composition,” *Geological Society of America Bulletin,* vol. 75, p. 767–774, 1964.

[232] A. E. SCHEIDEGGER, “EFFECT OF MAP SCALE ON STREAM ORDERS,” *International Association of Scientific Hydrology. Bulletin,* vol. 11, p. 56–61, September 1966.

[233] S. Spinello and P. Guitton, “Contour Line Recognition From Scanned Topographic Maps,” 2004.

[234] S. Yadav and J. Indu, “Estimation of vertical accuracy of Digital Elevation Models over complex terrains of Indian subcontinent,” in *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2016.

[235] D. Jinyang and Z. Yumei, “Automatic extraction of contour lines from scanned topographic map,” in *IEEE International IEEE International IEEE International Geoscience and Remote Sensing Symposium, 2004. IGARSS \textquotesingle04. Proceedings. 2004*.

[236] A. Soycan and M. Soycan, “Digital elevation model production from scanned topographic contour maps via thin plate spline interpolation,” *Arabian Journal for Science and Engineering,* vol. 34, p. 121, 2009.

[237] C. Li, P. Guo, P. Wu and X. Liu, “Extraction of Terrain Feature Lines from Elevation Contours Using a Directed Adjacent Relation Tree,” *ISPRS International Journal of Geo-Information,* vol. 7, p. 163, April 2018.

[238] N. Amenta, M. Bern and D. Eppstein, “The Crust and the Beta-Skeleton: Combinatorial Curve Reconstruction,” *Graphical Models and Image Processing,* vol. 60, p. 125–135, March 1998.

[239] J. Pouderoux and S. Spinello, “Global Contour Lines Reconstruction in Topographic Maps,” in *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007) Vol 2*, 2007.

[240] J. Sanchez-Reyes, “Comment on Defining a curve as a Bezier curve,” *Journal of Taibah University for Science,* vol. 14, p. 849–850, January 2020.

[241] G. Hu, J. Wu and X. Qin, “A novel extension of the Bézier model and its applications to surface modeling,” *Advances in Engineering Software,* vol. 125, p. 27–54, November 2018.

[242] X. Li, H. Shen, R. Feng, J. Li and L. Zhang, “DEM generation from contours and a low-resolution DEM,” *ISPRS Journal of Photogrammetry and Remote Sensing,* vol. 134, p. 135–147, December 2017.

[243] Contour Lines and Topo Maps [Online] Available at: , “https://www.greenbelly.co/pages/contour-lines,” [Accesses on: 6/04/2019].

[244] H.-h. Zhao and H. Liu, “Multiple classifiers fusion and CNN feature extraction for handwritten digits recognition,” *Granular Computing,* vol. 5, p. 411–418, February 2019.

[245] T. Tan, Y. Qian, H. Hu, Y. Zhou, W. Ding and K. Yu, “Adaptive Very Deep Convolutional Residual Network for Noise Robust Speech Recognition,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing,* vol. 26, p. 1393–1405, August 2018.

[246] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai and T. Chen, “Recent advances in convolutional neural networks,” *Pattern Recognition,* vol. 77, p. 354–377, May 2018.

[247] I. Arel, D. C. Rose and T. P. Karnowski, “Deep Machine Learning - A New Frontier in Artificial Intelligence Research [Research Frontier],” *IEEE Computational Intelligence Magazine,* vol. 5, p. 13–18, November 2010.

[248] K. Du, Y. Deng, R. Wang, T. Zhao and N. Li, “SAR ATR based on displacement- and rotation-insensitive CNN,” *Remote Sensing Letters,* vol. 7, p. 895–904, June 2016.

[249] M. D. Zeiler, D. Krishnan, G. W. Taylor and R. Fergus, “Deconvolutional networks,” in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010.

[250] Z.-Q. Zhao, B.-J. Xie, Y.-m. Cheung and X. Wu, “Plant Leaf Identification via a Growing Convolution Neural Network with Progressive Sample Learning,” in *Computer Vision – ACCV 2014*, Springer International Publishing, 2015, p. 348–361.

[251] P. F. Felzenszwalb, R. B. Girshick, D. McAllester and D. Ramanan, “Object Detection with Discriminatively Trained Part-Based Models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 32, p. 1627–1645, September 2010.

[252] R. Dalai and K. K. Senapati, “Comparison of Various RCNN techniques for Classification of Object from Image,” *International Research Journal of Engineering and Technology (IRJET),* vol. 4, 2017.

[253] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan and S. Belongie, “Feature pyramid networks for object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[254] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[255] K. He, X. Zhang, S. Ren and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[256] A. Graves, S. Fernández, F. Gomez and J. Schmidhuber, “Connectionist temporal classification,” in *Proceedings of the 23rd international conference on Machine learning - ICML \textquotesingle06*, 2006.

[257] X.-n. Li, M. Fu and S.-y. Wang, “An integrating KFCM and threshold segmentation fusion method for contour lines extraction from color scanned topographical maps,” in *2014 International Conference on Information Science, Electronics and Electrical Engineering*, 2014.

[258] P. Arrighi and P. Soille, “From scanned topographic maps to digital elevation models,” *Proceedings of Geovision,* vol. 99, p. 1–4, 1999.

[259] M. Olson, A. Wyner and R. Berk, “Modern neural networks generalize on small data sets,” *Advances in neural information processing systems,* vol. 31, 2018.

[260] B. Fu, X. Zhao, Y. Li, X. Wang and Y. Ren, “A convolutional neural networks denoising approach for salt and pepper noise,” *Multimedia Tools and Applications,* vol. 78, p. 30707–30721, August 2018.

[261] M. Liao, B. Shi and X. Bai, “Textboxes++: A Single-Shot Oriented Scene Text Detector,” *IEEE Transactions on Image Processing,* vol. 27, p. 3676–3690, August 2018.

[262] R. Ma, “DEM Generation and Building Detection from Lidar Data,” *Photogrammetric Engineering & Remote Sensing,* vol. 71, p. 847–854, July 2005.

[263] W. Chen, L. Sui, Z. Xu and Y. Lang, “Improved Zhang-Suen thinning algorithm in binary line drawing applications,” in *2012 International Conference on Systems and Informatics (ICSAI2012)*, 2012.

[264] T. Ghircoias and R. Brad, “CONTOUR LINES EXTRACTION AND RECONSTRUCTION FROM TOPOGRAPHIC MAPS,” 2010.