Chapter 3
Modified U-Net

3.1 Introduction

For image classification, object detection, and semantic segmentation, convolutional neural networks (CNNs) are beneficial. CNN has had tremendous success in a variety of fields(Jiang, 2019)(Barbedo, 2018)(S. Yadav1, 2019) . In 2012, Alex Net won the Image Net competition (A. Krizhevsky, 2012). The Alex Net inspired the development of a number of CNN-based networks, including inception-v3, GoogleNet, SegNet, VGG, U-Net and ResNet(C Szegedy, 2014)(Christian S, 2015)(K Xiangyu, 2015) . CNN is the foundation of many well-known networks such as U-Net. Because of ability of extraction of features this network is widely used deep learning based methods. In this chapter initially U-Net architecture is used to do semantics segmentation from HR RS images for Road Network. As this architecture takes more time to detect the image and train the method on the dataset. Therefore a novel modified U-Net approach is proposed that has lesser number of CNN layers that cause the faster train the model on the dataset as well as fast extract the images from HR RS images. The organization of the chapter 3 is as follows. Section 1 covers the proposed Modified U-Net method , section 2 describe the results and discussion of the proposed method, section 3 represent the experimental setup and finally section 4 include the applicability analysis of the modified U-Net with other well known existing methods.

3.2 Proposed Modified U-Net based Road Network Extraction System

As per literatures surveys, the deep learning based network extract higher level image characteristics when the numbers of the convolutions layers are high while simple image features are extracted by the lesser number of convolutions layer. In this thesis, detection of the road network from the satellite images required the simple two class data such as road and non road. This is possible with binary classification methods to

extract road features from the background (RS image). So it is not required deeper neural network architecture for semantic segmentation road network model. This will reduce the computational resources consumed by proposed Modified U-Net(Miral Patel, 2022). In this method simple VGG network was used as the baseline.

3.2.1 Block Diagram of the Modified U-Net

Figure 3.1 shows a block diagram for the proposed method for extracting road networks. The deep learning based road model need the higher number of the training samples It is hard to acquire the satellite and aerial image for train the deep learning based model. Therefore limited number of the input image is converted in higher amount by Data Augmentation method. These images are used to train the road surface detection model of the Modified U-Net. The accuracy of the model is measured by the various performance matrices such as IOU, DICE score, training time and testing time. However, due to limited number of the training samples available, it is required to optimize the number of the training samples, validation samples and testing samples. For this divide the total number remote sensing images into training, validation and testing splitting of the train%, val% and test%. For example 90% images of the dataset are used for the training of the model, 5% of the images are used for validation of the proposed model and rest of the 5% images is needed for the testing of the proposed model.

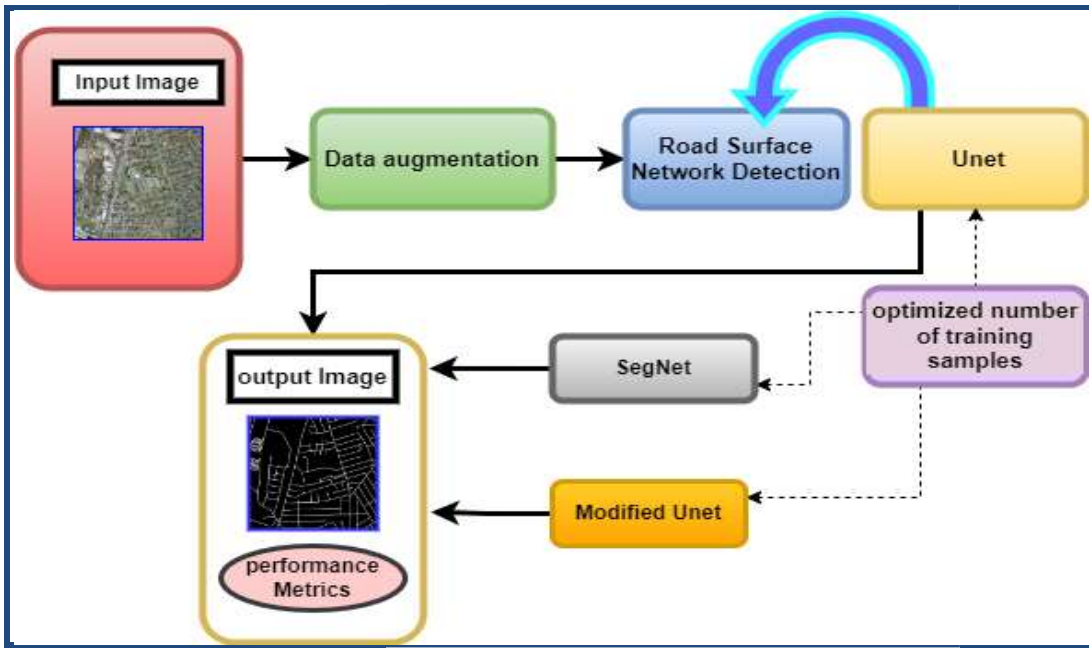


Figure 3.1: Proposed Road Network Extraction System Block Diagram

3.2.2 Architecture of U-Net

The U-Net architecture(C Szegedy, 2014) consists of two different paths, such as contracting path on the left side and expansive path on the right side. Here, the contracting path pursues the convolution network typical model. The appropriate data is gathered from the pre-processed output P_i through the contracting path, whereas the apt region to be segregated is localized with the expansive path. From the contracting path, the features with the higher-level values of pixels are generated which is then incorporated with the characteristic maps during the process of up sampling to restore the images. The entirely linked layers considerably minimized the required constraints to train the neural network with the minimum amount of data. Every stage of expansive path contains feature map up sampling, which is followed by up-convolution that halves the overall feature channels, thereby obtaining segmented output by transforming the feature map channels.

Here, the stimulation function accomplished in the Rectified Linear Unit (ReLU). Moreover, in each convolution, cropping procedure is significant because of the loss of border pixels. As a result, the segmented image result is signified as S_i . Originally it is proposed for the medical image segmentation purpose. The architecture shape looks like U so named as U-Net architecture. The architectural representation of U-Net model is shown in Figure 1. In this thesis, it is proposed for the road network extraction system from remote sensing images with modification in the architecture and named as Modified U-Net.

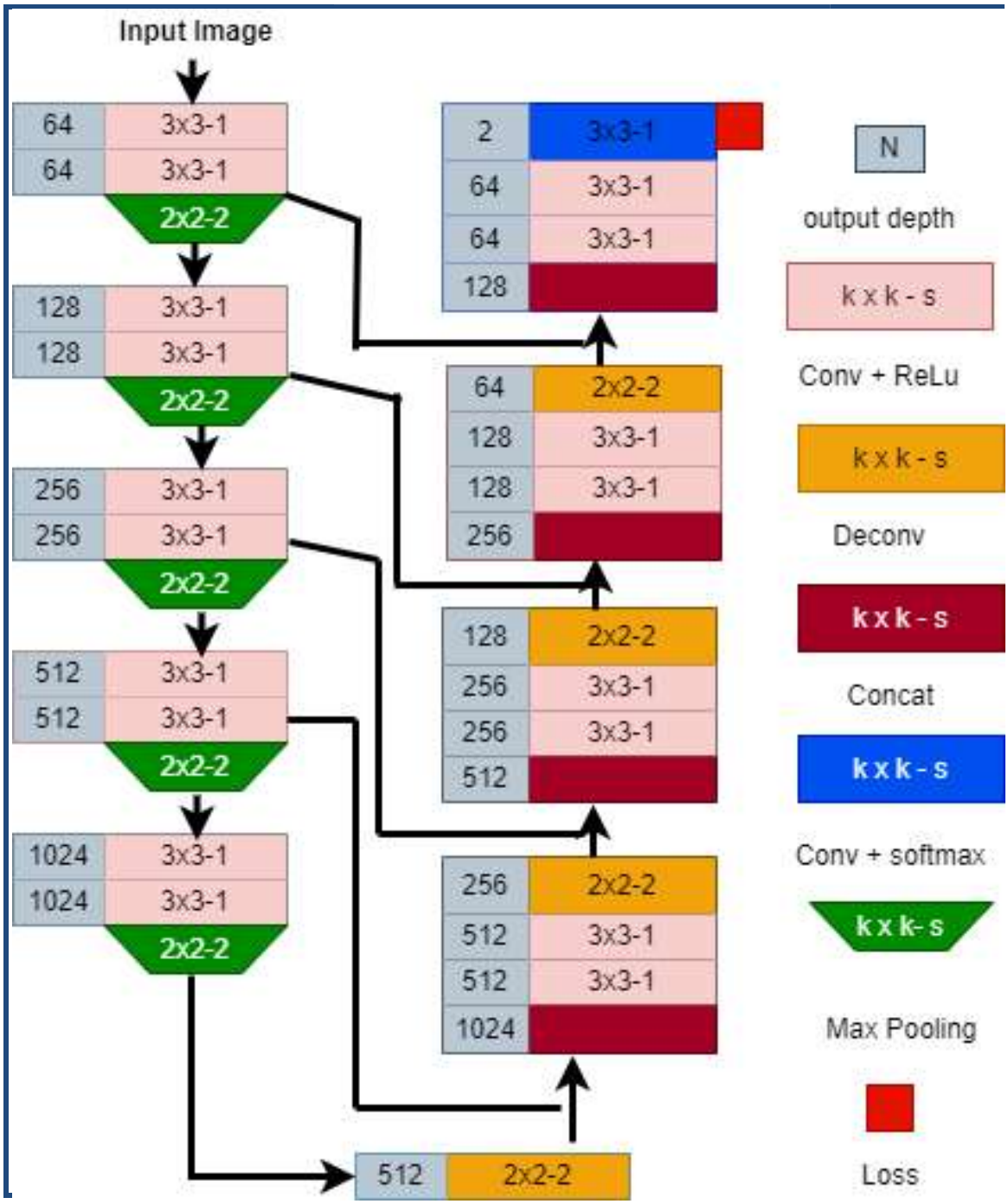


Figure 3.2: Architecture of U-Net

3.2.3 Modified U-Net Architecture

The U-Net perform very well in semantic segmentation tasks(Ronneberger O, 2015) . The U-Net network is extensively used in image segmentation for the two reasons: it is trained from end-to-end and performs well on a tiny dataset; and the quantity of train samples for U-Net is relatively small.

The detail architecture of each layer of the U-Net is described in Figure 2.2. It gives detailed information of the U-Net network at the encoding stage (contracting path) decoding stage(expansive path). In the contracting path, it has five convolution layer that makes total 10 convolution unit which has kxk kernel size and s stride followed by the ReLu activation function as indicated in the pink color in the Figure 2.2. At the encoding stage total five Max pooling layers, size of kxk filter and s stride are used as indicated in green color. The gray color box shows the depth of the output channel at each layer of the encoding stage and decoding stage of the U-Net architecture.

The Decoder stage consists of the total 8 convolution block followed by the activation function of the ReLu. It has four De-convolution unit that de-convolve the input featured image with kxk filter and s stride. After De-convolution of the image, it produce the features image that have the half of the size than the previous stage as indicated in mustard yellow. Decoder stage has four concating unit as shows in burgundy color. Last block of the decoder stage is the convolution block with softmax activation function that is used to classify the feature vector in the probability of the particular class as indicated in the blue in the color and red box indicated the loss function used to calculate the error between the target class and predicated class of the segmented output image.

However, the U-Net has deeper structure as it has to detect complicated features of the input image. But the proposed methods have to identify the road element from the aerial or satellite image that will be possible with binary segmentation task only. So it is not required the deeper architecture that need more computational memory. Moreover the U-Net model has lots of the learning parameter that need higher amount of the training time and less speed for the detection of the image.

In the modified U-Net, both the encoding stage and decoding stage of each layer is modified. The four up sample parts of the proposed architecture were reduced to three and the five down samples parts of the proposed architecture were reduced two as shown in the Figure 3.3. So, At the encoding stage of the modified U-Net consist of the seven convolution unit and three max pooling instead of the ten convolution unit and five max pooling used in the traditional U-Net architecture. The decoding stage

of the modified U-Net comprise of the three units. Each Unit has one concat unit, two repetitive convolution units and one de-convolution unit as presented in the Figure 3.3. However ,The decoder stage of the U-Net architecture for the road network detection system is consist of the six convolution function , three concat function and two de-convolution unit. Therefore, overall 13 convolution block with ReLu activation function, three max pooling, three concat block, three De-Convolution block and one convolution block with softmax activation function are required to detect the road network by Modified U-Net.

The final step of the decoder network has a softmax layer attached to it that transforms the output into probability maps. Cross entropy loss, which is used in this model to train the road detecting network, is defined as

$$L_{bce} = -\sum_{i=1}^{b}\sum_{j=1}^{n} GT_{ij}\log(predi_{ij}) + (1 - GT_{ij})\log(1 - predi_{ij}) \quad (3.1)$$

Where GT_{ij} is ground truth pixel of ith batch of jth pixel of the image and $predi_{ij}$ is predicted output pixel of ith batch of jth pixel of the image; b is batch size and n= no of pixel in the images. The sigmoid function is applied to the weighted sums of the hidden layer activations $predi_i$, to generate the outputs of modified U-Net model,

$$predi_i = \frac{1}{1+e^{-\theta_i}}, \quad (3.2)$$

$$\theta_i = \sum_{j=1} w_{ij}h_j \quad (3.3)$$

Using the chain rule, we can calculate the error's derivative with regard to each weight connecting the hidden and output units,

$$\frac{\partial L_{bce}}{\partial w_{ij}} = \frac{\partial L_{bce}}{\partial predi_{ij}}\frac{\partial pred_{ij}}{\partial \theta_{ij}}\frac{\partial \theta_{ij}}{\partial w_{ij}}. \quad (3.4)$$

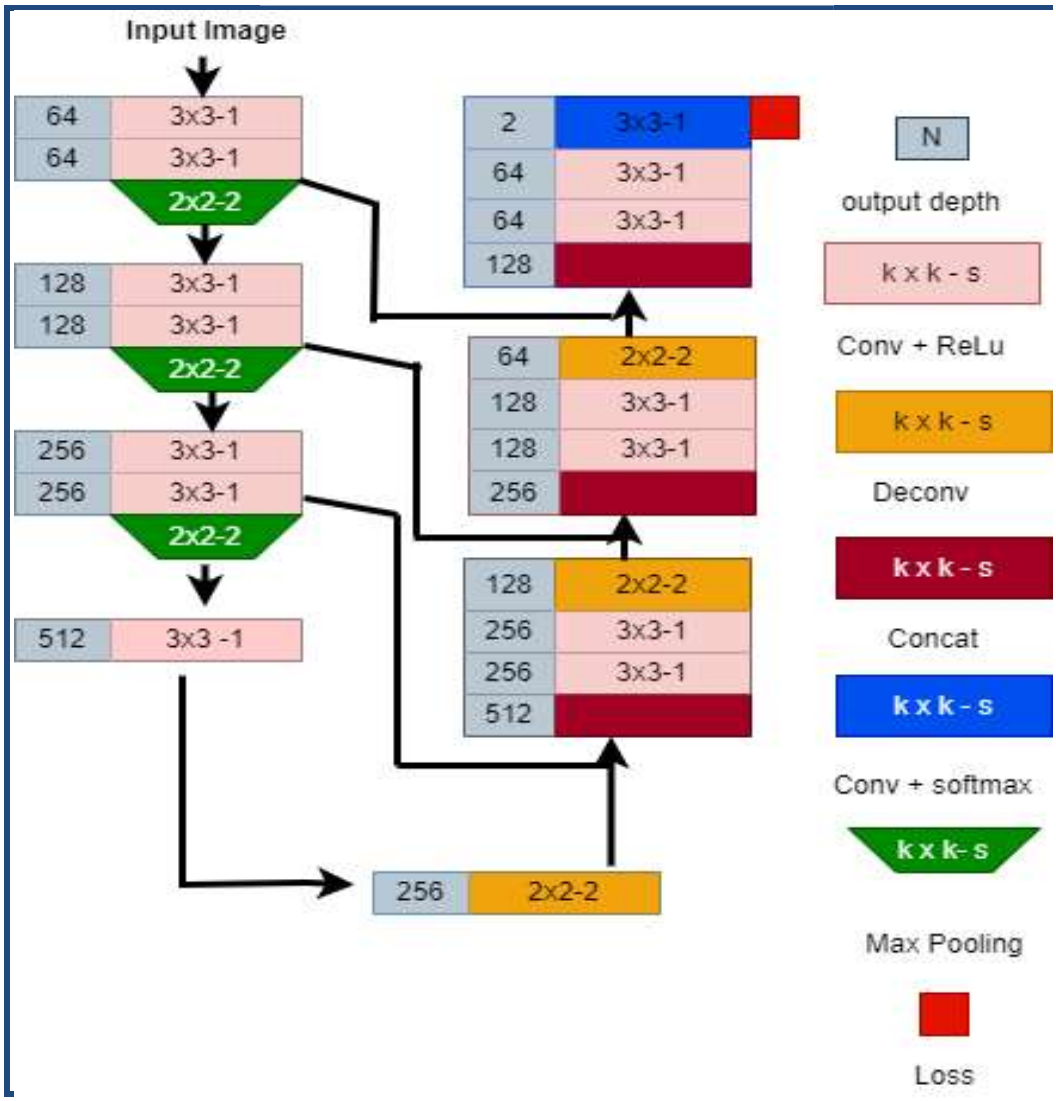


Figure 3.3: Architecture of Modified U-Net

3.2.4 SegNet Architecture

SegNet architecture used for semantic segmentation for the image that have encoder network and decoder network followed by pixel wise classification. The detailed architecture of the SegNet with convolutional block, up sampling, max polling, output depth of the each layer and the loss function described in the Figure 3.4. Each block in the encoder stage performs the convolution between the filter bank and the set of the feature map of the image. Then after it output fed into the non linear activation function of the ReLu. Following that, Max Pooling with kernel size of 2x2 and stride is two is performed and the resultant feature map is sub sampling by two. The reflected structure of the decoder in the decoder network up-samples its input feature

map(s) using the memorized max pooling indices from the corresponding encoder feature map(s). This step produces sparse feature map(s).

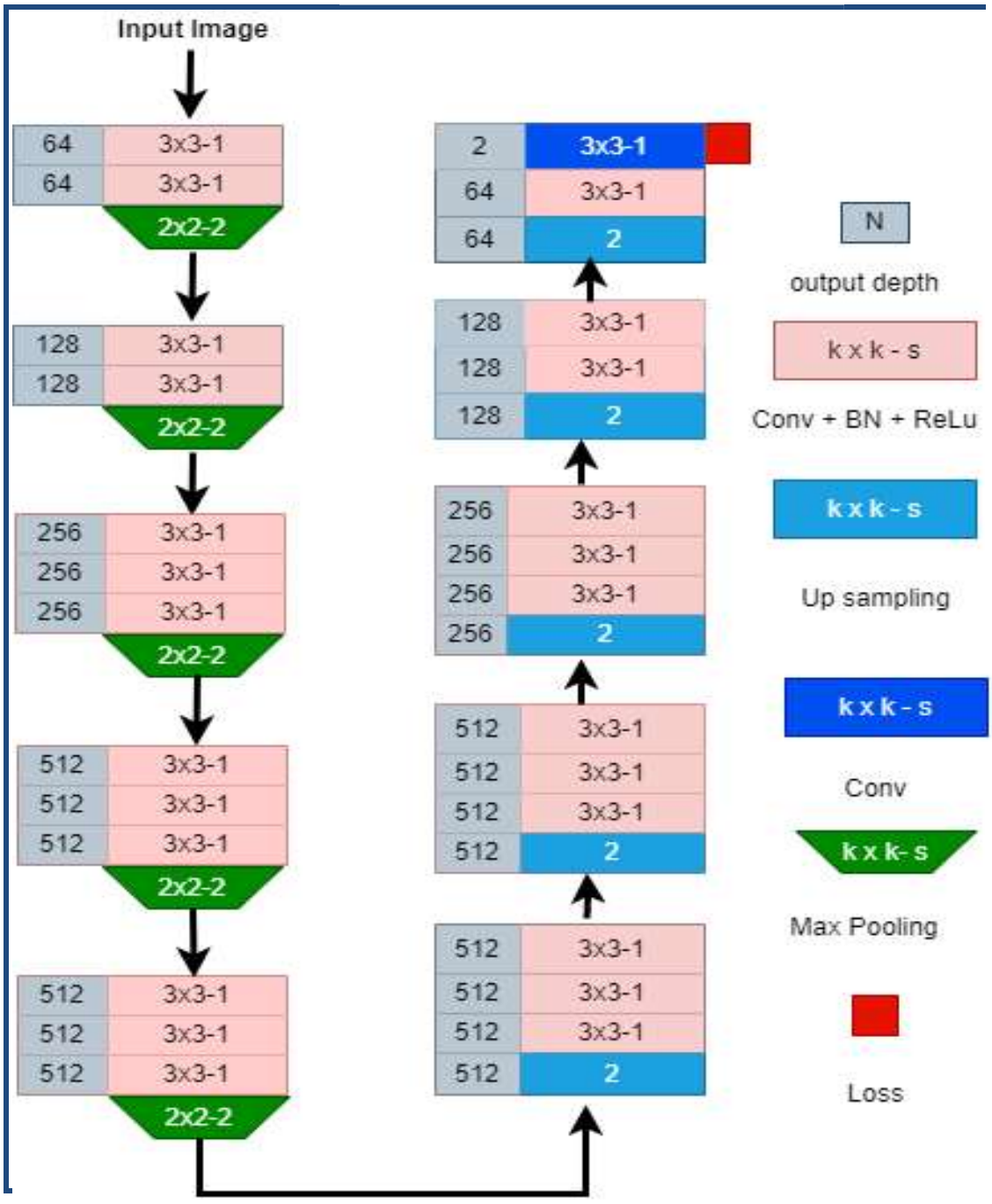


Figure 3.4: Network Structure of SegNet

Therefore overall encoder section has total 13 convolutional layers based on the VGG16 Network(I. Goodfellow, 2016)(V. Badrinarayanan, 2015)Each encoder layer has corresponding the decoder layer so decoder layer has also 13 convolutional layers.

The final layer is consisting of the multi class soft max classifier that classifies the each pixel of the previous layer of the decoder stage output feature image.

3.3 Experimental Setup

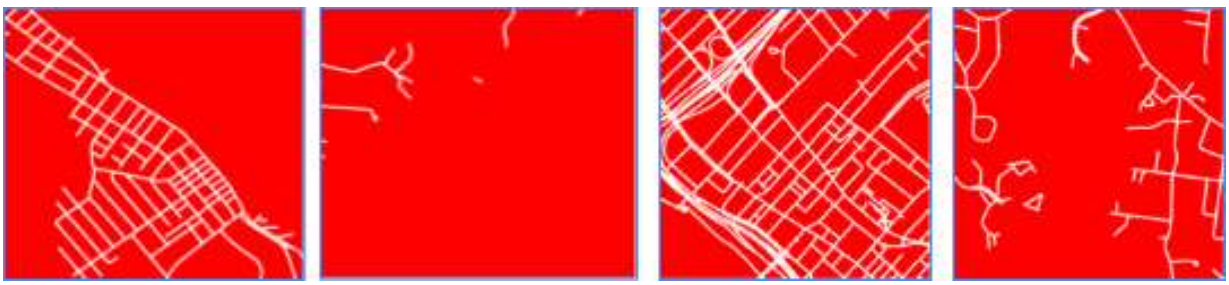
Any deep learning based system has required mainly high processing GPU. For implementation of method used PYTHON tool with Pytorch using PC having 8GB RAM windows 10 OS, and Intel i3 core processor.

3.3.1 Dataset

We choose publically available of Massachusetts Roads Dataset consists of 1634 aerial images of the state of Massachusetts(Massachusetts dataset) . Each image is 1500×1500 pixels in size, covering an area of 2.25 square kilometers. Initially dataset has training set of 1208images, a validation set of 212 images and a test set of 214 images. Figure 3.5 represented some samples of remote sensing images and Figure 3.6 shows ground truth of this sample images.

(a) Image 1 (b) Image 2 (c) Image 3 (d) Image 4

Figure 3.5: Remote Sensing Images



(a) Image 1 (b) Image 2 (c) Image 3 (d) Image 4

Figure 3.6: Ground Truth Images of figure 3.5

3.3.2 Data Augmentation

Initially, the U-Net model has been trained by the whole original Massachusetts road dataset. Due to the huge amount of images and big image size needs high training time to train the road segmentation model. So the training sample reduces by half and now it is a total of 604 images. These total images are divided into a training set, validation set, and testing set. Training and testing samples of the dataset are divided into various percentages for evaluation of the road network segmentation model performance as shown in Table 3.1. Due to less number of training samples used by the model, the data augmentation techniques are used to increase the number of sampled images. The data augmentation images is done using different techniques such as flipping, cropping, and rotation suggested by (Shorten, 2019).

Training – Testing split	No of Training images	No of Validation images	No of Testing images
90%-5%	543	30	31
80%-10%	483	60	61
70%-15%	422	91	91
60%-20%	**362**	**121**	**121**
50%-25%	302	151	151
40%-30%	241	182	181
30%-35%	181	212	211

Table 3.1: Training -Testing Split

3.3.3 Performance Metrics

In this study, four quantitative criteria were used to evaluate the segmentation results. The overall pixel accuracy (Acc), Intersection over Union (IoU), Dice score, and road accuracy were used to assess and compare the segmentation performance (eqs. (3.5) – (3.10)). These parameters are averaged for all the training samples. The Acc, Dice, road accuracy, and IoU were averaged over all images in the testing set. Moreover, the testing time was also measured to assess the segmentation speed of single image and training time was also measured for how long time is required to train the module.

$$\text{iou} = \frac{\text{intersection}}{\text{union}} \tag{3.5}$$

$$\text{iou} = \frac{\sum(\text{predi})(\text{GT})}{\sum \text{pre} \quad \sum \text{GT} - \sum(\text{predi})(\text{GT})} \ 100\% \tag{3.6}$$

$$\text{dicescore} = \frac{2 * \text{intersection}}{\text{union} + \text{interse}} \tag{3.7}$$

$$\text{dicescore} = 2 * \frac{\sum(\text{predi})(\text{GT})}{\sum \text{predi} + \text{GT}} \ 100\% \tag{3.8}$$

$$\text{roadaccuracy} = \frac{\sum \text{TP}}{\sum \text{TP} + \sum \text{FN}} \ 100\% \tag{3.9}$$

$$\text{overallaccuracy} = \frac{\sum \text{TP} + \sum \text{TN}}{\sum \text{TP} + \sum \text{TN} + \sum \text{FP} + \sum \text{FN}} 100\% \tag{3.10}$$

where, TP is true positive;

TN is true negative;

FP is false positive;

FN is false negative.

3.3.4 Road Segmentation Network Training

The study uncovers appropriate hyper-parameter combinations for use with land cover classification issues utilizing multispectral pictures and optimizes convolution neural network architecture (K Xiangyu, 2015) .

Figure 3.7 shows the algorithm for the road surface detection from remote sensing images. The network was optimized using the Adam optimizer. A learning rate of 1×10^{-3} was used. It was important to lower the learning rate because if a bigger learning rate was specified, the weights combined would diverge from the ideal answer. For

binary segmentation, the loss function used cross-entropy loss. The training batch size was set to 4, the number of hidden layers set to 64, and the number of epochs to 100. The loss value, accuracy, dice score, and IOU of the training set and validation set were noted during the training phase.

Algorithm 1: Road Network Extraction System

Input : Training Set = $\{(Y_1,V_1,X_1\},.....,(Y_N,V_s,X_n)\}$
Where Y_i=1,.....,N (Training),V_i=1,.....,S (Validation),
X_i=1,.....,n (Testing)
$Y_N=Y_N^+$
Initialize : Initial Learning Rate α, Batch Size , Epochs T,
Initial Loss function L(v, θ,t)

Initialize: $v_1,\theta_1=0$, $\beta=0.99$
Build Unet()
if torch.no_grad() ==true; (Training phase)
start timer():
for t=1 to T **do** (update parameters)

$$g_t = \nabla\theta f_t(\theta_t - 1)$$
$$v_t = \beta(v(t-1) + (1-\beta)g_t^2$$
$$\theta_t = \theta(t-1) - \alpha(\hat{m}_t/\sqrt{\hat{v}_t} + \varepsilon)$$

end for
output: trained θt (resulting parameter)
calculate iou, dice score, accuracy, loss
if torch.no_grad() != true (validation phase)
calculate iou, dice score, accuracy, loss
Plot T →·accuracy (for training and validation)
Plot T →·Loss (for training)
stop timer():
if torch.no_grad()!= true (testing phase)
calculate iou, dice score, accuracy;
import confusion matrix;
visualize the result

Figure 3.7: Modified U-Net Training Algorithm

3.4 Results and Discussion

3.4.1 Result Analysis based on Number of Training Samples and Training Time

Initially whole Massachusetts publically available dataset used to train the basic U-Net module for the extraction of the road network from the remote sensing images. It has total 1208 training samples to train the model. It needs significant training time to train the basic U-Net model. It is very lengthy, tedious and time consuming task to train the model from the whole dataset.

Training – Testing split	No of the Training images	No of the Validation images	No of the Testing images	training time(sec)	training time (Hr)
90%-5%	543	30	31	14010.86	3.89
80%-10%	483	60	61	13877.64	3.85
70%-15%	422	91	91	11748.96	3.26
60%-20%	**362**	**121**	**121**	**5151.07**	**1.43**
50%-25%	302	151	151	4892.18	1.35
40%-30%	241	182	181	4800.2	1.33
30%-35%	181	212	211	4300.3	1.19

Table 3.2: Training Time of Various Dataset Samples Splitting

Therefore, it is selected the number of the sample for newly reduced dataset that covers all type of the complex road scenario in the remote sensing images. It covers the long road under the effect of the occlusion, tall buildings and sharp turn. Initially, the numbers of the remote sensing samples are reduced by 50% to train the U-Net model for road network detection. As a result, 604 training samples are chosen from the Massachusetts road dataset. These images have a complex background. The training, validation, and testing sets were separated from these training samples. In addition, we sought to minimize the model's memory and training requirements by determining the optimal amount of training samples. As a result, the initially trained module employed 90 % of samples for training, 5 % for validation, and the remaining 5 % for testing from 604 images. Same as training samples are 80%, validations samples are 10% and testing samples are also 10% and so on. It changed the training sample count from 90 % to 30 % successively.

The model performance evaluated in terms of training time and testing time at different training and testing image splitting as shown in table 3.2. The model takes training time 3.89 and 1.4 hours for 90% and 60% training samples respectively. The training time continuously decreased as number of the training samples decreased.

Training time required by the basic U-Net model for the different dataset splitting is shown in the Figure 3.8. The graph indicated that for the 60%, 50%, 40% and 30% of the training samples used to train the road detection model is required the approximate same training time. Others training samples splitting required approximate double the training time and therefore optimal solution for number of the training samples necessary for train the road surface detection model would be 60% and lower .

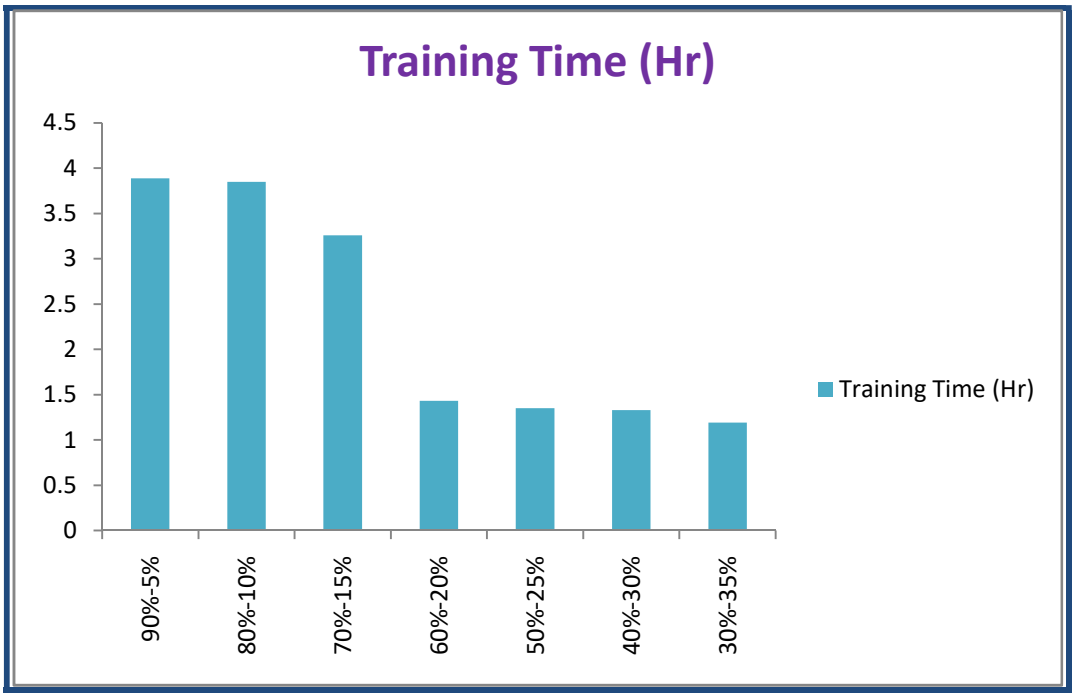


Figure 3.8: Training Time of Various Dataset Samples Splitting

For deep learning based methods, selection of the hyper-parameters value should be chosen carefully. In the proposed methods hyper-parameters are learning rate, epochs, hidden layers, batch sizes and optimizer. Initially learning rate value is selected 0.1 and other hyper-parameters values were fixed but it gave very poor performance for the selected dataset. Then after, gradually the value of the learning rate is decreased from 0.1 to 0.00001 and check the model performance by changing other parameters

too. When model is trained for the learning rate 0.0001, hidden layers is 64 and number of the epochs equal to 10, it gave very bed performance for any number of the training samples. Then, the epoch's values were changed from 20, 30 and 40 for the road surface detection model based on the basic U-Net. Here considered epoch 30 to find the optimum value of the training samples because higher value of epochs needed greater training time to train the model that was time consuming and laborious job. Therefore, the basic U-Net model performance compared the results to the value of epoch 30 along with changing the training-testing splitting.

Training – Testing split	**Training Phase**						
	Training				**Validation**		
	Loss	**IOU**	**DICE**	**Acc**	**IOU**	**DICE**	**Acc**
90%-5%	9.81	79.85	88.37	91.80	87.10	93.05	95.83
80%-10%	8.11	80.10	88.50	92.40	80.66	89.10	93.11
70%-15%	8.87	78.62	87.54	91.32	80.43	88.96	92.47
60%-20%	**7.60**	**79.25**	**87.97**	**90.20**	**79.85**	**88.55**	**90.71**
50%-25%	5.71	80.39	88.66	91.07	77.33	86.91	88.95
40%-30%	5.84	80.52	88.76	91.08	77.98	87.33	88.87
30%-35%	2.99	82.56	87.79	78.95	78.95	87.96	89.26

Table 3.3: Training Phase Parameters Comparison of epoch30

As shown in Table 3.2 the training time is approximate 3.89 hr for 90 % training samples. If the number of the training samples is gradually decreased, it observed that accuracy of the road surface model by basic U-Net is also decrease. The road detection model's performance is measured by loss, IOU, DICE score and Accuracy during the training phase and validation phase as shown in the Table 3.3.

The basic U-Net model for road detection is measured during the testing phase by the evaluate parameters such as IOU,DICE ,Overall Accuracy and Road Accuracy as presented in the Table3. 4. It shows overall accuracy values are 88.24% , 88.04%, and 97.27% for the training –testing splitting of 50%-25%,40%-30% and 30%-35% respectively. Similarly, Road accuracy values of the basic U-Net model are 64.61% , 0% and 0.12% for the training –testing splitting of 50%-25%,40%-30% and 30%-35% respectively. These indicated that from the background road parameters are poorly detected. The overall accuracy value for the 90%-5% training-testing splitting

is 95.94% but model needs very large amount of the training time. Moreover also it has more number of the learnable parameters to train the model. It causes the model heavy that occupied more CPU resources. However, there are the two options to choose the number of the training samples as 70% and 60%.

Training – Testing split	Testing			
	IOU	DICE	Overall Accuracy	Road Accuracy
90%-5%	88.59	93.92	95.94	79.27
80%-10%	88.79	94.03	96.19	82.18
70%-15%	85.67	92.12	94.09	80.72
60%-20%	**81.87**	**89.68**	**91.50**	**82.93**
50%-25%	79.71	88.24	89.54	64.61
40%-30%	79.39	88.04	89.53	0
30%-35%	79.20	87.27	88.72	0.12

Table 3.4: Testing Phase Parameters Comparison when epoch30

Table 3.5 gives the data about approximate same training time of the basic U-Net model. In this, number of epochs were set in such a way that given the approximate similar training time of the model for the various number of the training samples. 70% of the training samples required the 9321.23 sec to train the module at 23 epochs and it has 68.32 % of road accuracy. While 60 of the training samples required the 9488.41 sec training time and number of the epochs were needed 60. It generate the road accuracy of 91.66% that is more than 20% better than the 70% training samples are used to train the model. This way optimum number of training samples are identified and later on overall training samples are divided : 60% for training purpose, 20% validation and rest of 20% samples are used for testing.

Training – Testing split	Training Time (sec)	Number of Epochs	Testing Phase	
			Overall Accuracy (%)	Road Accuracy (%)
90%-5%	9941.59	20	93.24	70.87
80%-10%	9425.51	22	92.18	69.84
70%-15%	9321.23	23	91.20	68.32
60%-20%	9488.41	60	94.16	91.66
50%-25%	9152.78	62	93.40	87.23
40%-30%	9234.07	70	93.20	74.17
30%-35%	9267.67	71	93.18	70.19

Table 3.5: Approximate the Same Training Time for Different Epoch

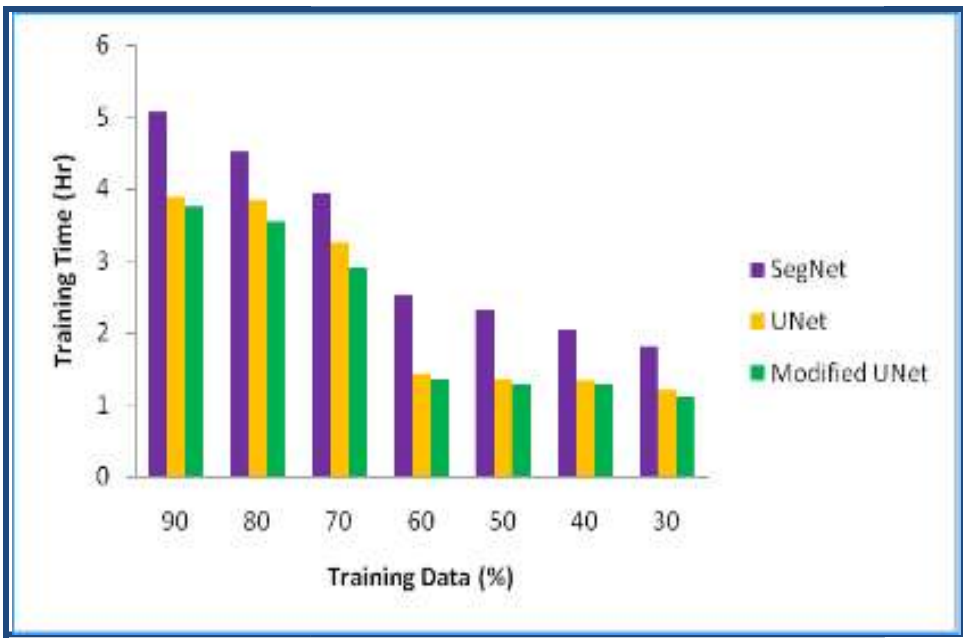
3.4.2 Analysis of Training Time and Testing Time of modified U-Net

With the rapid growth of computer hardware in recent years, high-end GPU or GPU clusters have made network training easier. However, given the concern of cost-effectiveness in training time and commercial cost, trade-offs between layer depth, the number of channels, kernel sizes, and other network attributes must still be considered when designing network architectures(He & Sun, 2015) for experimental research and practical application.

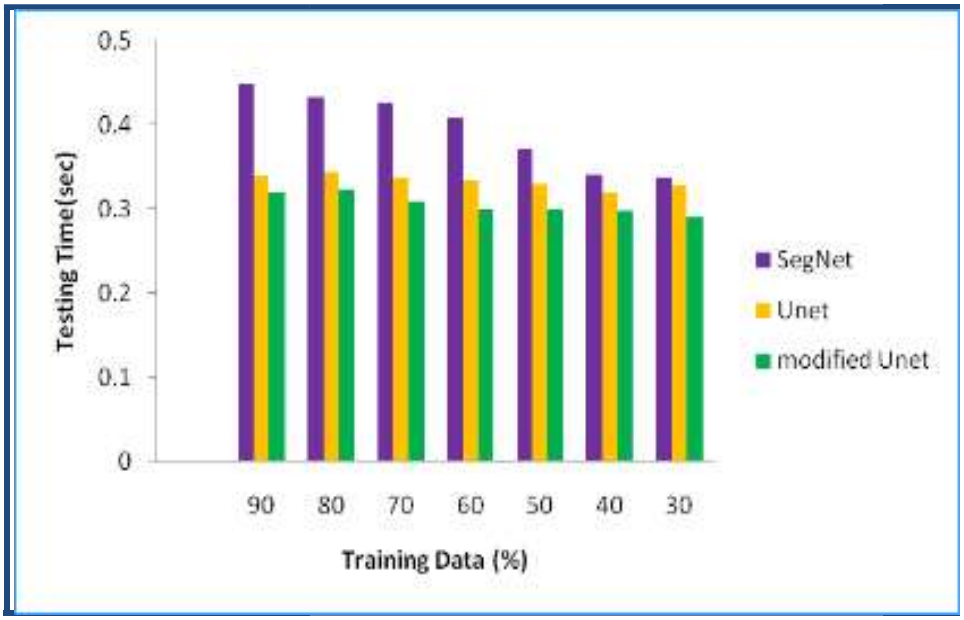
The various evaluation parameters, training time, and inference time were compared to other state-of-the-art deep learning algorithms such as SegNet, and U-Net about various numbers of training and testing samples. The SegNet architecture was used as indicated in Figure 3.5(V. Badrinarayanan, 2015). It's worth noting that numerous factors, including parameters and model structure, can also affect the running time of deep models, including training and testing time(Canziani, 2016).

Figure 3.9 shows SegNet has the longest training for all training –testing distribution of any other model. This is due to the deeper structure of SegNet with more Convolutional layers increases the complexity of the model as well as the number of parameters. Moreover, SegNet has the highest inference times compared to any model

for all possible training data. The modified U-Net has training and testing time is shorter than SegNet and basic U-Net. This is due to the number of convolutional layers being less in the Modified U-Net compared to the traditional U-Net. Hence less number of learning parameters is required to train the module. Therefore, the least training time and testing time was noted for modified U-Net compared to other state of art methods.



(a) – Training Time



(b) – Testing time

Figure 3.9: Comparison with other State of Art Methods

3.5 Applicability Analysis of Modified U-Net

Model Name	Training			Testing			
	DICE (%)	IOU (%)	Overall Acc (%)	DICE (%)	IOU (%)	Overall Acc (%)	Road Accuracy (%)
SegNet	93.9	88.74	92.62	91.59	85.18	92.3	91.7
UNet	94.6	93.71	95.1	91.2	87.4	94.2	92.86
Proposed Unet	93.74	93.28	94.26	92.68	92.19	93.48	93.3

Table 3.6: Evaluation of Segmentation Result with Well Known Methods

The SegNet and U-Net were constructed by the convolutional neural network. Their semantics segment accuracy was higher compared to other deep learning based model. SegNet obtained the segmentation result by step-by-step up sampling and convolutional layers. Basically it needs higher number of training samples. The numbers of training samples were relatively used small in this road surface segmentation from the remote sensing images. Therefore, the performance of this algorithm was not good. Moreover, the optimizer used in SegNet method was SGD which also one of the reason of poor performance of SegNet. To obtain same accuracy, IOU and DICE score, SegNet required more number of epochs to train the modeled with lesser number of training samples. So this would be cause of larger training time and testing time.

U-Net recovered the details of the result step by step up sampling and merging the features layers from the backbone. Therefore, U-Net could segment the details better from the remote sensing images. The target of this image segmentation task was to extract road part from the remote sensing images. The complexity of the image segmentation task was low. So, the encoding part of the U-Net network was simplified according to the characteristics of the low difficulty of the segmentation task.

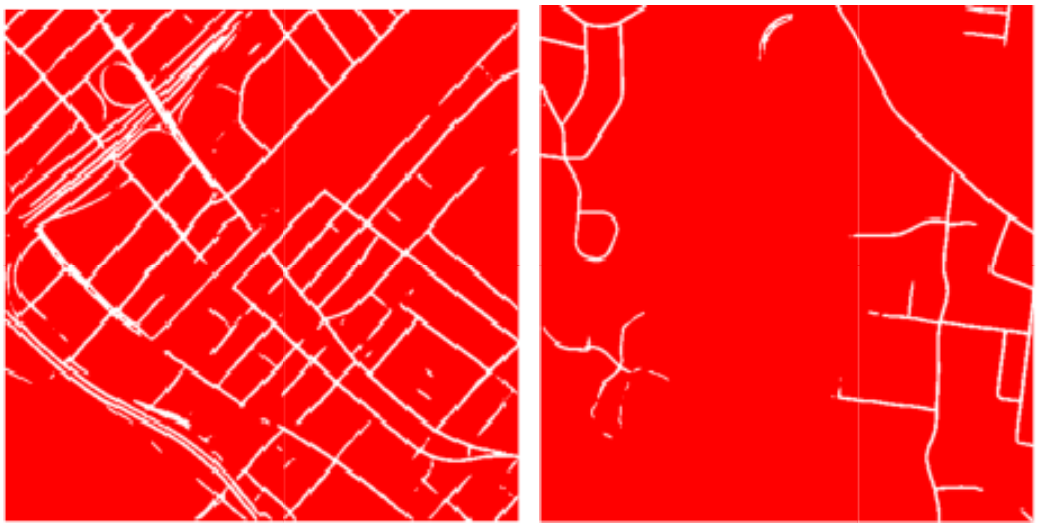
The last two convolution units (including two convolution layers and one pooling layer) of the VGG network were removed, leaving only the first three convolution unit and the two convolution layers in the decoding unit (including one up sampling

layer, one concatenate layer, and two convolution layers) were changed into one layer convolution layer. This way the redundant part of the basic U-Net model was removed. Therefore, the simplified network achieved the best segmentation result while the training and testing time were relatively small. The IOU, DICE score and overall accuracy on the training set were 93.28 %, 93.74 % and 94.26 % respectively during the training phase of the modified U-Net. However, the IOU, DICE score, Overall Accuracy and Road accuracy were occurred during the testing phase were 92.19 %, 92.68 %, 93.48 % and 93.3 % respectively in Table 3.6.

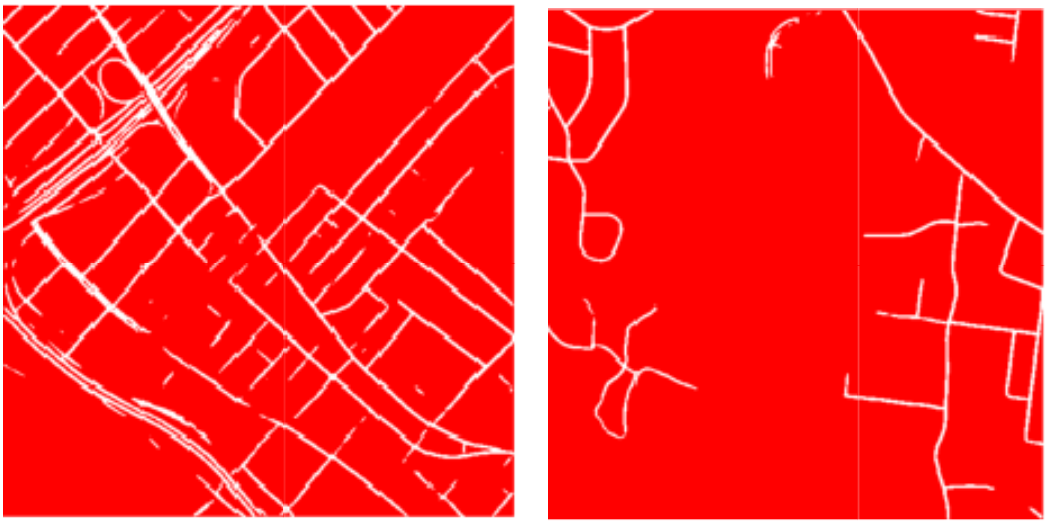
(a) Original Remote Sensing Images

(b) Ground Truth

(c) SegNet Output Images



(d) Basic U-Net Output Images



(e) Modified U-Net Output Images

Figure 3.10: Visual Comparison of Segmentation Methods

Figure 3.10a and Figure .10b represent original and its ground truth images respectively. Figure 3.10c, Figure .10d and Figure 3.10e shows SegNet, basic U-Net and modified U-Net detected outputs. The proposed modified U-Net is extracting equally good road network as basic U-Net with lower training and testing time.