# Chapter 4
# Gradient Descent Sea Lion Optimization

## 4.1 Introduction

This chapter covers the method of road network extraction from HR RS images by newly designed GDSLO optimization techniques. Firstly input is pre-process to remove the noise from the input database. Then after road region detection is performed on the pre –processed images. Finally FCN is used to detect road edges and road centerline from road segmented output. The organization of this chapter is as follows. The GDSLO method is covered in Section 1, the results and discussion are presented in Section 2, a comparison of other approaches is discussed in Section 3, and the experimental results are shown in Section 4.

### 4.1.1 Proposed Deep Learning Enabled Approach for Road Segmentation and Edge-Centerline Extraction

This method primary objective is to create a technique for extracting roads from HR remote sensing images called GDSLO-based U-Net. This method for road network extraction includes the several steps such as pre-processing, the road surface detection, edge detection of the road, and extracting the centerline of the road are some of the processes included in this method. The input image is initially sent to the pre-processing stage, where it is cleaned up by using the T2FCS filter to get rid of noise and outside artifacts(Kumar, 2019) .The pre-processed image is then delivered to the road segmentation stage, where the road surfaces are successfully recovered using U-Net(Ronneberger O, 2015). By use of newly developed optimization algorithm known as GDSLO is used to train the U-Net. SGD and SLnO (Masadeh, 2019) are integrated in the design of the proposed GDSLO(Kothari, 2022). Following road segmentation, road edge identification using FCN(J. LONG, 2015) is performed based on the pre-processed output and segmented output. The pre-processed input image and the road segmented output are taken into consideration to successfully extract the road centerline.

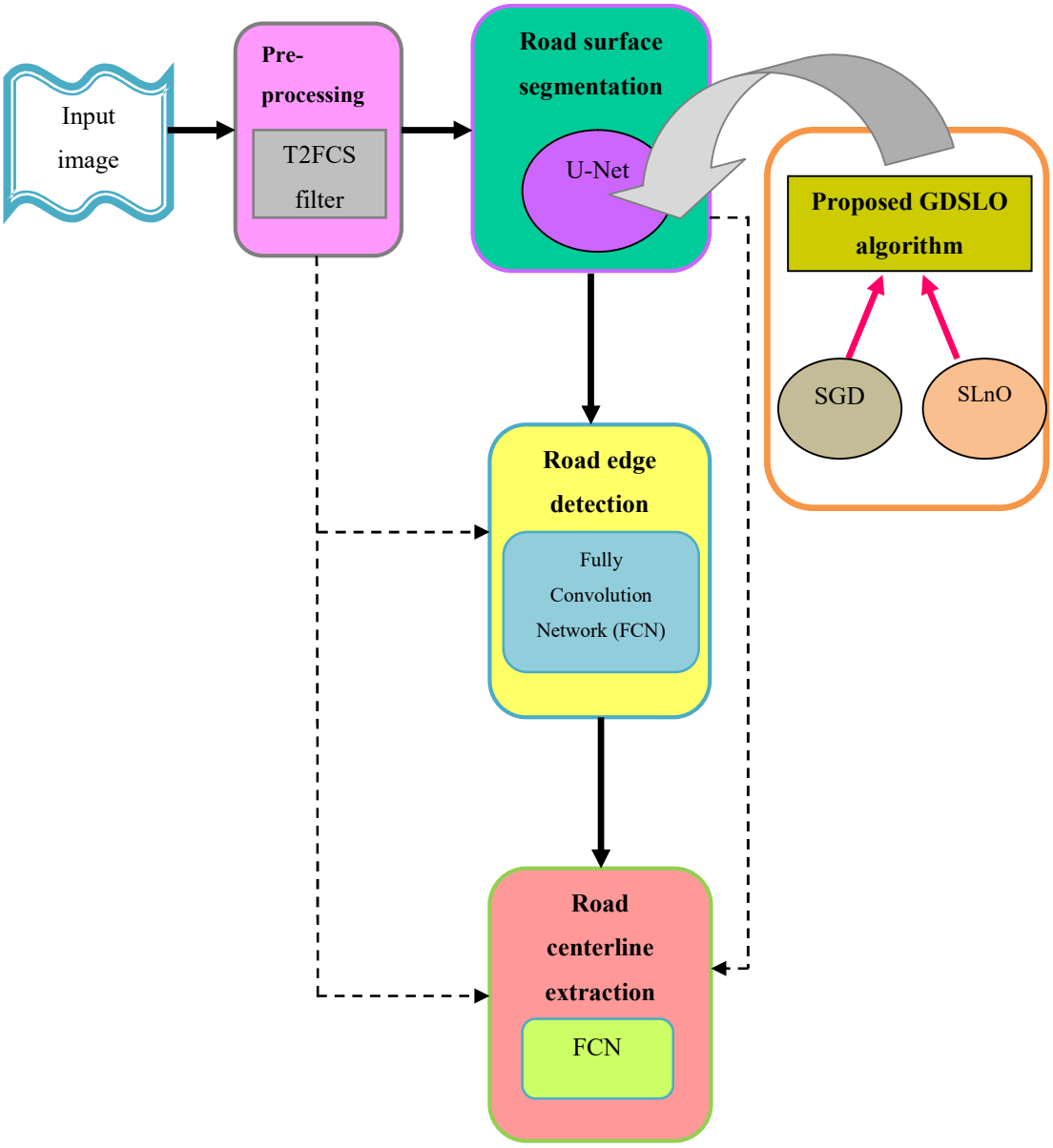



**Figure 4.1: Schematic View of The Established Deep Learning Enabled Method for Road Segmentation and Edge-Midway Extraction**

### 4.1.2 Acquisition of Input

Input images are collected from the database and assume the set of data $D$ with $\psi$ amount of input, which can be represented as,

$$D = \{S_1, S_2, ...S_z, ...S_\psi\} \tag{4.1}$$

where, $D$ symbolize dataset, total number of images is represented by $\psi$ , and $S_z$ represents the input image at $z^{\text{th}}$ index , which is presented as an input to pre-processing stage.

### 4.1.3 Pre-processing using T2FCS Filter

Pre-processing stage of the proposed method aim is to eliminate noise or falsifications from the input images. The input image is pre-processed in this case using the T2FCS filter(Kumar, 2019) .Each associate is designated as and based on a type II fuzzy system in the design of the T2FCS filter, which incorporates Cuckoo Search (CS) optimization. Here, type 2 fuzzy groups are modifying the image to remove noise without affecting the image pixels. Moreover, a type 2 fuzzy system is used to handle a significant quantity of gradation of uncertainty. T2FCS filter thereby aids in the detection of noises from distorted input. $P_i$ is a result of pre-processing an image that has been tailored for road segmentation.

### 4.1.4 Road Segmentation using GDSLO-based U-Net

Road segmentation is carried out in this phase in order to extract the road surfaces. Segmentation is a technique for dividing the result of pre-processing into different parts, such as image pixels or road object segments. The U-Net methodology is used in this instance to segment the roads(Ronneberger O, 2015). However, the proposed GDSLO method, which is hybridization of SGD(Qi Qian, 2015) and SLnO (Masadeh, 2019) is used for U-Net training.

#### 4.1.4.1 Training of U-Net using Developed GDSLO Algorithm

This section explains how the proposed optimization technique, known as GDSLO, was used to train the U-Net(Ronneberger O, 2015). Here, the SGD and SLnO hybridization results in a novel designed GDSLO (Masadeh, 2019). The sea lions' successful hunting tactics served as the model for the SLnO (Masadeh, 2019) algorithm. Additionally, sea lions' whiskers help it recognize targets. The SLnO approach improves generalization capability and avoids premature convergence, enabling accurate identification of the target that become the ideal solution. Moreover

, SGD (Qi Qian, 2015) is an iterative approach used for optimizing the objective function with proper smoothness properties.

Additionally, the combination of SGD with SLnO improves the system's overall performance and achieves the best resolution by minimizing the computational challenges. The suggested GDSLO algorithm's algorithmic phases are shown below.

***1) Initialization***

Initializing a solution $P$ can be expressed as,

$$P = \{P_1, P_2, \ldots, P_m, \ldots, P_z;\ 1 \le m \le z\} \tag{4.2}$$

where, $z$ denotes complete solutions, and $P_m$ signifies the $m^{th}$ result.

***2) Fitness evaluation***

The fitness function computes the optimal solution such that the fitness function with the smallest error value is considered to be the best solution. The fitness equation is formulated as,

$$F = \frac{1}{\ell}\sum_{\delta=1}^{\ell}\left[T - O_r^*\right]^2 \tag{4.3}$$

where, $F$ denotes fitness measure, $O_r^*$ represents output generated by U-Net, and $T$ implies target outcome.

***3) Detecting and tracking phase***

The form, size, and position of prey are predicted by sea lions' whiskers using the SLnO (Masadeh, 2019) algorithm. The SLnO considers the target prey to be the current optimal solution or a close. Additionally, the sea lion moves toward its intended prey and is described as,

$$P(g+1) = S(g) - L.Q \tag{4.4}$$

$$P(g+1) = S(g) - |2MS(g) - P(g)|.Q \tag{4.5}$$

Consider $\vec{M} > 0.5$,

$$P(g+1) = S(g) - \left(2M.S(g) - P(g)\right).Q \tag{4.6}$$

$$P(g+1) = S(g)(1-2M.Q) + P(g).Q \tag{4.7}$$

The optimal solution can be obtained by minimizing the optimization problems by combining SLnO with SGD. The usual SGD equation is as follows:

$$P(g+1) = P(g) - \upsilon_h \nabla R(P(g)) \tag{4.8}$$

$$P(g) = P(g+1) + \upsilon_h \nabla R(P(g)) \tag{4.9}$$

Where, the learning rate is represented as $\upsilon_h$, $\nabla R(P(g))$ that shows the gradient of the position vector loss function.

By substituting (4.9) in (4.7), the equation becomes,

$$P(g+1) = S(g)(1-2M.Q) + (P(g+1) + \upsilon_h \nabla R(P(g))).Q \tag{4.10}$$

$$P(g+1) = S(g)(1-2M.Q) + Q \times P(g+1) + \upsilon_h \nabla R(P(g))Q \tag{4.11}$$

$$P(g+1) - Q \times P(g+1) = S(g)(1-2M.Q) + \upsilon_h \nabla R(P(g))Q \tag{4.12}$$

$$P(g+1)(1-Q) = S(g)(1-2M.Q) + \upsilon_h \nabla R(P(g))Q \tag{4.13}$$

$$P(g+1) = \frac{1}{(1-Q)}\left[S(g)(1-2M.Q) + \upsilon_h \nabla R(P(g))Q\right] \tag{4.14}$$

where, $(g+1)$ signifies the next iteration, $M$ denotes the random vector within the range [0.5,1], $S(g)$ implies the position vector of target prey, $P(g)$ symbolize the position vector of the sea lion, $L$ signifies the total distance between the target prey and sea lion, $Q$ and specifies the parameter that linearly reduces from 2 to 0 for each iteration.

***4) Vocalization stage***

When sea lions hunt and chase together as a subgroup, they communicate with one another by vocalizing. When a sea lion spots a potential meal, it signals the rest of the group to surround and attack the target, which is given as,

$$P_{leader} = \left|(C_1 + (1 + C_2))/C_2\right| \tag{4.15}$$

Where, $C_1$ and $C_2$ represent the speed of sounds in water and air respectively and $P_{leader}$ denotes the speed of sea lion's sound. As a result, speed of sound in water is defined as,

$$\vec{C}_1 = Sin\theta \tag{4.16}$$

Speed of sound in air is expressed as,

$$\vec{C}_2 = Sin\phi \tag{4.17}$$

***5) Attacking stage***

To construct the sea lions' hunting behavior, two distinct phases called diminishing encircling and circle updating were identified.

***a) Dwindling encircling strategy***

This dwindling encircling behavior relies on $Q$ that supports the sea lion leader to move in the track of prey and encircle them.

***b) Circle updating position***

The sea lions hunt bait balls of fishes initializing from edges which are expressed as,

$$P(g+1) = |S(g) - P(g)| \cdot Cos(2\pi n) + S(g) \tag{4.18}$$

Consider $S(g) > P(g)$,

$$P(g+1) = \left(S(g) - P(g)\right) \cdot Cos(2\pi n) + S(g) \tag{4.19}$$

$$P(g+1) = S(g).\left(Cos(2\pi n) + 1\right) - P(g)Cos(2\pi n) \tag{4.20}$$

By substituting (9) in (20), the equation becomes,

$$P(g+1) = S(g).\left(Cos(2\pi n) + 1\right) - \left(P(g+1) + \upsilon_h \nabla R\left(P(g)\right)\right).Cos(2\pi n) \tag{4.21}$$

$$P(g+1) = S(g).\left(Cos(2\pi n) + 1\right) - P(g+1).Cos(2\pi n) - \upsilon_h \nabla R\left(P(g)\right).Cos(2\pi n) \tag{4.22}$$

$$P(g+1) + P(g+1).Cos(2\pi n) = S(g).\left(Cos(2\pi n) + 1\right) - \upsilon_h \nabla R\left(P(g)\right).Cos(2\pi n) \tag{4.23}$$

$$P(g+1)(1+Cos(2\pi n)) = S(g).(Cos(2\pi n)+1) - \upsilon_h \nabla R(P(g)).Cos(2\pi n) \quad (4.24)$$

$$P(g+1) = \frac{1}{1+Cos(2\pi n)}\left[S(g).(Cos(2\pi n)+1) - \upsilon_h \nabla R(P(g)).Cos(2\pi n)\right] \quad (4.25)$$

where, $Cos(2\pi n)$ denotes the behavior of sea lions swimming in circle shaped path around the prey, the distance among the finest solution and search agent is denoted as $|S(g) - P(g)|$, $||$ denotes the absolute value, and $n$ represents the random number that lies in the range of [-1, 1].

***6) Searching for prey (Exploration phase)***

In the exploitation phase, the best search agent updates sea lion locations, but in the exploration phase, sea lion locations are updated by taking into account the sea lion that was randomly selected. Nevertheless, the equation is written as

$$L = \left|2M.P_{rand}(g) - P(g)\right| \quad (4.26)$$

Where, $P_{rand}(g)$ represents randomly selected sea lions from the current population.

$$P(g+1) = P_{rand}(g) - L.Q \quad (4.27)$$

***7) Evaluation of solution feasibility***

The fitness function is used to calculate the best solution, with the best fitness value being taken into account.

***8) Termination***

Until the ideal answer is found, all the aforementioned processes are again carried out.

## 4.2 GDSLO Pseudo Code

Table 4.1 describes the GDSLO pseudo code.

| | |
|---|---|
| **1** | Input: **Solution set** $P$ **, maximum iterations** $g_{\max}$ |
| **2** | **Output:** Best solution $P(g+1)$ |
| **3** | **Begin** |
| **4** | Solution initialization |
| **5** | Choose $P_{rand}(g)$ |
| **6** | Estimate fitness for each search agent using equation (4.3) |
| **7** | $P^*$ indicates the best-fitting potential search agent |
| **8** | **If** $g < g_{\max}$ |
| **9** | Estimate $\vec{P}_{leader}$ by equation (4.16) |
| **10** | **If** $(P_{leader} < 0.25)$ |
| **11** | **If** $(\mid Q \mid < 1)$ |
| **12** | Modify the current location agent search based on equation (4.26) |
| **13** | **Else** |
| **14** | select random search agent $P_{rand}(h)$ |
| **15** | Modify the current location agent search based on equation (4.14) |
| **16** | **End if** |
| **17** | Modify the current location agent search based on equation (4.28) |
| **18** | **End if** |
| **19** | **If** search agent does not belong to any $P_{leader}$ |
| **20** | Go to step 9 |
| **21** | **Else** |
| **22** | calculate the search agent fitness based on equation (4.3) |
| **23** | Upgrade $P$ , if solution is optimum |
| **24** | **Return** $P$ |
| **25** | **End if** |
| **26** | **End if** |
| **27** | **End** |

**Table 4.1: Pseudo Code of Proposed Method GDSLO Algorithm**

The road segmentation output from GDSLO+U-Net is effective in extracting road surface and is denoted as $S_i$.

## 4.3 Road Edge and Centerline Detection using FCN

The method of extracting the boundaries of roadways with a single pixel-width is called road edge detection. The full network of road centerlines is then extracted using a centerline extraction method. In addition, road segmentation output $S_i$ and pre-processed output $P_i$ are taken into account while performing road edge detection and road centerline extraction using FCN(J. LONG, 2015) .Each layer of information in a convnet is a three-dimensional array that has a dimension of $i \times j \times k$, where $i$ and $j$ stands for spatial dimensions, and $d$ indicates the dimension of the channel or attribute. The first layer consists of image with pixel dimension $i \times j$, and $e$ color channels. Higher layer positions are known as receptive fields because they line up with the positions of the images to which they are path-connected. Additionally, Convnets are built using translation invariance. Convolution layer, pooling layer, and activation functions are among Convnets' components. These components depend on the relative spatial coordinates and operate on local input regions.

Assume the data vector $x_{ij}$ at position $(i, j)$ in a particular layer, and $y_{ij}$ for the successive layer. Nevertheless, the output result is computed using the functions represented as,

$$y_{ij} = t_{sk}\left(\left\{x_{ki+\vec{\imath},kj+\vec{\jmath}}\right\} 0 \le \vec{\imath}, \vec{\jmath} \le s\right) \tag{4.28}$$

where, the size of kernel is expressed as $s$, $k$ denotes the sub-sampling factor or strides, $t_{sk}$ calculate the type of the layer: average pooling or convolution matrix multiplication, activation function with element wise nonlinearity or spatial max with max-pooling, and so on for other kinds of layers.

By adhering to the transformation rule, which is provided as, the functional form under composition is maintained with strides and kernel dimension represented as,

$$t_{sk} \circ b_{s'k'} = (t \circ b)s' + (s-1)k', kk' \tag{4.29}$$

Since the general deep net computes the nonlinear general function, a net with just these layers measures the nonlinear filter, also known as the FCN or deep filter.

An FCN often produces results with similar spatial sizes based on inputs of any dimension. A job is also specified by an FCN with a real-valued loss function. If the loss function is the overall quantity over the spatial sizes of the last layer, $r(x;\vartheta) = \sum_{ij} r'(x_{ij};\vartheta)$ its gradient is the total sum computed over the gradients of every spatial element. Hence, the SGD on measured $r$ on overall images is same as the SGD on $r'$, considering all the receptive fields of final layer as a mini-batch.

Furthermore, layer-by-layer computation of the back-propagation and feed forward calculation over the entire image is thought to be more effective than separate patch-by-patch computation when the receptive fields substantially overlap. The output of the detected road edge-centerline is then shown as $E_i$.

## 4.4 Results and Discussion

This section identifies the execution effects of a GDSLO-based U-Net while taking into account the calculation techniques, notably precision, recall, and f-measure.

### 4.4.1 Experimental Setup

The implementation of the proposed methods is performed in the PYTHON tool with Pytorch using the RoadNet dataset (RoadNet dataset) with PC having 8GB RAM windows 10 OS, and Intel i3 core processor.The experimental settings for the designed GDSLO-enabled U-Net are displayed in Table 4. 2.

| Parameter | Value |
|---|---|
| Learning rate | 0.0001 |
| Momentum | 0.9 |
| Step size | 5 |
| Gamma | 0.1 |

| | |
|---|---|
| Number of epochs | 200 |

**Table 4.2: Experimental Parameters**

### 4.4.2 Dataset Description

This collection (RoadNet dataset) compiles different Google Earth images of Ottawa and other typical Canadian urban areas. In this collection, there are 21 sections with an approximate 8 km diameter and pixels with a three-dimensional resolution of 0.21 m. Here, each photograph has manually tagged road exteriors, road borders, and road midways. This dataset learns multi-scale and multi-level features to cope with occlusion and shadow difficulties, and the width of the road varies from 10 to 80 pixels.

### 4.4.3 Performance Metrics

The effectiveness of the well-known GDSLO-enabled U-Net approach is evaluated in terms of three evaluation criteria, namely precision, recall, and F1-measure.

***Precision:*** Precision is a measure that defines the fraction of accurately computed outcomes, and is expressed as,

$$P = \frac{a}{a+b} \tag{4.30}$$

Where, $a$ specifies true positives, and $b$ denotes false positives.

***Recall:*** It is the ratio of the true positives to the addition of true positives and false negatives, and the equation is given as,

$$R = \frac{a}{a+c} \tag{4.31}$$

Where, $a$ signifies the true positives, and $c$ denotes the false negatives.

***F1-measure:*** F1-measure calculates the mean difference between the precision and the recall measure, and is expressed as,

$$FM = 2*\left(\frac{P*R}{P+R}\right) \tag{4.32}$$

Where, $P$ shows the precision, and $R$ denotes the recall.

## 4.5 Comparative Analysis of Road Network Extraction Systems

This section explains the comparative assessment of proposed GDSLO-enabled U-Net method using the publically available dataset by varying the training data in terms of evaluation metrics. It detects the road surface, road edges, and road centerline from HR RS images and measures the performance metrics.

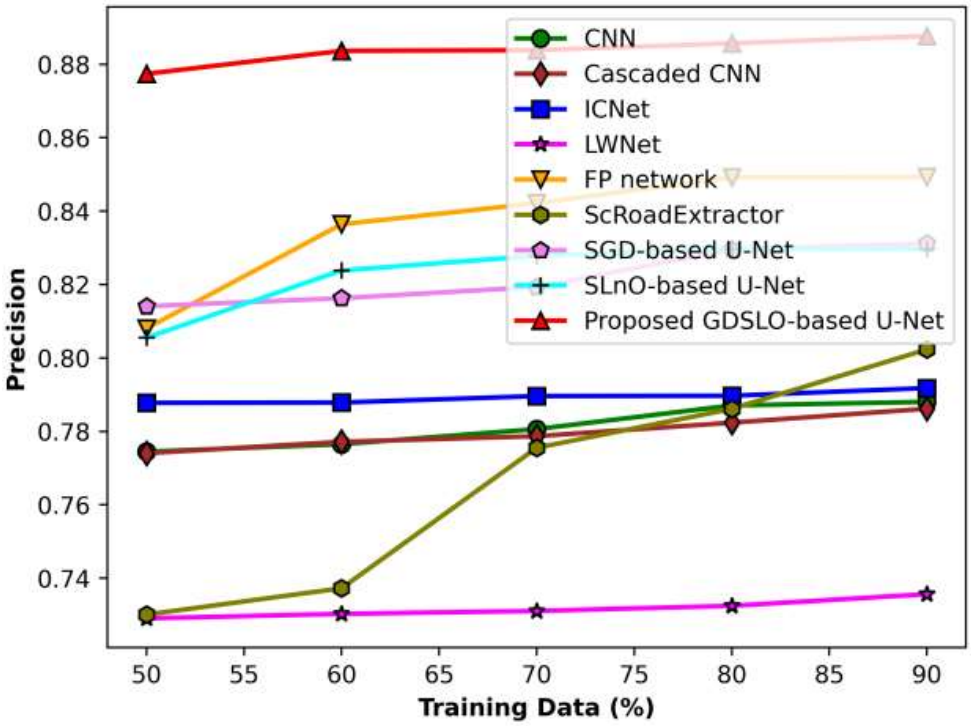
The numerous comparison methods used for the evaluation performance include CNN (Y. Liu, 2019), Cascaded CNN(Xiang, 2017) , ICNet(Senchuri, 2021) , LWNet(Ziyi Chen, 2022) , FP network(Shamsolmoali, 2020) , ScRoadExtractor(Wei Y. a., 2021) , SGD-based U-Net, SLnO-based U-Net, and newly created GDSLO-based U-Net, respectively.

### 4.5.1 Analysis Based on Road Surface Extraction

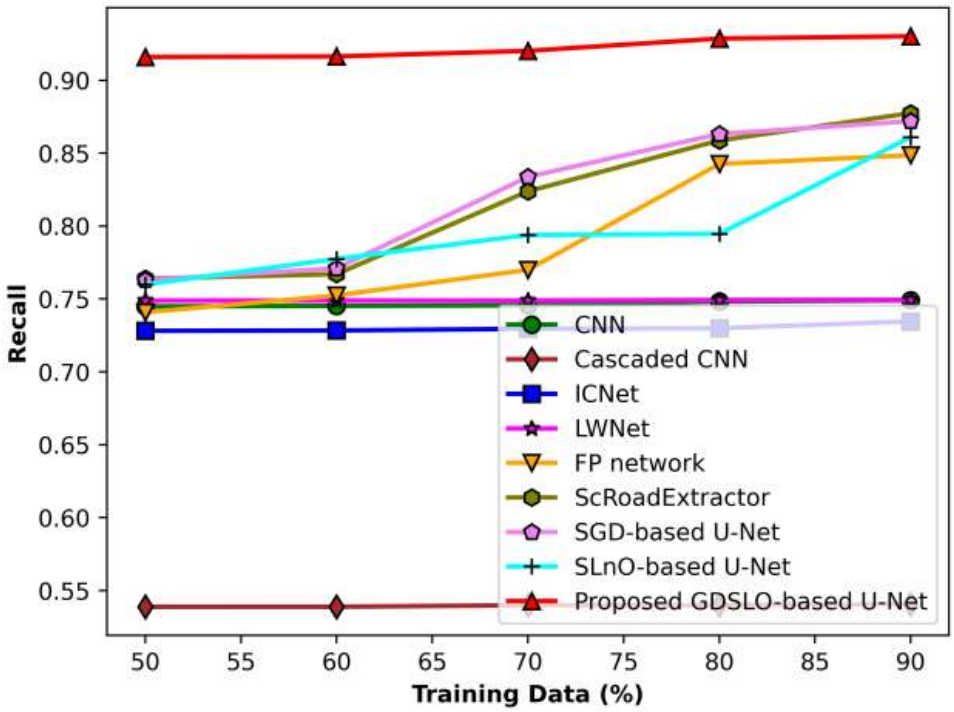
Using dataset and changing the training data related to the estimate standards, Figure 4.2 shows the study of a well-established technique. Analysis based on precision is shown in Figure 4.2a. The established GDSLO-based U-Net achieves a precision of 0.877 when analyzing 50% of the training data, however the precision values measured by current techniques, such as CNN, Cascaded CNN, ICNet, LWNet, FP network, ScRoadExtractor, SGD-based U-Net, and SLnO-based U-Net, are 0.774, 0.774 0.787, 0.729, 0.808, 0.730, 0.814, and 0.80 respectively. Figure 4.2b depicts the assessment using the recall measure. For 60% of the training data, the established GDSLO+ U-Net measured a recall value of 0.916, while the most popular techniques, including CNN, Cascaded CNN, ICNet, LWNet, FP network, ScRoadExtractor, SGD-based U-Net, and SLnO-based U-Net, computed recall values of 0.745, 0.538, 0.728, 0.748, 0.752, 0.767, 0.771 and, 0.777. Figure 4.2c shows the results of the analysis using the F1-measure.

For 70% of the training data, the CNN achieved an F1-measure value of 0.770, Cascaded CNN achieved 0.650, ICNet achieved 0.763, LWNet achieved 0.762, FP achieved 0.771, ScRoadExtractor achieved 0.772, SGD-based U-Net achieved 0.772,

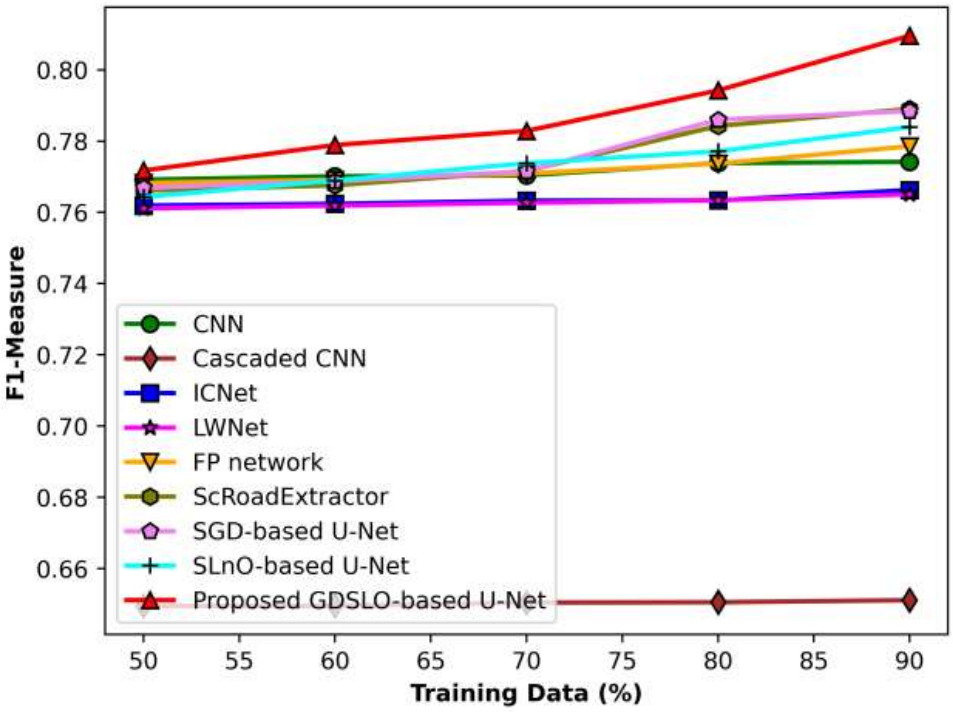
SLnO-based U-Net achieved 0.774, and developed GDSLO-based U-Net achieved 0.782.




(a) Precision




(b) Recall




(c) F1-Measure

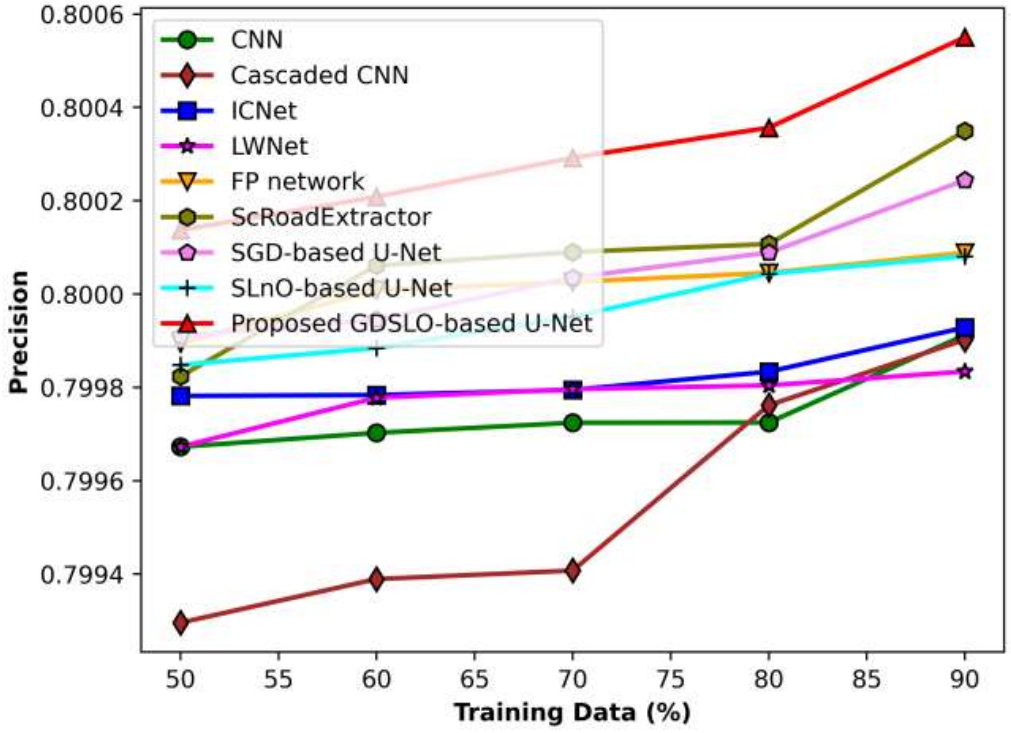
**Figure4.2: Road Surface Extraction Results**

### 4.5.2 Analysis Based on Road Edge Detection

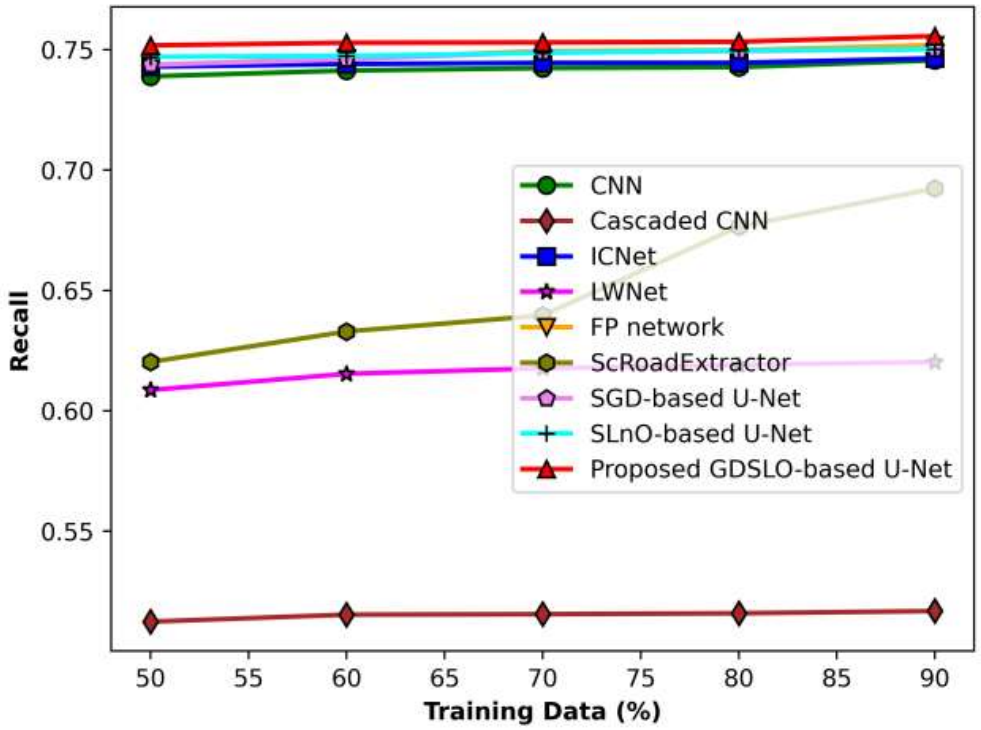
The analysis of the new developed method for road edge detection is made using dataset as shown in Figure 4.3. Precision based analysis can be represented in the Figure 4.3a. For the 60% of the training data, the newly designed GDSLO-based-U-Net calculated a precision value is 0.800. Moreover, the precision value for the other well known methods like as CNN, Cascaded CNN, ICNet, LWNet, FP network, ScRoadExtractor, SGD-based U-Net, and SLnO-based U-Net is 0.799, 0.799, 0.799, 0.799, 0.800, 0.800, 0.800, and 0.800. Recall based evaluation is present in the Figure 4.3b. Recall achieved by CNN is 0.741, Cascaded CNN is 0.515, ICNet is 0.744, LWNet is 0.615, FP network is 0.749, ScRoadExtractor is 0.640, SGD-based U-Net is 0.749, SLnO-based U-Net is 0.749, and created GDSLO-based U-Net is 0.752 with 70% of the training data.

Figure 4.3c illustrates how the F1-measure can be used to conduct the analysis. With 80% of the training data, the established GDSLO-based U-Net achieves an F1-measure of 0.785. The F1-measure, meanwhile, is 0.769, 0.640, 0.766, 0.698, 0.772,

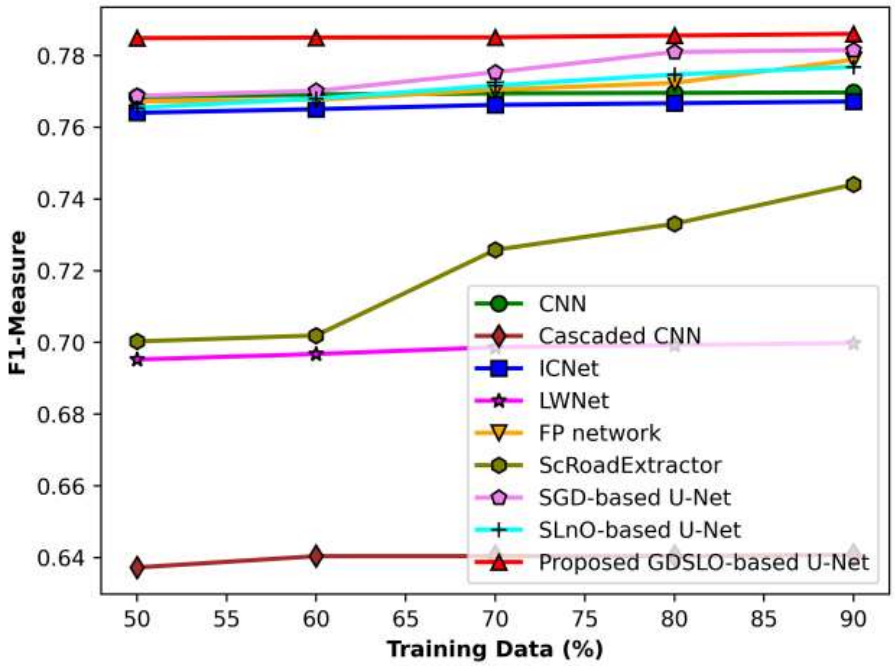
0.733, 0.781, and 0.775 according to popular approaches such as CNN, Cascaded CNN, ICNet, LWNet, FP network, ScRoadExtractor, and SGD-based U-Net.




**(a)** Precision




**(b)** Recall



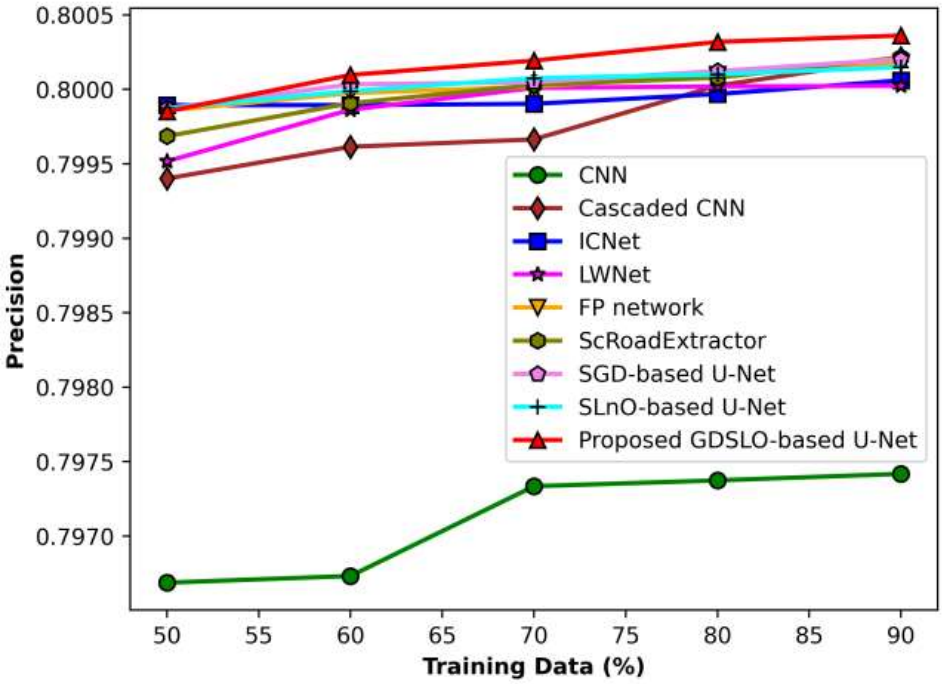

**(c)** F1-Measure

**Figure 4.3: Road Edge Detection Result**

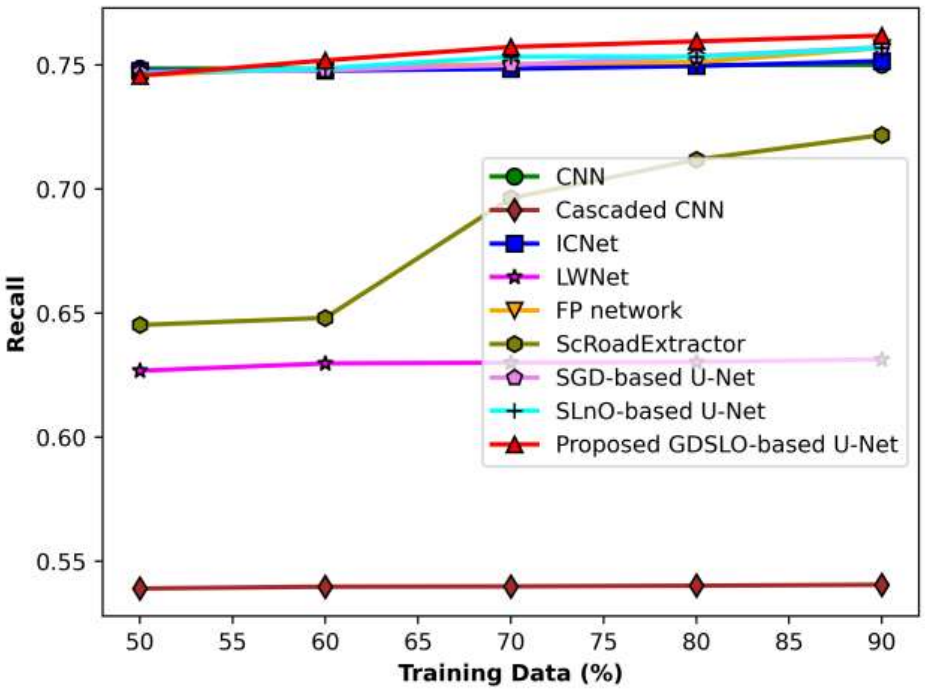
### 4.5.3 Analysis Based on Road Centreline Detection

The analysis of the newly designed GDSLO technique depending upon the various percentages of the training data is revealed in Figure 4.4. Precision based evaluation is representing in the Figure 4.4a. when 50% of the training data is consider , established GDSLO+U-Net evaluates a precision of 0.7998, on the other hand prevailing methods such as CNN, Cascaded CNN, ICNet, LWNet, FP network, ScRoadExtractor, SGD-based U-Net, and SLnO-based U-Net, computed a precision value of 0.7966, 0.7994, 0.7998, 0.7995, 0.7998, 0.7996, 0.7998, and 0.7998. Figure 4.4b shows the analysis based on recall parameter. For 60% of the training data ,newly designed GDSLO+ U-Net recall value is evaluate by 0.751, however the recall value calculated by the current existing methods, such as CNN, Cascaded CNN, ICNet, LWNet, FP network, ScRoadExtractor, SGD-based U-Net, and SLnO-based U-Net, is 0.748, 0.539, 0.747, 0.629, 0.748, 0.648, 0.747, and 0.748 respectively. Analysis is done with regard to F1-measure is depicted in Figure 4.4c. when 70% of the training data is considered , F1-value is measured by CNN, Cascaded CNN, ICNet, LWNet, FP network, ScRoadExtractor, SGD-based U-Net, SLnO-based U-

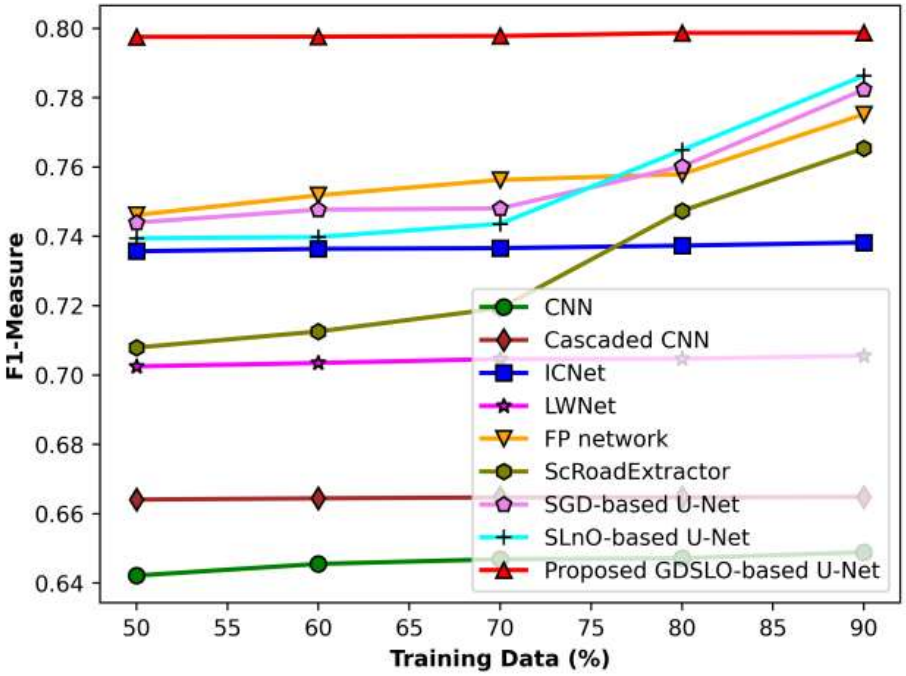
Net, and developed GDSLO-based U-Net is 0.646, 0.664, 0.736, 0.704, 0.756, 0.719, 0.748, 0.743, and 0.797 respectively.




(a) Precision




(a) Recall

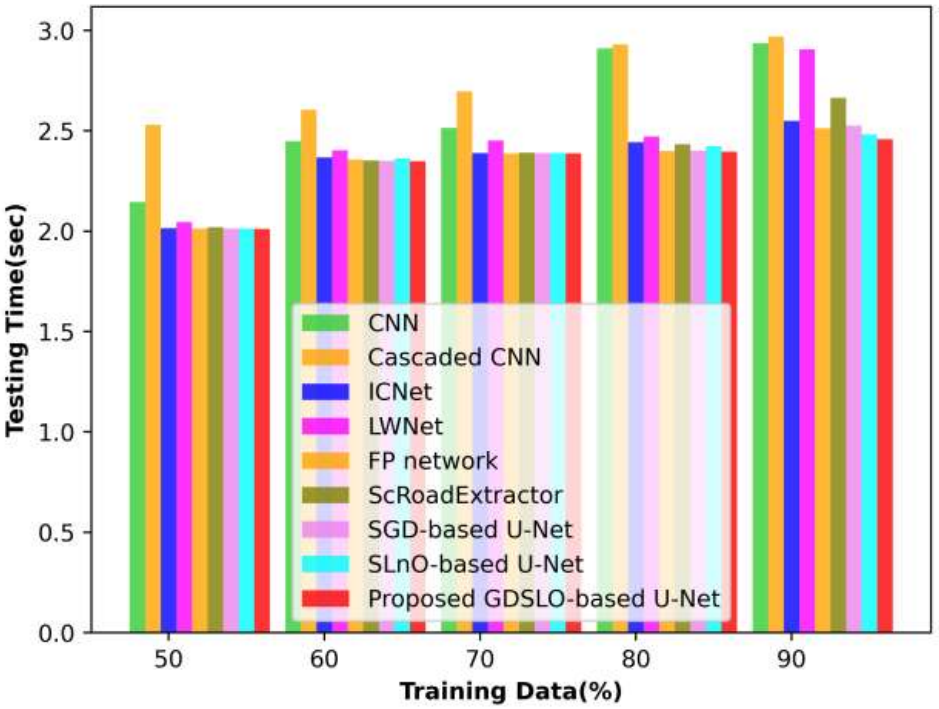



(b) F1- Measure

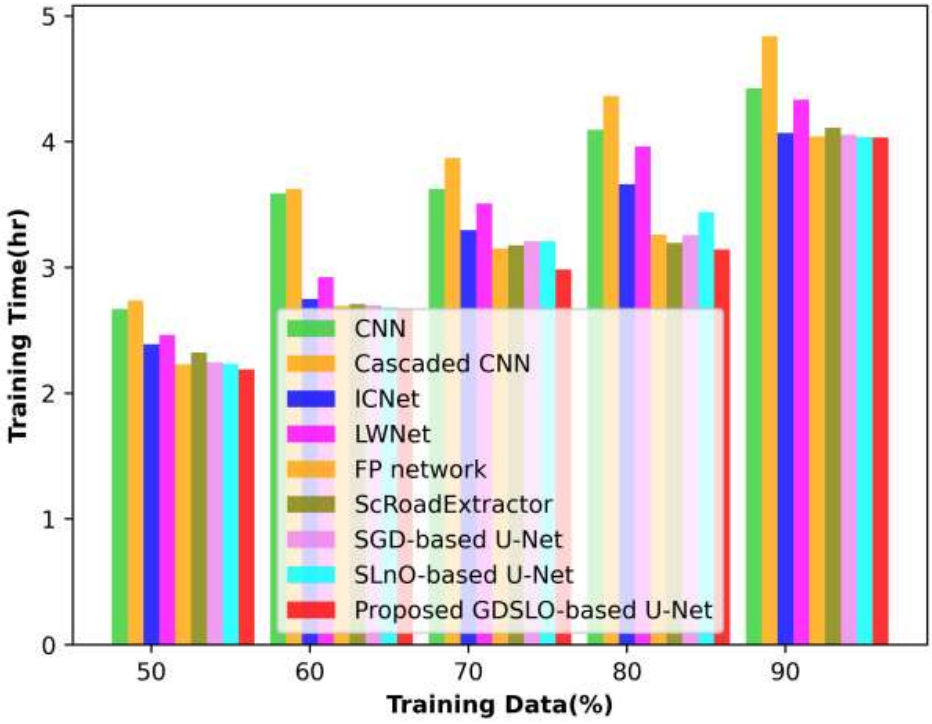
**Figure 4.4: Road Centreline Detection Results**

### 4.5.4 Analysis Based on Testing Time and Training Time

Figure 7shows the analysis based on testing time and training time of the developed GDSLO method with other current existing methods. The analysis based on the testing time depicted in the Figure 4.5a. For 60% of training data, value of testing time calculated by CNN, Cascaded CNN, ICNet, LWNet, FP network, ScRoadExtractor, SGD-based U-Net, and SLnO-based U-Net, and developed GDSLO+ U-Net is 2.447sec, 2.606sec, 2.366sec, 2.402sec, 2.352sec, 2.351sec, 2.362sec, and 2.349sec. The assessment depending on training time is depicted in Figure 4.5b. In varying training data 70%, the established GDSLO-based U-Net measured a training time of 2.984hrs, whereas the training time computed by the CNN, Cascaded CNN, ICNet, LWNet, FP network, ScRoadExtractor, SGD-based U-Net, and SLnO-based U-Net, 3.625hrs, 3.872hrs, 3.297hrs, 3.509hrs, 3.152hrs, 3.175hrs, 3.210hrs, and 3.210hrs.




(a) Testing time




(a) Training time

**Figure 4.5: Testing and Training Time Comparative Result of GDSLO**

## 4.6 GDSLO Results and Comparative Analysis

The different comparison methods used for the evaluation are, in order, CNN(Y. Liu, 2019) , Cascaded CNN(Xiang, 2017), ICNet(Senchuri, 2021), LWNet (Chen, 2017), FP network (Shamsolmoali, 2020), ScRoadExtractor(Wei Y. a., 2021) , SGD-based U-Net, SLnO-based U-Net, and designed GDSLO-based U-Net by varying the training data of 90% as shown in Table 4.3.

For the road surface extraction, the developed GDSLO-based U-Net measured a value of precision, recall, and F1-measure is 0.888, 0.930, and 0.809 respectively. Likewise, the developed GDSLO-based U-Net calculated a precision value 0.801, recall value 0.756 and F1 measure value 0.786 for road edge detection. In addition, the precision, recall and F1-measure value achieved by the developed GDSLO-based U-Net for road centreline detection is, 0.800, 0.762, and 0.799 respectively.

| **Methods/Metrics** | | **Precision** | **Recall** | **F1-measure** |
|---|---|---|---|---|
| **Road surface extraction** | ***CNN*** | 0.788 | 0.749 | 0.774 |
| | ***Cascaded CNN*** | 0.786 | 0.541 | 0.651 |
| | ***ICNet*** | 0.792 | 0.735 | 0.766 |
| | ***LWNet*** | 0.736 | 0.749 | 0.765 |
| | ***FP network*** | 0.849 | 0.849 | 0.779 |
| | ***ScRoadExtractor*** | 0.802 | 0.877 | 0.789 |
| | ***SGD-based U-Net*** | 0.831 | 0.872 | 0.788 |
| | ***SLnO-based U-Net*** | 0.830 | 0.861 | 0.784 |
| | ***Proposed GDSLO-based U-Net*** | **0.888** | **0.930** | **0.810** |
| **Road edge detection** | ***CNN*** | 0.799 | 0.746 | 0.770 |
| | ***Cascaded CNN*** | 0.799 | 0.517 | 0.641 |
| | ***ICNet*** | 0.799 | 0.746 | 0.767 |
| | ***LWNet*** | 0.799 | 0.620 | 0.700 |
| | ***FP network*** | 0.800 | 0.752 | 0.779 |
| | ***ScRoadExtractor*** | 0.800 | 0.692 | 0.744 |
| | ***SGD-based U-Net*** | 0.800 | 0.750 | 0.782 |

| | | | | |
|---|---|---|---|---|
| | ***SLnO-based U-Net*** | 0.800 | 0.750 | 0.777 |
| | ***Proposed GDSLO-based U-Net*** | **0.801** | **0.756** | **0.786** |
| **Road centreline detection** | ***CNN*** | 0.797 | 0.750 | 0.649 |
| | ***Cascaded CNN*** | 0.800 | 0.541 | 0.665 |
| | ***ICNet*** | 0.800 | 0.752 | 0.738 |
| | ***LWNet*** | 0.800 | 0.631 | 0.706 |
| | ***FP network*** | 0.800 | 0.757 | 0.775 |
| | ***ScRoadExtractor*** | 0.800 | 0.722 | 0.765 |
| | ***SGD-based U-Net*** | 0.800 | 0.757 | 0.782 |
| | ***SLnO-based U-Net*** | 0.800 | 0.757 | 0.786 |
| | ***Proposed GDSLO-based U-Net*** | **0.800** | **0.762** | **0.799** |

**Table 4.3: Comparative Analysis of Well Known Methods with GDSLO**

The relative discussion of the developed approach in comparison to the widely used methods, including CNN, Cascaded CNN, ICNet, and LWNet, FP network, ScRoadExtractor, SGD-based U-Net, and SLnO-based U-Net, for testing time and training time, is illustrated in Table 4.4. The tested time for the developed GDSLO-based U-Net was 2.458 seconds, while the training time was 4.566 hours.

| **Metrics** | **CNN** | **Cascaded CNN** | **ICNet** | **LWNet** | **FP network** | **ScRoad Extractor** | **SGD-based U-Net** | **SLnO-based U-Net** | **Proposed GDSLO-based U-Net** |
|---|---|---|---|---|---|---|---|---|---|
| **Testing time (sec)** | 2.936 | 2.969 | 2.549 | 2.905 | 2.513 | 2.664 | 2.525 | 2.482 | **2.458** |
| **Training time (hr)** | 4.784 | 4.900 | 4.783 | 4.754 | 4.041 | 4.112 | 4.055 | 4.036 | **4.566** |

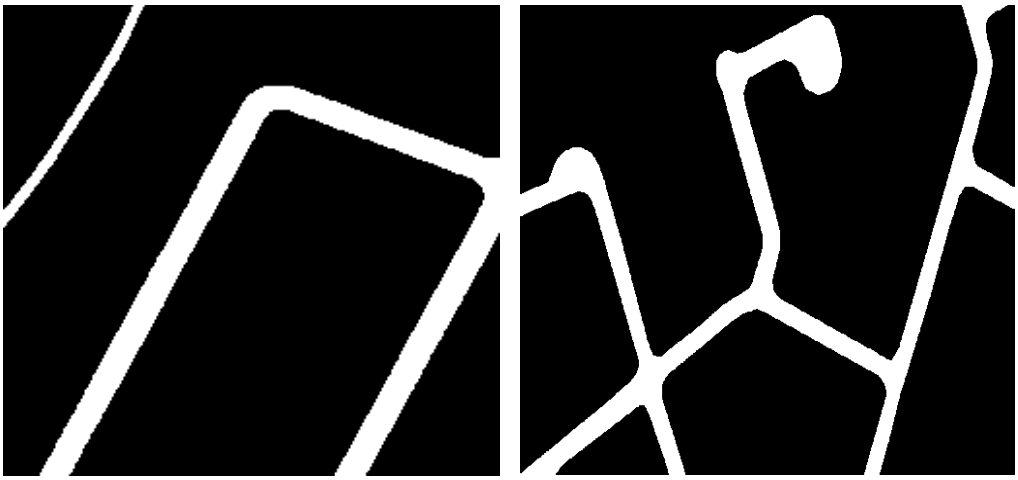
**Table 4.4: Comparative Analysis of GDSLO Method with Existing Methods Based on Testing Time and Training Time**

### 4.6.1 Experimental Outcomes of GDSLO

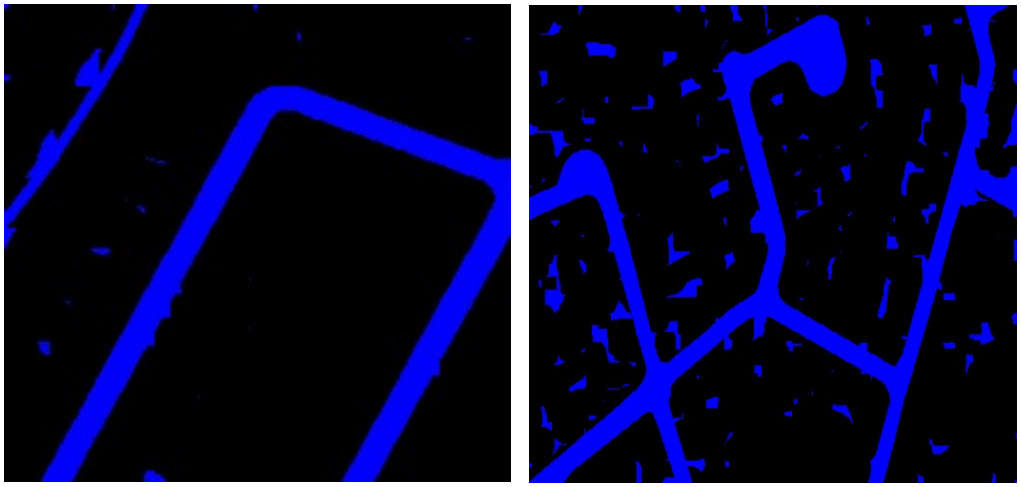
The above Figures 4.6, 4.7 and 4.8 shows the visual output of the proposed GDSLO-based U-Net for road surface, edges and centerline detection from RS images. The

proposed GDSLO based U-Net giving better performance compare to existing literature methods.

**(a) Original Images**



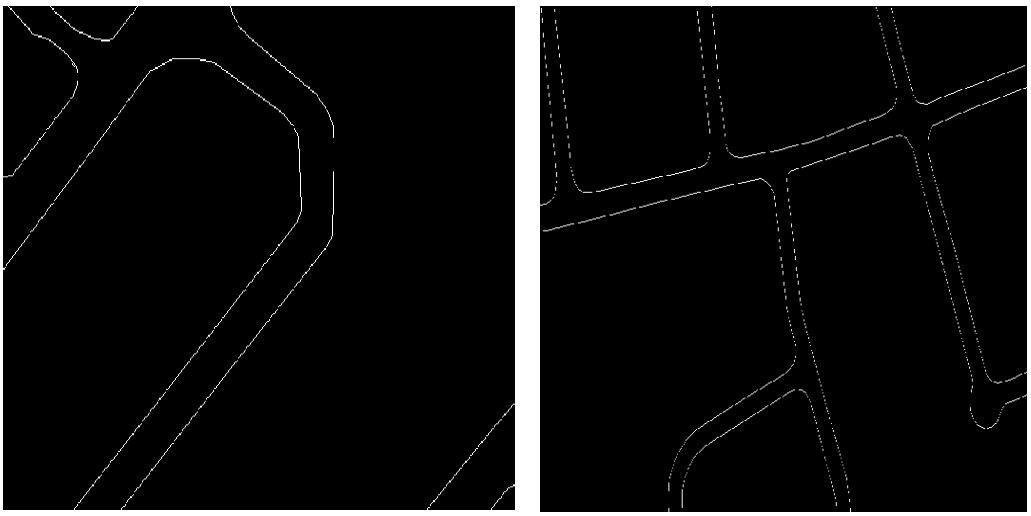
**(b) Ground Truth Images**



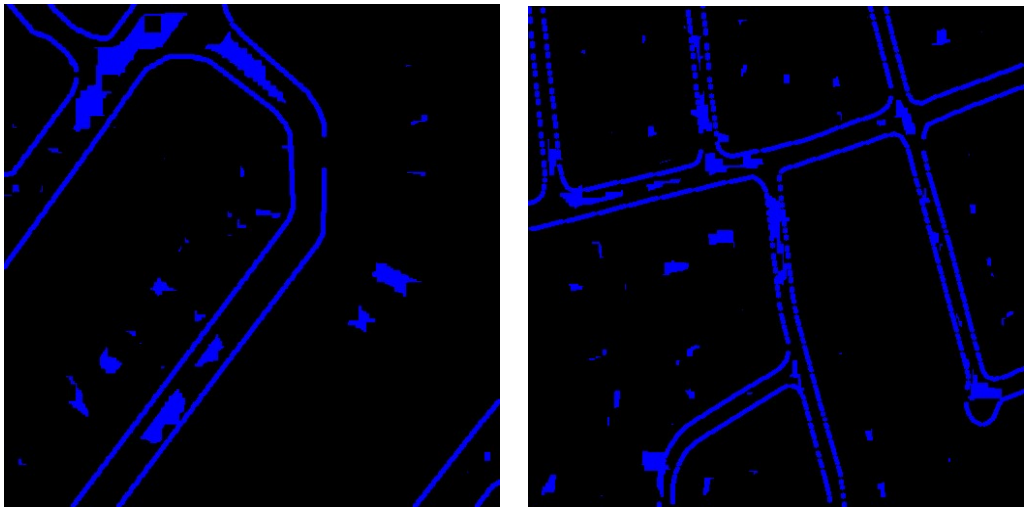
**(c) Predicted Images**

**Figure 4.6: Experimental Outcomes of Surface Detection using GDSLO Method**

**(a) Original Images**



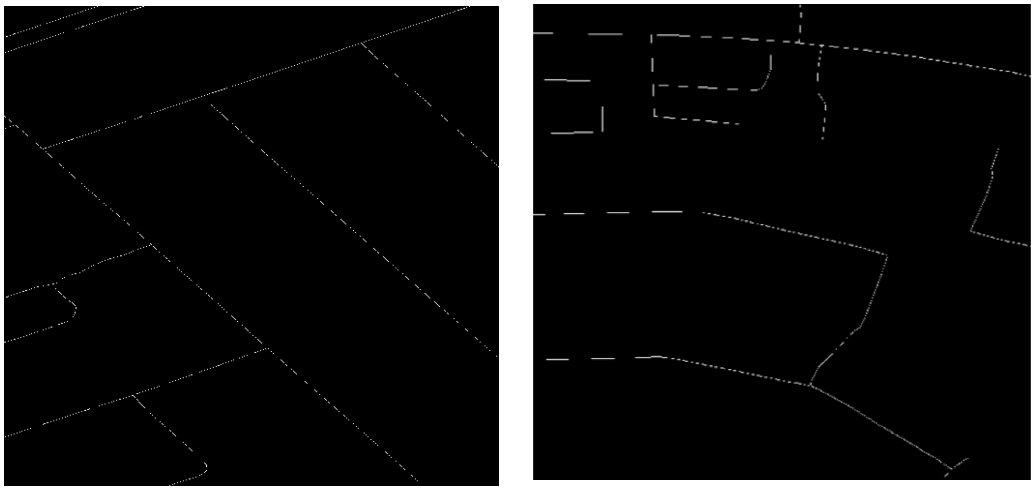
**(b) Ground Truth Images**



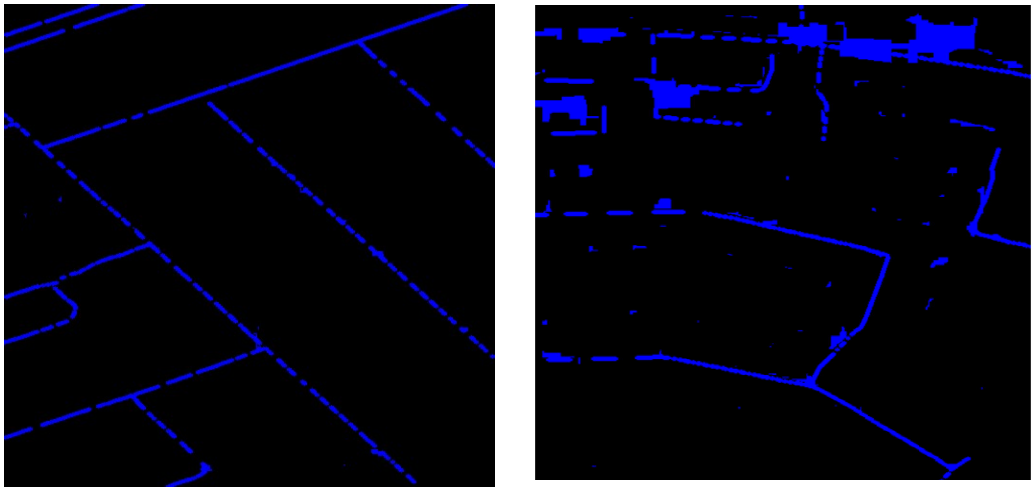
**(c) Predicted Images**

**Figure 4.7: Experimental Outcomes of Edge Detection using GDSLO Method**

**(a) Original Images**



**(b) Ground Truth Images**



**(c)Predicted images**

Figure 4.8: Experimental Outcomes of Centreline Detection using GDSLO Method