Segmentation of satellite imagery

JIMUT BAHAN PAL

Term Project - 2020
Ramakrishna Mission Vivekananda Educational and Research Institute
June 2021

Advisor: Tamal Maharaj

Segmentation of satellite imagery

By

Jimut Bahan Pal

B1930050

the thesis is submitted to department of computer science
in fulfilment of the requirements for the degree of
master of science

Under the guidance of

Tamal Mj

Ramakrishna Mission Vivekananda Educational and Research Institute
Howrah - 711202

May, 2021

Acknowledgements

It is ritual that scholars express their gratitude to their supervisors. This acknowledgement is very special to me to express my deepest sense of gratitude and pay respect to my supervisor, **Tamal Mj**, Department of Computer Science, for his constant encouragement, guidance, supervision, and support throughout the completion of my project. His close scrutiny, constructive criticism, and intellectual insight have immensely helped me in every stage of my work. I would like to thank him for patiently answering my often-naive questions related to Deep Learning in Computer Vision.

I would like to thank all the professors of Department of Computer Science for creating an intellectually stimulating environment that enabled me to think more carefully and methodically about my research than I had ever done before. I would like to thank Swathy Prabhu Mj, Dripta Mj for the stimulating discussions that he had shared with me during the Advanced Machine Learning course.

I'm grateful to my father, Dr. Jadab Kumar Pal, Deputy Chief Executive, Indian Statistical Institute, Kolkata for constantly motivating and supporting me to develop this documentation along with the proofreading. I will also mention about my brother Jisnoo Dev Pal and my mother Sumita Pal, for supporting me.

Declaration

I hereby certify that the submitted work is my own work, was completed while registered as a candidate for the degree stated on the Title Page, and I have not obtained a degree elsewhere on the basis of the research presented in this submitted work. Where collaborative research was involved, every effort has been made to indicate this clearly.

Jimut Bahan Pal, B.Sc.(Hons)

jpal.cs@gm.rkmvu.ac.in

May, 2021

Certification

This thesis titled, **Segmentation of satellite imagery**, submitted by **Jimut Bahan Pal** as mentioned below has been accepted as satisfactory in partial fulfillment of the requirements for the degree M.Sc. in Computer Science in May, 2021.

Supervisor:

Tamal Mj
PhD, University at Buffalo, Buffalo, NY, USA
Department of Computer Science
Ramakrishna Mission Vivekananda Educational and Research Institute.

Abstract

Image segmentation is ubiquitous in daily life and is used almost everywhere. It has become concerning to build models which can segment image data with very less computational resources. This project has investigated one such new model from the UNet family, and build a powerful model using the available tools which outperforms the state of the art significantly in a lot of popular datasets. The model can not only be used for segmentation of satellite imagery, but also medical imagery.

Dedicated to my parents Dr. Jadab Kr. Pal and Sumita Pal.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 What is image segmentation ?

Image segmentation deals with partitioning a digital image into a set of image objects. It specifically deals with assigning labels to each pixel of image which share some common [1] characteristics. With the help of image segmentation, it makes humans and machines easier to analyze certain images. It also helps in object localization, for which one may find certain object/ region of interest from an image with too many objects.

Image segmentation are used in a variety of sectors, like surveillance where it is used to track and focus on people's images from video feed. One of the most important sectors is in hospitals where it is used for medical image segmentation to get knowledge about particular diseases present in images of patient which makes the clinical process automated. Segmentation is also used in detection to detect certain objects in an image, making the image easier to understand, it is used for visually impaired people to guide them in their day-to-day tasks. Segmentation is also used in image retrieval by famous search engines such as Google, for retrieving similar images. In robotics, segmentation is used for precise location of objects to perform certain tasks by robots, resulting in automation in factories, helping humans in redundant tasks. Earlier segmentation tasks were using in matching fingerprints for security and locating criminals by getting cues from crime scenes. The most important area is autonomous driving, it is used for detecting pathways, obstacles and planning how a car will move in wild. Some of the examples are shown in Figure

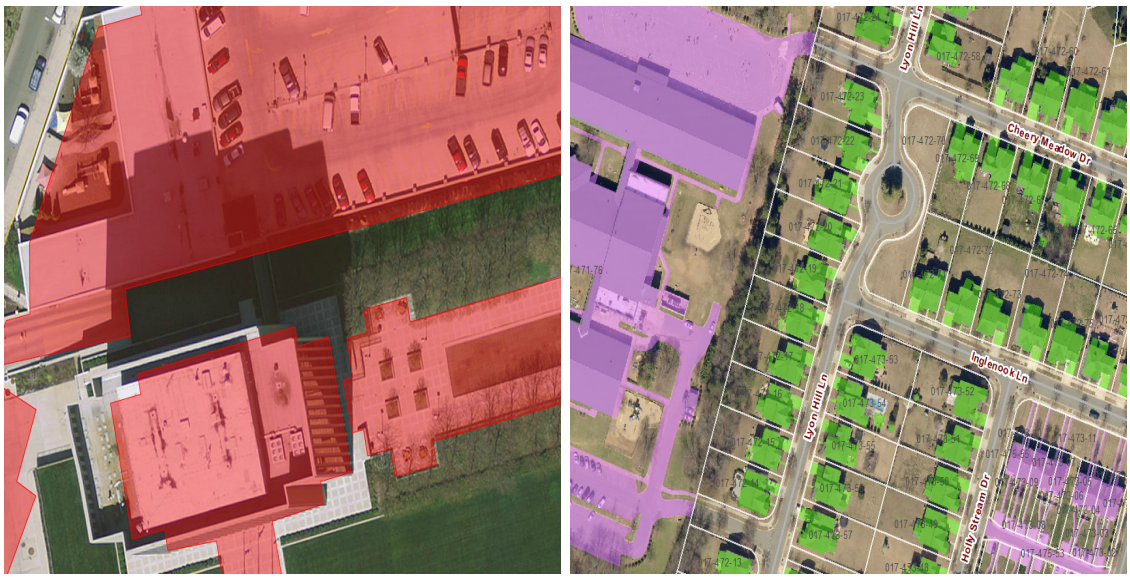


Figure 1.1: Examples of segmented satellite imagery. Left: P.C.: eijournal, Right: P.C.: mecknc.gov

1.2.

1.2 What is Segmentation of Satellite Imagery?

Segmentation of satellite imagery deals with dividing an aerial satellite imagery into different parts for better analysis. One may use the segmentation masks generated by roads to better analyze different positions of the maps. One could count the number of buildings in a specified area of the map to get better knowledge about the population density. Some may check the natural habitat ratio by counting the number of trees and ponds/ waterbodies present in the map. The automatic analysis can help investigators for better analysis of large amount of data where manual go-through is not possible.

1.3 Why do we need Segmentation of Satellite Imagery?

There are various factors which promote the need for segmentation of satellite imagery. Firstly, we can get overview of settlement content from a plethora of data where manual labelling is not possible. These kinds of analysis help to get the de-

velopment index of an area. These also helps to assign important resources such as hospital, fire station, etc to certain areas where there is very little of the available resources, hence helps in area/ city planning.

On a surveillance level, it helps to track objects from satellite video feeds and find lost objects from a wide area. These also helps in detecting trafficking and other stuffs in border area, where policing is not possible, for faster allocation of resources. Automatic labelling of roads and buildings helps to get different ideas about the concentration of populations in a given area. Segmented roads can also help to calculate distances between two places hence, helping cartographers in their jobs. Satellite segmentation deals with segmenting objects of various shapes and sizes, hence creating a robust model will also help in medical sectors and robot visions. There are limitless innovative opportunities in this field, and we have just scratched the surface of the iceberg.

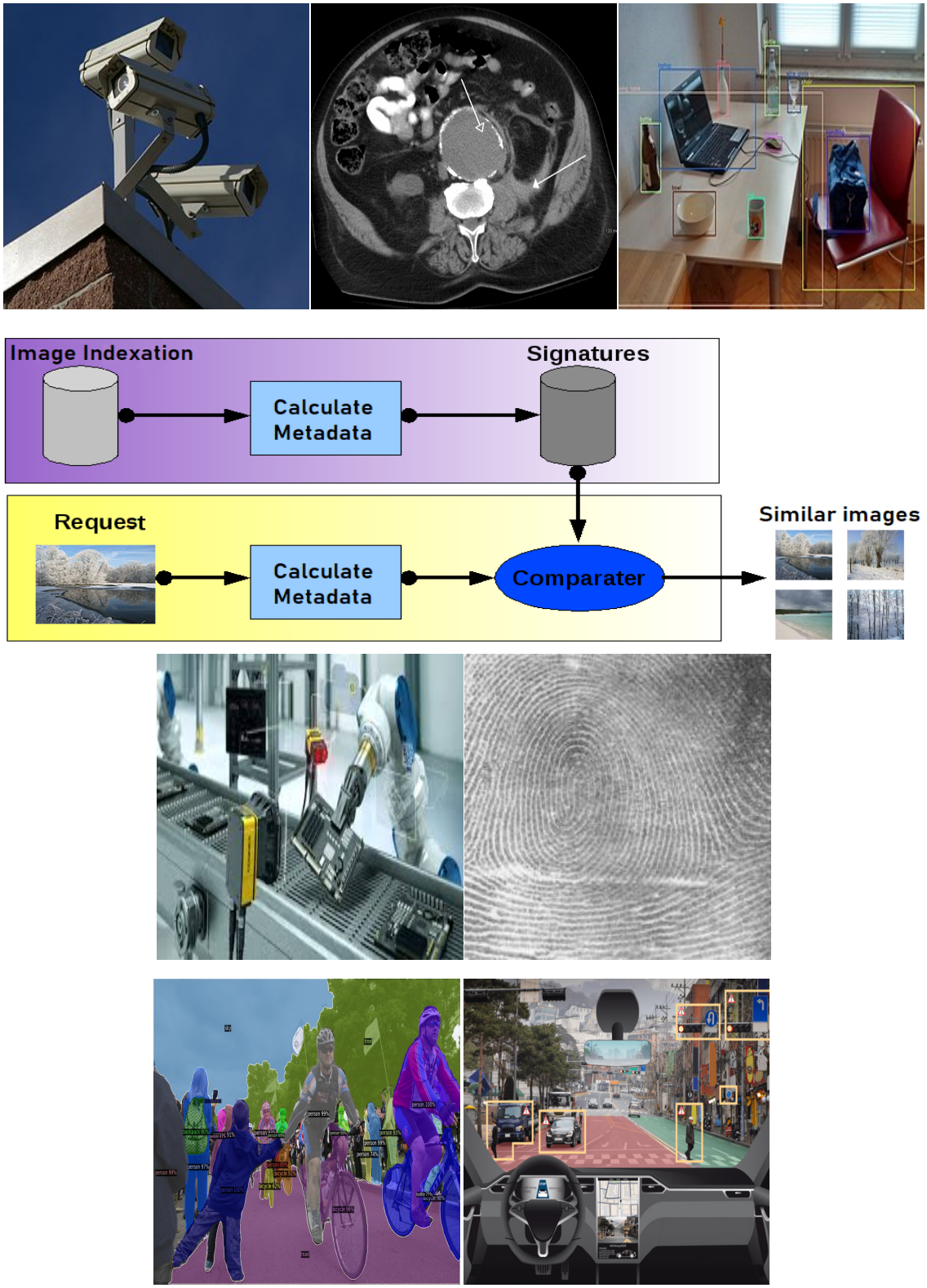


Figure 1.2: From top left to bottom right: Some sectors which utilizes image segmentation pipeline in their systems. Surveillance (P.C.: Wikimedia), Medical Segmentation (P.C.: Wikimedia), Detection (P.C.: Wikimedia), Image retrieval (P.C.: Wikimedia), Machine Vision (P.C.: Cognex), Finger Print Recognition (P.C.: Wikimedia), Image Tagging (P.C.: Facebook Research), Driving Cars (P.C.: Cloudfront)

Chapter 2

Overview of the datasets

2.1 Roads

2.1.1 Modified Pix2Pix dataset

Modified Pix2Pix dataset contains 1096 images of dimension 600x1200 of image and mask pairs, i.e., each tile is 600x600x3 with 3 channel masks as shown in Figure 2.1. We have applied Otsu's thresholding to get single channel road masks for each of the datsets. Hence, we get 1096 image mask pairs from the dataset. The resulting size of the dataset is 238.65 MB.

2.1.2 Jimutmap (Apple Maps Scraper, 2019)

We have created a scraper to get any amount of data from apple maps (`https://satellites.pro/`). The tiles are related to the latitude and longitude by the following reltions:

$$x_{tile} = 2^{zoom} \times \frac{(longitude + 180)}{360}$$

$$lat_{radius} = lat_{degree} \times \frac{\pi}{180}$$

$$y_{tile} = \frac{2^{zoom} \times (1 - \frac{\log\left(\tan(lat_{radius}) + \frac{1}{\cos(lat_{radius})}\right)}{\pi})}{2}$$

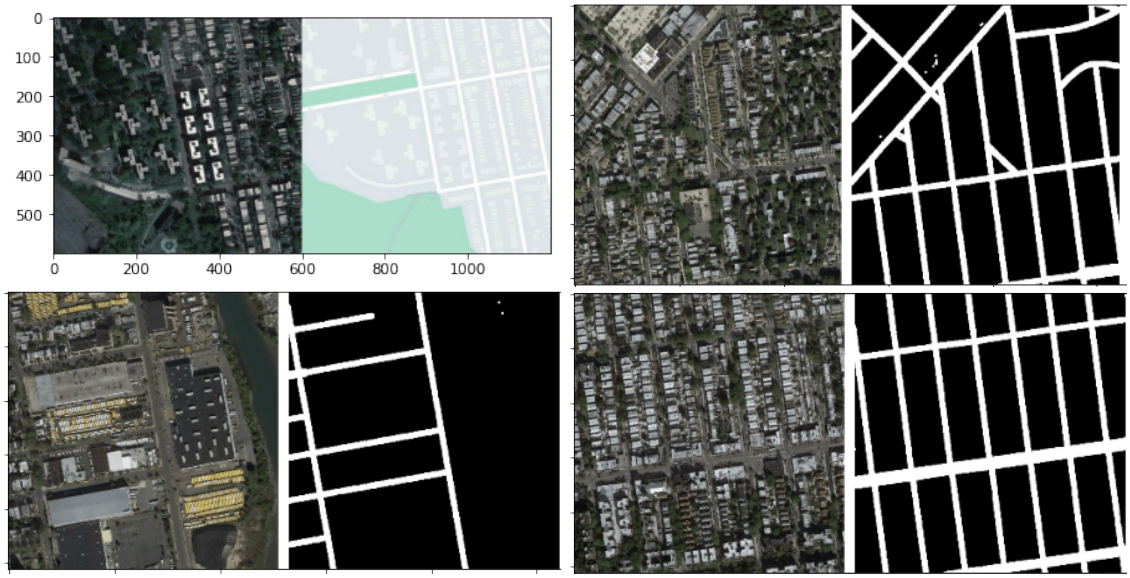


Figure 2.1: From top left to bottom right: A sample of image from Pix2Pix datset, others are the image masks pairs along with threshold applied masks.

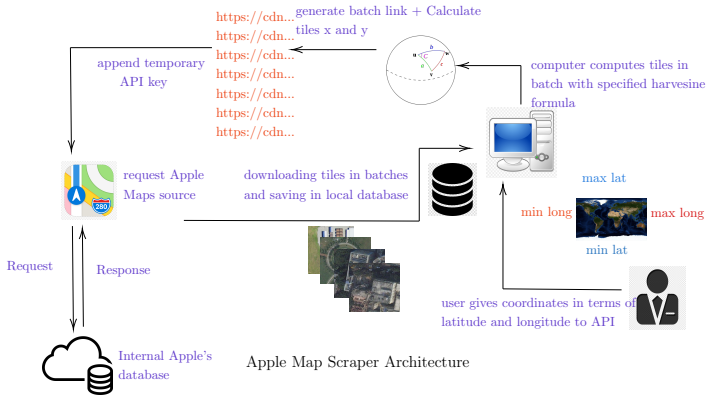


Figure 2.2: Architecture of Apple Maps Scraper, a.k.a. jimutmap.

These complex equations form a way to index tiles of a given latitude and longitude. The zoom and resolution can be adjusted via the zoom parameter, which can be at max = 19 which is the most zoomed version. All the tiles that are retrieved are of 256x256 dimension. For a particular place we scrap the image and the road mask tiles. The reason behind this complex equation is haversine [2] distance between two places, due to the Earth's geoid shape. We exploit these vulnerabilities to get any tiles present in the apple maps. We apply some thresholding similar to the creation of modified pix2pix dataset. The tiles are scraped using Jimutmap [3] by providing coordinates of the bounding box of a location. We use multi-threading to scrap all the tiles, which is way much faster. This scraper tool is open sourced under GPL-3

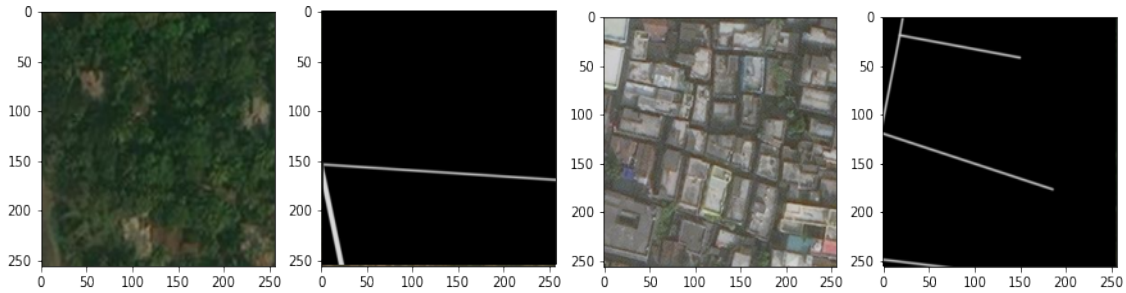


Figure 2.3: Samples obtained from jimutmap dataset which comprises of the major Kolkata area. We can see that the roads may be occluded by trees present in the map. There are a wide texture difference from forest to clumsy buildings all over the dataset.

license, which can be easily installed via a pip command i.e.,

$$pip3\ install\ jimutmap$$

The PYPI web page is given here `https://pypi.org/project/jimutmap/`. The architecture of jimutmap scraper is shown in Figure 2.2. We have downloaded 236606 image-mask pairs comprising about 2.45 GB, from latitude 22.35 to 23.1 and longitude 88.0 to 88.6, which is the major Kolkata region.

For a cleaned and smaller version of the dataset we have discarded the tiles which doesn't have any road mask, so now the JIMUT_MAPS.zip dataset has about 61950 tiles which is about 1.07 GB. A sample of 256x256 map tile is shown in Figure 2.3. The tiles may be occluded by trees and looks challenging. There are a wide range of color values, which needs to be evaluated for determining the road masks, as shown in the figures.

2.1.3 Massachusetts Roads Dataset

This dataset was introduced by Volodymyr Mnih [4], in his Ph.D. thesis. It contains about 1120 files of size 1500x1500x3 with single channel mask. The size of the dataset is about 4.8 GB. Tiles of the dataset has a higher scale, i.e., zoomed out version. Few samples of the data are shown in Figure 2.4, which shows the quality of the dataset. This dataset was used to test how the model will identify thin roads which becomes much harder when the models are deep.

From the data sample as shown in Figure 2.4 we can see that some tiles are

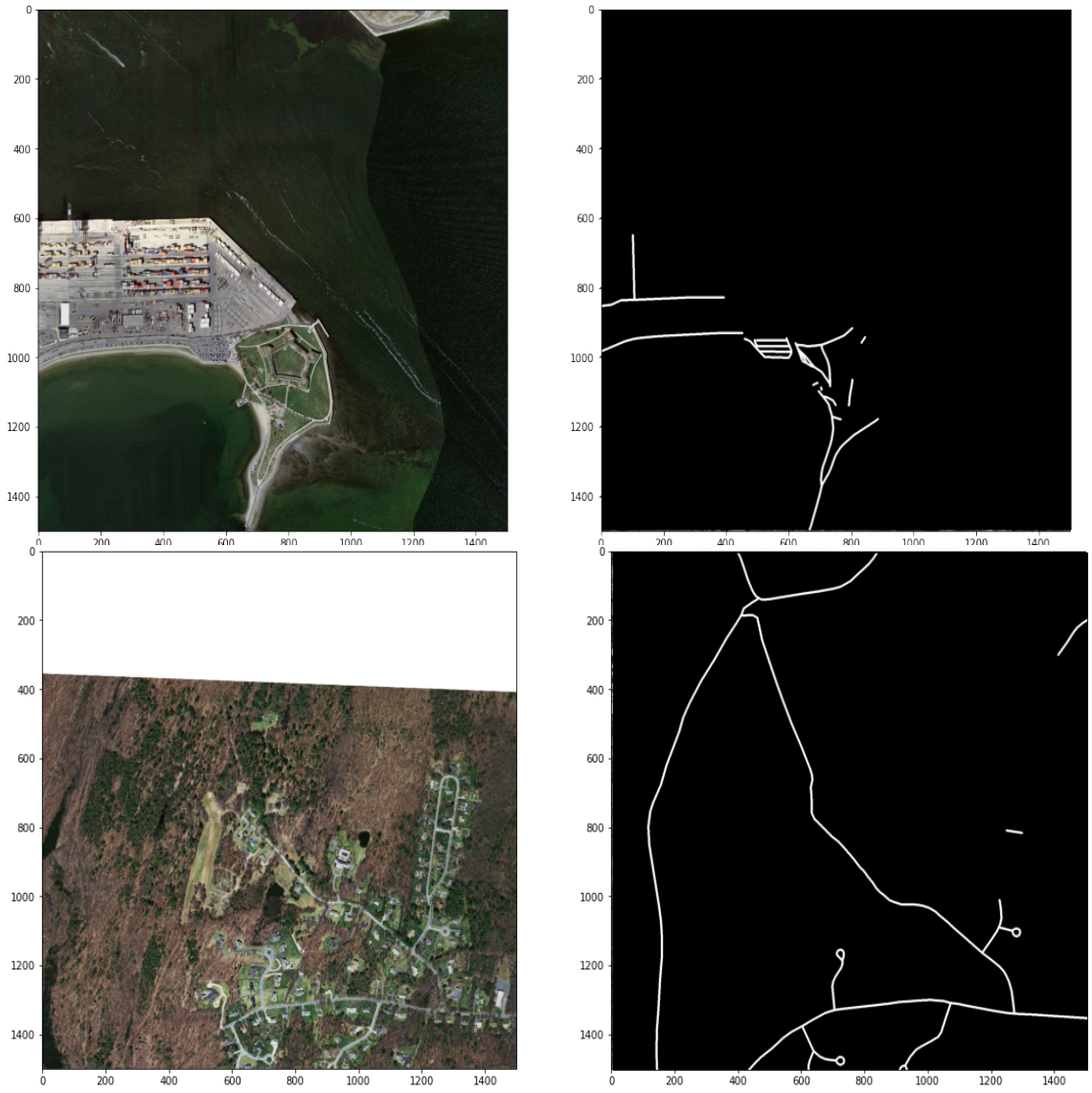


Figure 2.4: Samples shown from Massachusetts Roads Dataset, showing some incomplete images. Data augmentation would increase these inconsistencies proportionately and hence will be of no value. We can see that there are road masks, but the corresponding image is white, which is corrupted (below), due to privacy issues, etc.

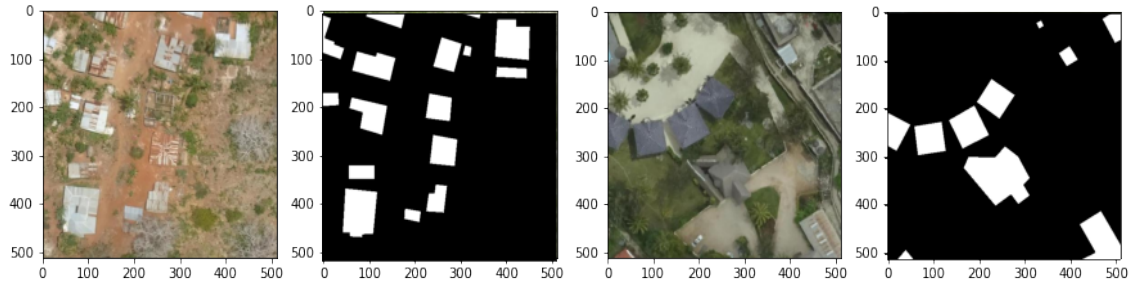


Figure 2.5: Samples shown from Zanzibar OpenAI Building Footprint Mapping dataset.

corrupted. There are several tiles which are corrupted, for instance the regions are occluded in the map while there are segmentation masks and vice versa. This dataset is very challenging, and augmentation of data won't help much, due to these issues.

2.2 Buildings

2.2.1 Building Dataset - Zanzibar OpenAI Building Footprint Mapping (Kaggle)

This dataset [5] was taken from Kaggle (`https://www.kaggle.com/sayantandas30011998/zanzibar-openai-building-footprint-mapping?select=znz-segment-z19`). It consists of 2691 building mask pair. Size of the dataset is 471.86MB. Each tile consists of 500x500x3 dimension, mask is of single channel. Samples from the dataset is shown in Figure 2.5

2.2.2 Building Dataset - Incubit Challenge Dataset

This building's dataset is taken from the Incubit Challenge Dataset. There are 3 classes present in this dataset i.e.:

- **Buildings**
- **Houses**
- **Sheds/Garages**

Task is to find instances of these objects. There are a total of 50 images from training dataset, 22 from validation dataset 8 from test set. The images are of different shapes and sizes, but were resized to 512x512 before passing it to RPN

Figure 2.6: A sample from Incubit Challenge Dataset.

(Region Proposal Network) with a batch size of 2. Since, bounding boxes and masks are given here, we use Mask RCNN to implement this kind of instance segmentation models. A sample of data is shown in Figure 2.6.

2.3 Ship dataset

2.3.1 Ships in Satellite Imagery dataset (Kaggle)

This dataset was taken from Kaggle `https://www.kaggle.com/rhammell/ships-in-satellite-imagery`. It consists of 4000 data tiles containing ship and not containing ship. Each tile consists of

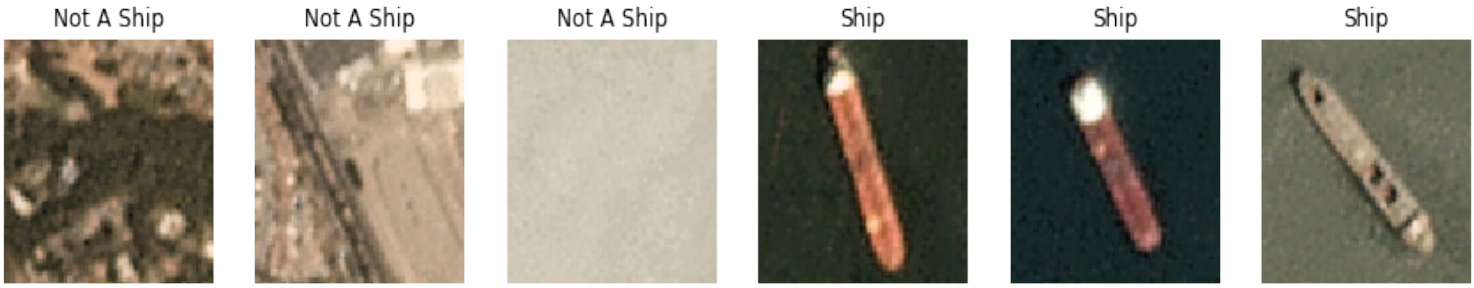


Figure 2.7: Samples shown from Ships in Satellite Imagery dataset.

80x80x3 dimension. Size of dataset is 185.45MB. Sample from the dataset is shown in Figure 2.7. This problem can be considered as a classification problem, if we consider several grids, and search through them deterministically.

Chapter 3

Metrics Used

Here are some of the metrics used for analyzing the performance of the segmentation models, the notations are as follows:

True Postive (**TP**), True Negative (**TN**), False Postive (**FP**), False Negative (**FN**), Accuracy (**AC**), Recall or Sensitivity (**SE**), Specificity (**SP**), Precision (**PC**), F1-score (**F1**), Jaccard Similarity (**JS**), Dice Coefficient (**DC**), Ground truth (**GT**), and segmented result (**SR**).

$$AC = \frac{TP + TN}{TP + TN + FP + FN}$$

$$SE = \frac{TP}{TP + FN}$$

$$SP = \frac{TN}{TN + FP}$$

$$PC = \frac{TP}{TP + FP}$$

$$F1 = \frac{2 \times (PC \times SE)}{PC + SE}$$

$$JS = \frac{GT \cap SR}{GT \cup SR}$$

$$DC = 2 \times \frac{GT \cap SR}{GT + SR}$$

Since this is a single mask segmentation problem, each pixel may be formulated as a classification problem. We can use the Binary Cross Entropy as the loss function, which can be formulated as below:

$$L_{y'}(y) := -\frac{1}{N}\sum_{i=1}^{N}(y'_i \log(y_i) + (1 - y'_i)\log(1 - y_i)) \tag{3.1}$$

Here, y_i is the predicted class per pixel, y'_i is the original pixel value of segmentation mask. All the pixels in the segmentation mask are averaged to get the overall loss. We have used the NADAM optimizer which can be considered as a combination of RMSprop and Stochastic Gradient Descent with Nesterov momentum.

Chapter 4

Related works and comparison with MultiResUNet

4.1 MultiResUNet

There are various models [6] that are there in the UNet [7] family, one of the state-of-the-art is MultiResUNet as shown in Figure 4.1. Related literature deals with minor variants of UNets widely, so we thought it would be better to consider MultiResUNet which can be considered as the potential successor to UNet. It uses multiple residual blocks for better accumulation of gradients. The residual blocks help in faster learning of features. Residual blocks help in faster learning of features. Instead of passing the encoder feature maps directly to the decoder (as in case of general UNets), they use a sequence of convolutional layers (non-linear operations) which reduce the gap between encoder and decoder features. Residual block is shown in Figure 4.2.

4.2 Evaluation of MultiResUNet model to modified Pix2Pix dataset

The MultiResU-UNet model is applied to the modified pix2pix dataset for 150 epochs. Nadam optimizer is used, with a learning rate of 10e-05. We got a test loss of 0.1872, test dice coefficient of 0.5409, jaccard index of 0.3735, test recall of

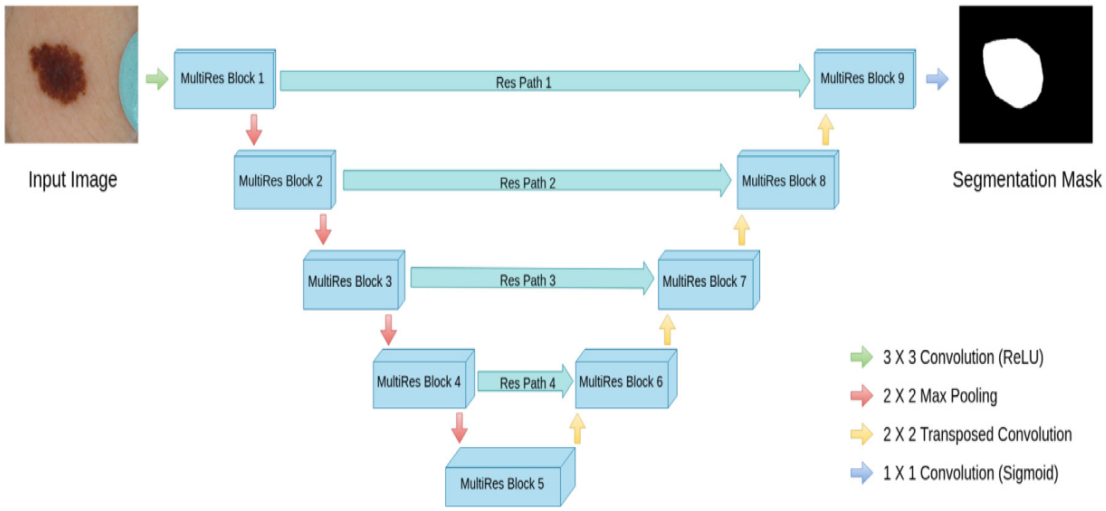


Figure 4.1: Image of MultiResUNet model.

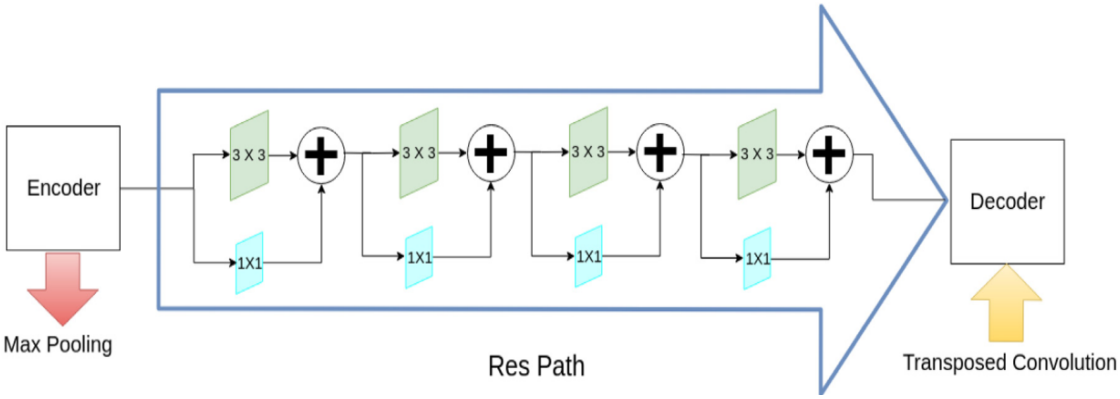


Figure 4.2: Image of Residual block.

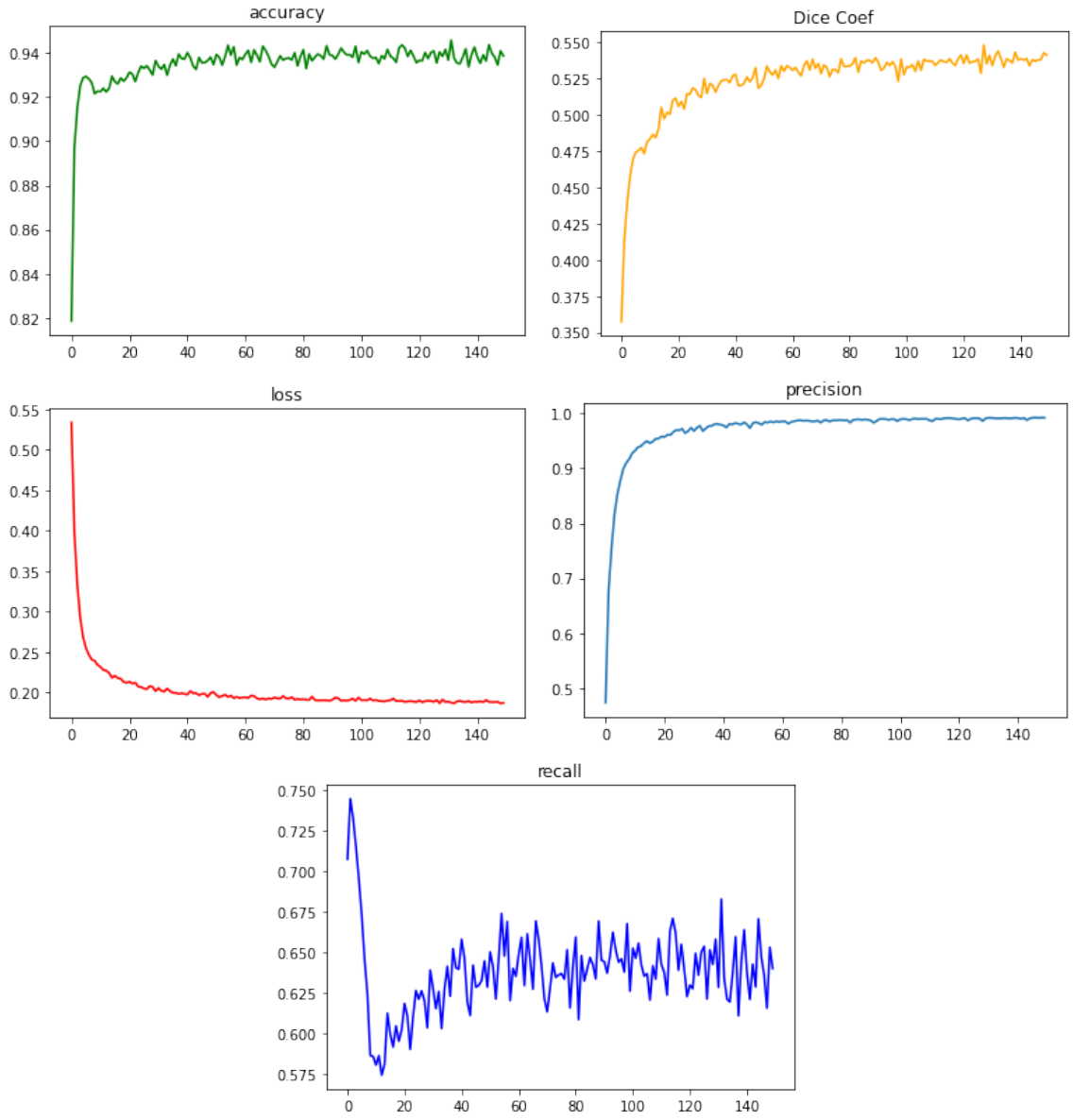


Figure 4.3: From top left: Accuracy, Dice Coefficient, Loss, Precision, and Recall on modified Pix2Pix dataset.

0.6399, test precision of 0.9919 and accuracy of 0.9386. The dataset is challenging and the mask generated is quite good. The graph obtained during training is shown in Figure 4.3.

The predictions obtained from the model are shown in Figure 4.4.

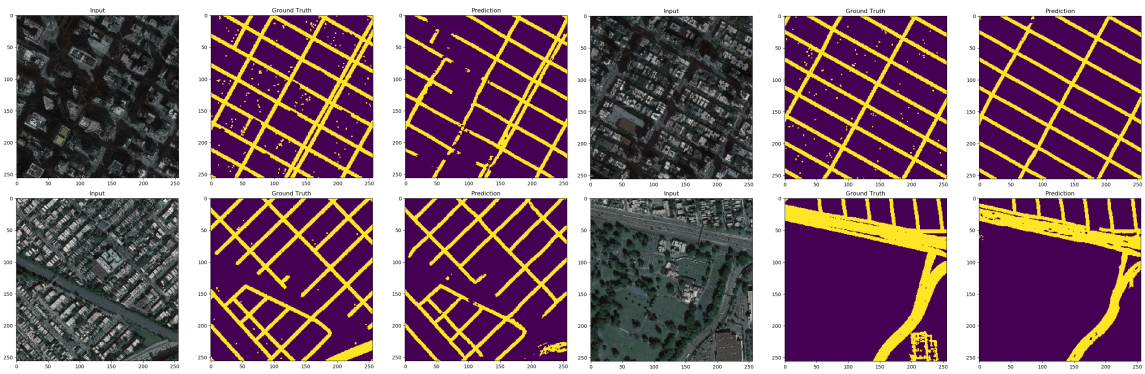


Figure 4.4: Images, ground truth and predictions by MultiResUNet models on modified Pix2Pix dataset.

4.3 Evaluation of MultiResUNet model to jimutmap dataset

The MultiResU-UNet model is applied to the jimutmap dataset for 10 epochs. Nadam optimizer is used, with a learning rate of 10e-05. Got a test loss of 0.9745, test dice coefficient of 0.0255, jaccard index of 0.0130, test recall of 0.2721 and a test precision of 0.0948 and accuracy of 0.9266. The dataset is challenging and the mask generated is not so good. The graph obtained during training is shown in Figure 4.5.

The predictions obtained from the model are shown in Figure 4.6.

4.4 Evaluation of MultiResUNet model to Massachusetts Roads dataset

The MultiResU-UNet model is applied to the Massachusetts Roads dataset for 150 epochs. Adam optimizer is used, with a learning rate of 10e-05. Got a test loss of 0.0620, test dice coefficient of 0.5505, jaccard index of 0.3841, test recall of 0.6195 and a test precision of 0.9628 and accuracy of 0.9810. The training graph is shown in Figure 4.7. The dataset is challenging and the mask generated is quite good as shown in Figure 4.8.

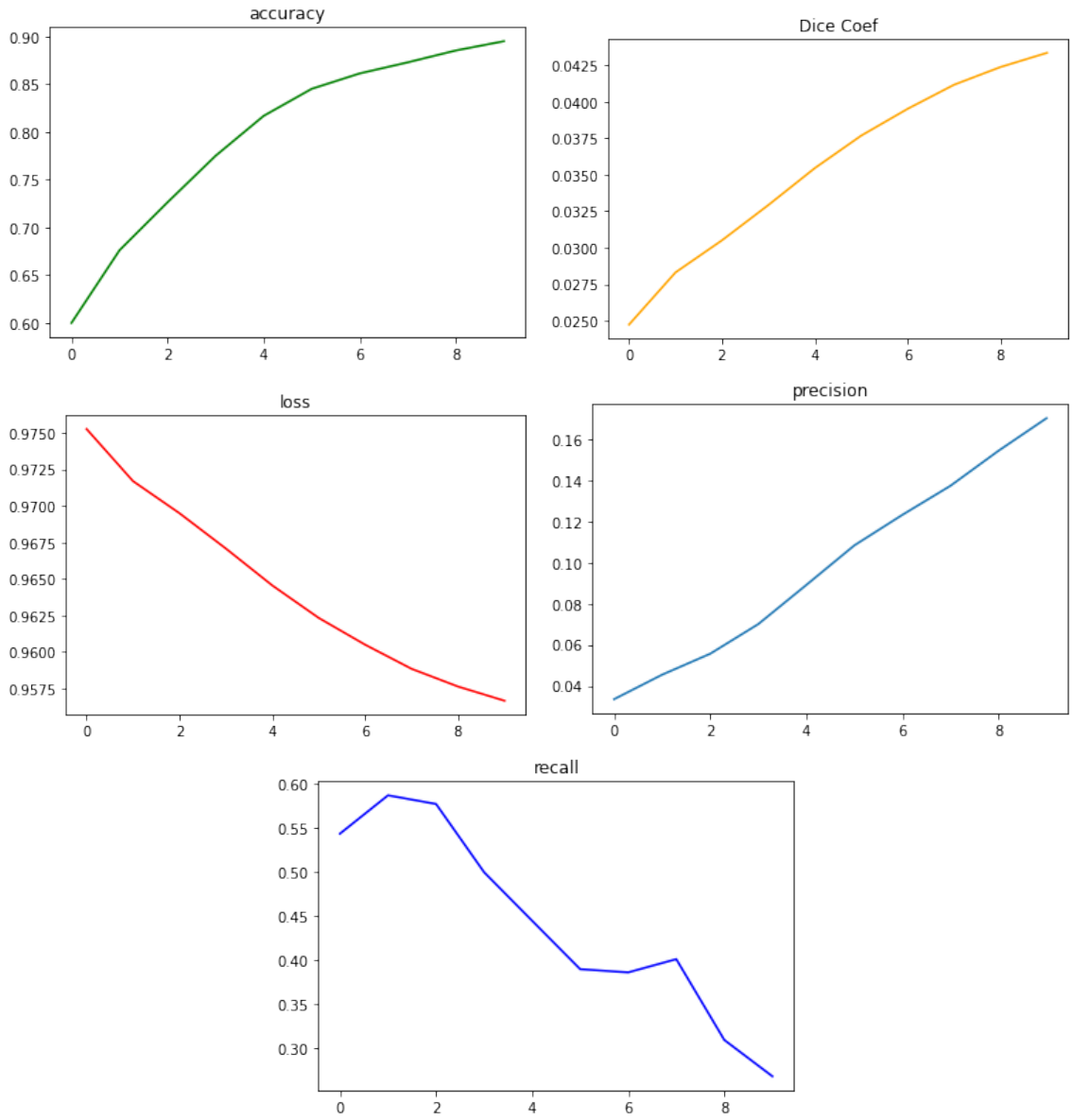


Figure 4.5: From top left: Accuracy, Dice Coefficient, Loss, Precision, and Recall on jimutmap dataset.

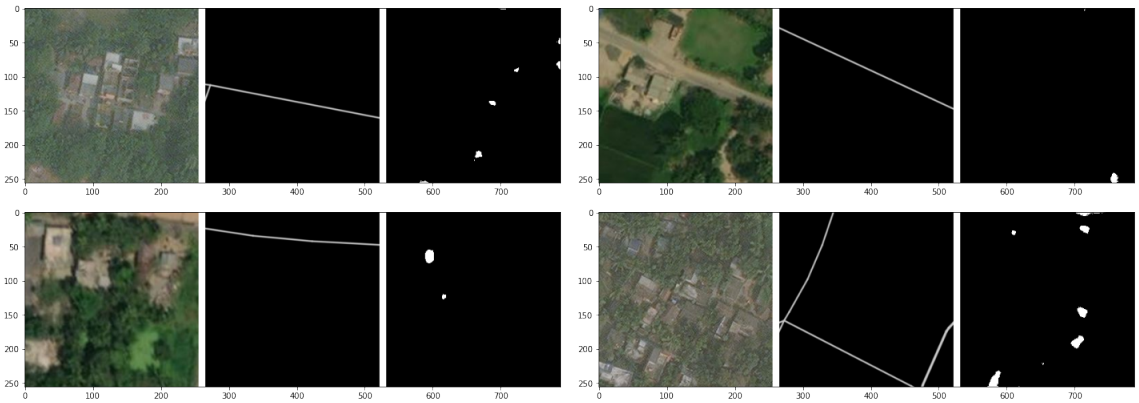


Figure 4.6: Images, ground truth and predictions by MultiResUNet models on jimutmap dataset.

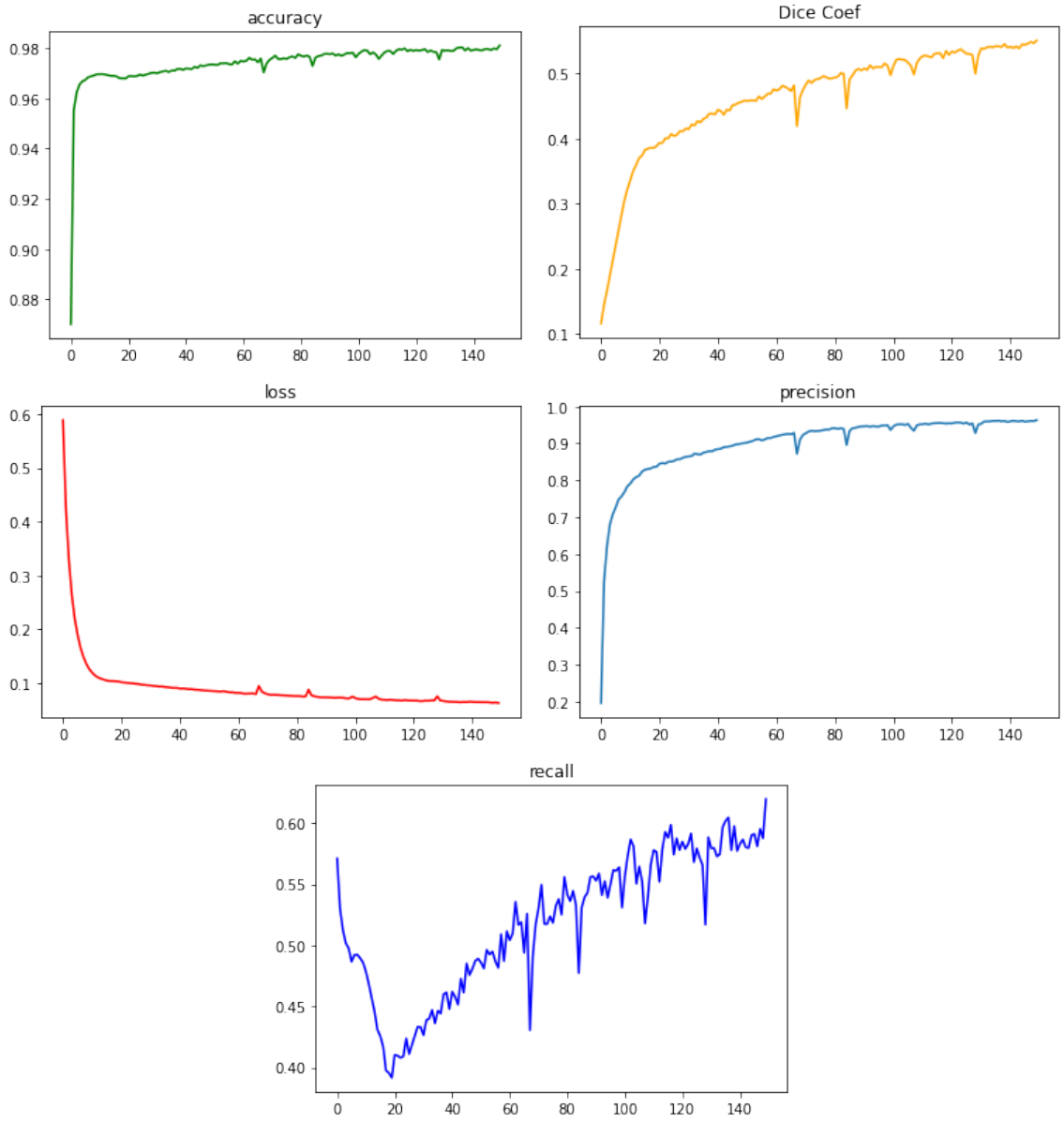


Figure 4.7: From top left: Accuracy, Dice Coefficient, Loss, Precision, and Recall on Massachusetts Roads dataset.

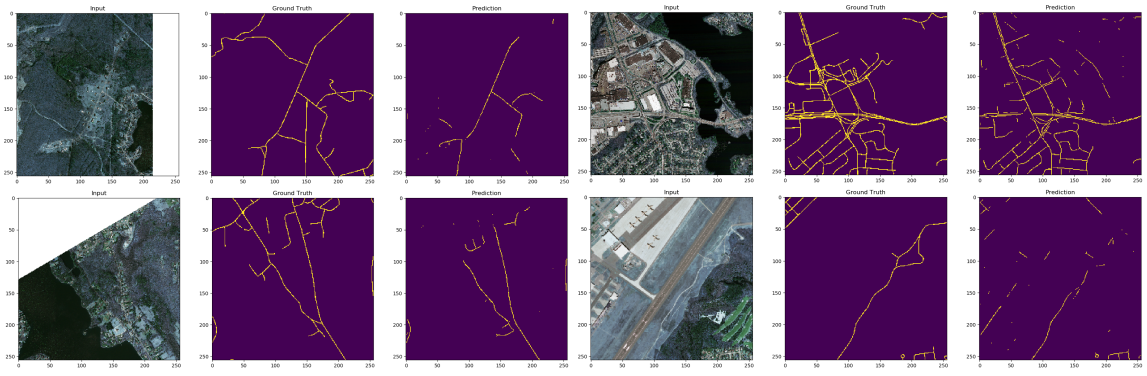


Figure 4.8: Images, ground truth and predictions by MultiResUNet models on Massachusetts Roads dataset.

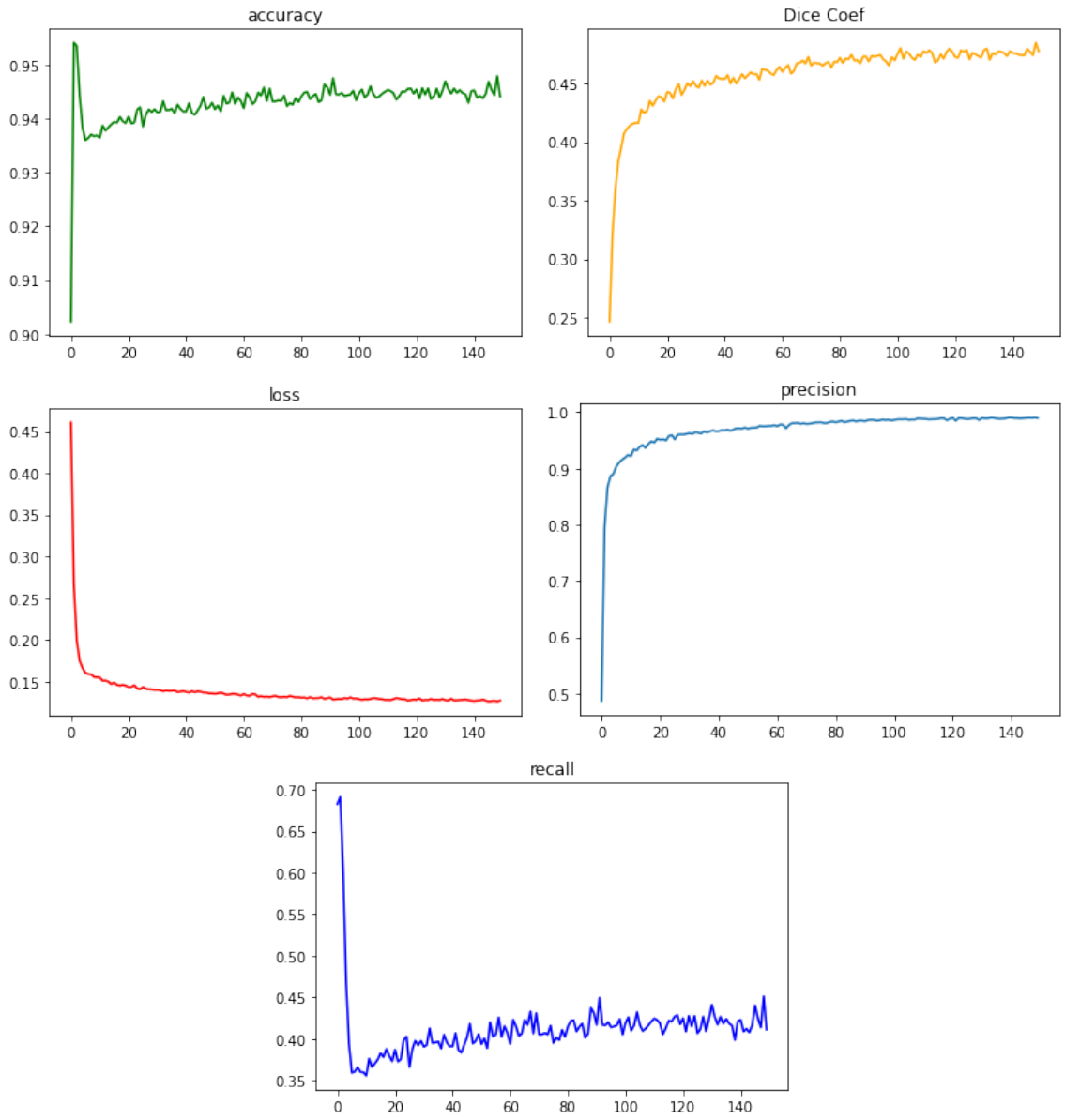


Figure 4.9: From top left: Accuracy, Dice Coefficient, Loss, Precision, and Recall on Zanzibar OpenAI Building dataset.

4.5 Evaluation of MultiResUNet model to Zanzibar OpenAI Building dataset

The MultiResU-UNet [8] model is applied to the Zanzibar OpenAI Building [9, 10] dataset for 150 epochs. Adam optimizer is used, with a learning rate of 10e-05. Got a test loss of 0.1274, test dice coefficient of 0.4773, Jaccard index of 0.3227, test recall of 0.4110 and a test precision of 0.9894 and accuracy of 0.9440. The training graph is shown in Figure 4.9. The dataset is challenging [11–13] and the mask generated is quite good as shown in Figure 4.10.

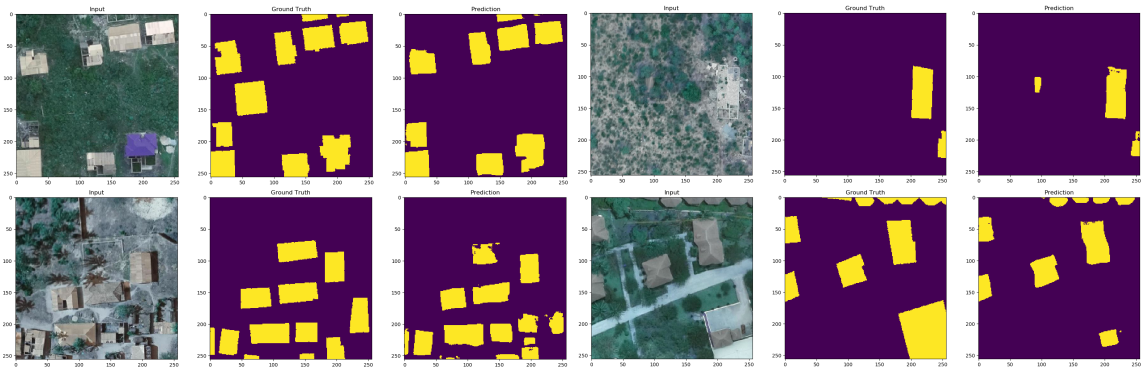


Figure 4.10: Images, ground truth and predictions by MultiResUNet models on Zanzibar OpenAI Building dataset.

	Dice Coeff	Jaccard	Recall	Precision	Accuracy
modified pix2pix	53.09 ± 2.34	38.35 ± 2.78	63.99 ± 1.89	**98.19 ± 3.11**	92.86 ± 2.34
jimutmap	20.33 ± 2.71	25.15 ± 1.37	36.13 ± 2.31	43.21 ± 3.27	53.72 ± 3.21
Massachusetts Roads	**54.03 ± 2.73**	36.57 ± 0.53	**62.57 ± 3.23**	**96.28 ± 2.32**	97.81 ± 2.73
Zanzibar OpenAI Building	47.73 ± 3.21	32.27 ± 3.42	42.10 ± 3.31	**97.91 ± 3.32**	93.31 ± 2.31

Table 4.1: Table showing the results obtained from 5 fold cross validation from each of the datasets by the application of MultiResUNet model.

4.6 Results of 5 fold cross validation

Result of 5 fold cross validation applied to each of the datasets to record the variations are shown in Table 4.1. Those represented by **bold** performs better than it's corresponding models.

The results are presented as $\mu \pm \sigma$, where μ is the mean and σ is the standard deviation of the five folds, i.e.,

$$\sigma = \sqrt{\frac{\sum(x_i - \mu)^2}{N}}$$

This shows that MultiResUNet is better when the feature maps need to capture thinner details, which might be lost in deeper layers in other models.

Chapter 5

Proposed Modified UNet model

5.1 About the proposed model

The proposed model is influenced by deep classifiers which uses Convolution, batch normalization and ReLU as activations in series. The deeper the model gets; the more semantic features are captured. Batch normalization and residual paths helps to mitigate the vanishing gradient problem to some extent. The proposed model is shown in Figure 5.1. We have seen that this 50-layer model is optimal for segmenting certain datasets, going deeper will result in vanishing gradient, resulting in bad segmentation. The model has about 5M parameter, which is smaller than any model, so can be embedded into any low powered devices for run time segmentation.

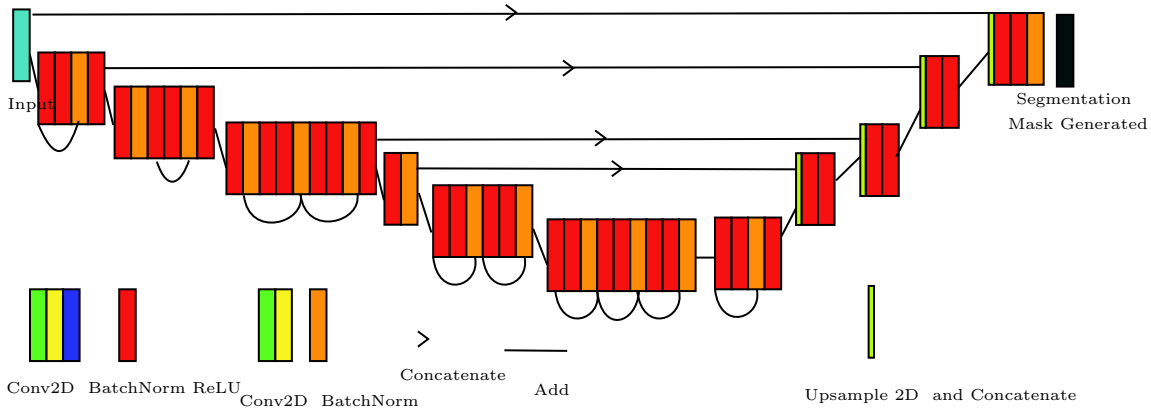


Figure 5.1: A very Deep UNet Model is Proposed.

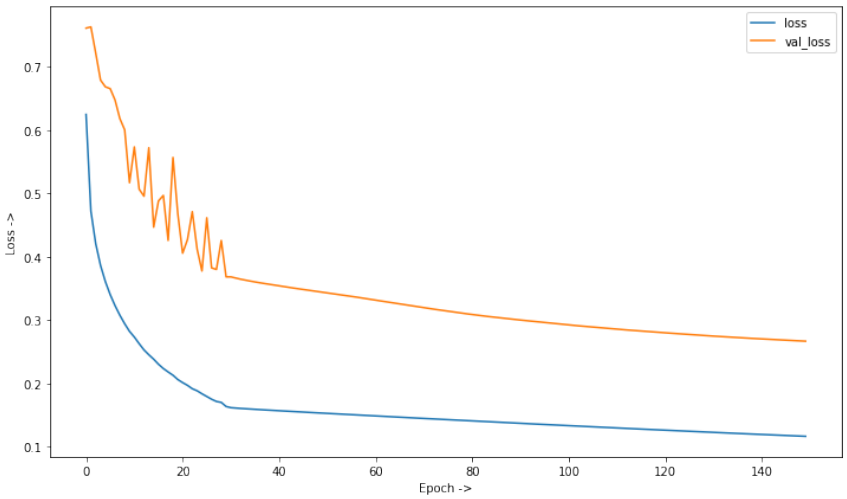


Figure 5.2: Graph of loss obtained during training and validation for Pix2Pix dataset when applied to Modified UNet.

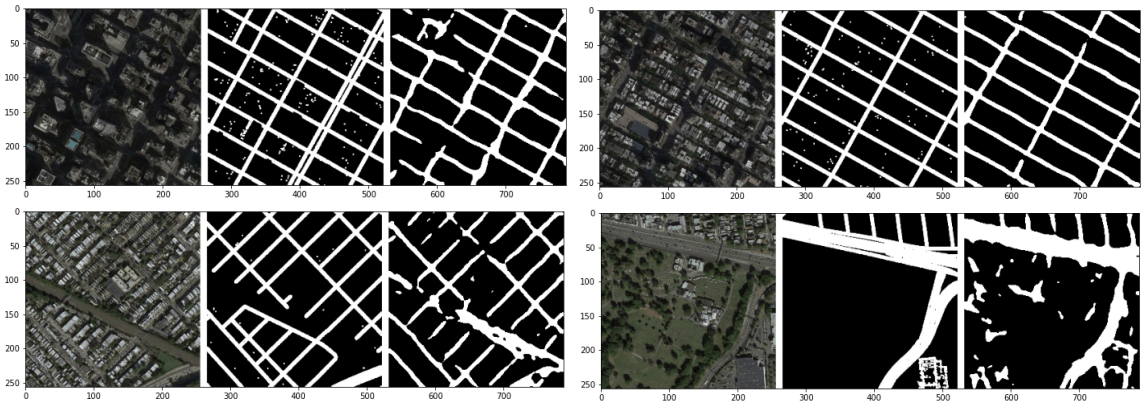


Figure 5.3: Images, ground truth and predictions by Modified UNet model on modified Pix2Pix dataset.

5.2 Evaluation of Modified UNet model to modified Pix2Pix dataset

The Modified UNet model is applied to the pix2pix dataset for 150 epochs. Nadam optimizer is used, with a learning rate of 10e-05. A test loss of 0.2616, test dice coefficient of 0.7392, test recall of 0.6537 and a test precision of 0.8173 is obtained. The dataset is challenging and the mask may differ here and there from the ground truth. The graph obtained during training is shown in Figure 5.2. The results of prediction of the ground truth masks is shown in Figure 5.3.

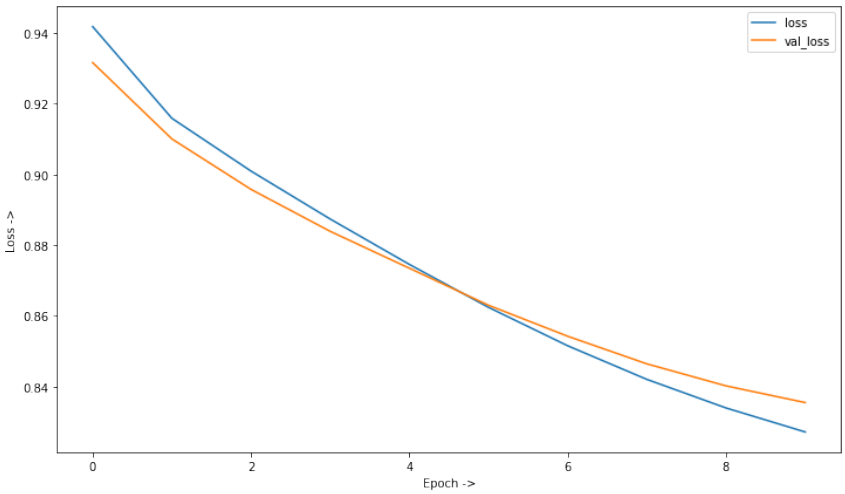


Figure 5.4: Graph of loss obtained during training and validation for jimutmap dataset when applied to Modified UNet.

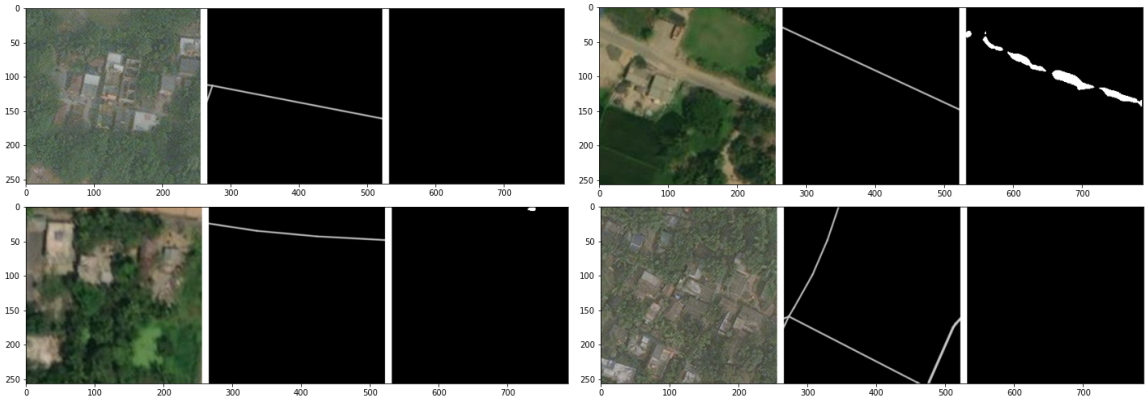


Figure 5.5: Images, ground truth and predictions by Modified UNet model on jimutmap dataset.

5.3 Evaluation of Modified UNet model to jimutmap dataset

The Modified UNet model is applied to the jimutmap dataset for 10 epochs. Nadam optimizer is used, with a learning rate of 10e-05. A test loss of 0.8332, test dice coefficient of 0.1667, test recall of 0.2769 and a test precision of 0.2379 is obtained. The training graph is shown in Figure 5.4. The dataset is very challenging and the mask generated is not up to the mark. Sample of the mask generated along with the ground truth is shown in Figure 5.5.

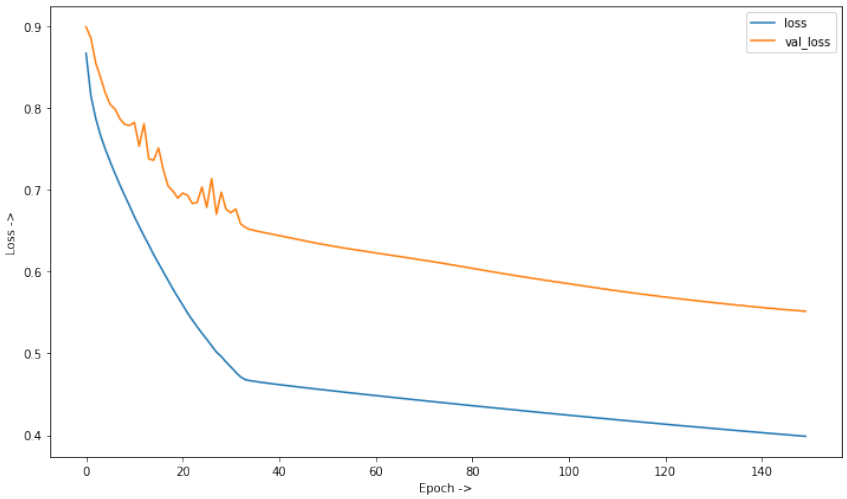


Figure 5.6: Graph of loss obtained during training and validation for Massachusetts road dataset when applied to Modified UNet.

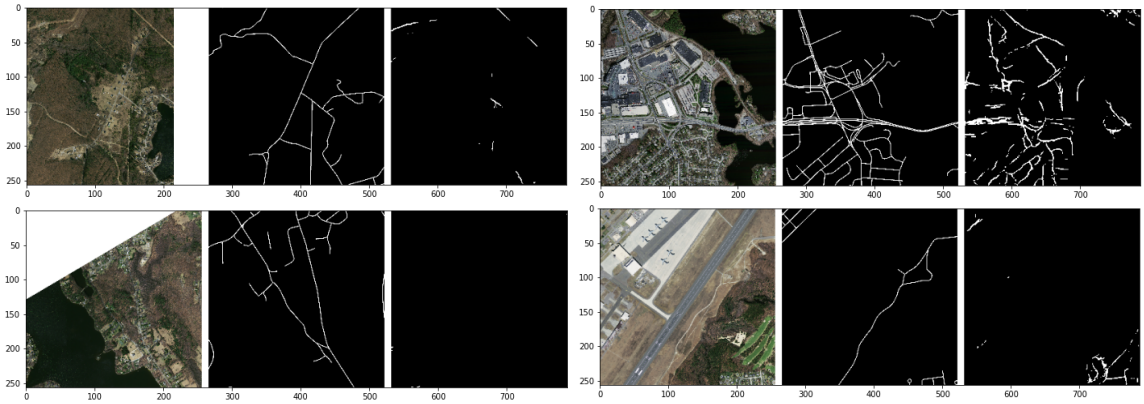


Figure 5.7: Images, ground truth and predictions by Modified UNet model on Massachusetts Road dataset.

5.4 Evaluation of Modified UNet model to Massachusetts Road dataset

The Modified UNet model is applied to the Massachusetts Road for 150 epochs. Nadam optimizer is used, with a learning rate of 10e-05. We received a test loss of 0.5539, test dice coefficient of 0.4456, test recall of 0.4525 and a test precision of 0.5323. The training graph of loss is obtained as shown in Figure 5.6. The dataset is challenging and the mask generated is not up to the mark since the ground truth is thinner as shown in Figure 5.7.

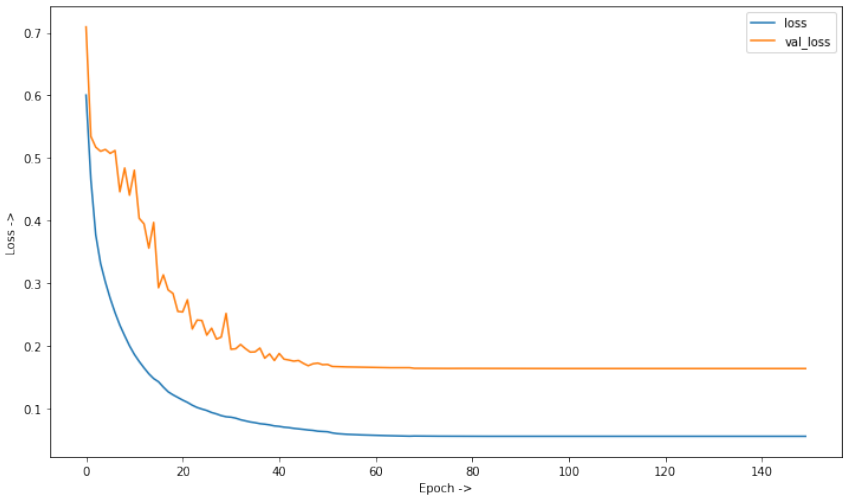


Figure 5.8: Graph of loss obtained during training and validation for Zanzibar OpenAI Building dataset when applied to Modified UNet.

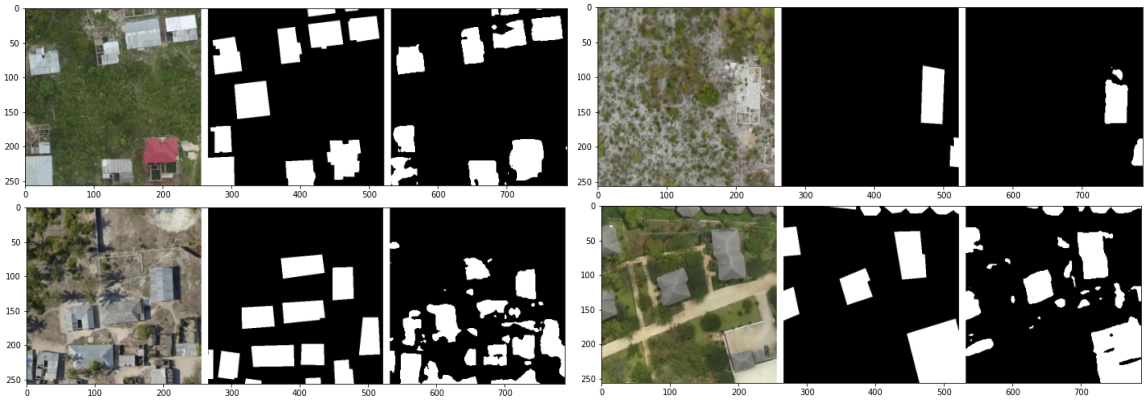


Figure 5.9: Images, ground truth and predictions by Modified UNet model on Zanzibar OpenAI Building dataset.

5.5 Evaluation of Modified UNet model to Road-Zanzibar OpenAI Building dataset

The Modified UNet model is applied to the Zanzibar OpenAI Building dataset for 150 epochs. Nadam optimizer is used, with a learning rate of 10e-05. Got a test loss of 0.1873, test dice coefficient of 0.8121, test recall of 0.7806 and a test precision of 0.8624. The training graph is shown in Figure 5.8. The dataset is challenging and the mask generated is quite good as shown in Figure 5.9.

	Dice Coefficient	Jaccard	Recall	Precision	Accuracy
modified pix2pix	**72.93 ± 2.31**	**63.17 ± 2.56**	**64.37 ± 1.12**	81.73 ± 1.78	**98.18** ± 3.21
jimutmap	**31.27 ± 3.31**	**37.17 ± 1.93**	**46.53 ± 3.11**	**63.73 ± 3.11**	**97.83 ± 3.18**
Massachusetts Roads	43.51 ± 2.73	32.17 ± 3.15	44.25 ± 2.31	53.23 ± 2.27	92.17 ± 2.19
Zanzibar OpenAI Building	**81.11 ± 3.21**	**75.11 ± 1.31**	**79.30 ± 2.70**	85.24 ± 3.21	**97.52 ± 2.98**

Table 5.1: Table showing the results obtained from 5 fold cross validation from each of the datasets by the application of Modified UNet model.

5.6 Results of 5 fold cross validation

Result of 5 fold cross validation applied to each of the datasets to record the variations are shown in Table 5.1. Those represented by **bold** performs better than it's corresponding models.

The results are presented as $\mu \pm \sigma$, where μ is the mean and σ is the standard deviation of the five folds, i.e.,

$$\sigma = \sqrt{\frac{\sum(x_i - \mu)^2}{N}}$$

This shows that Modified UNet performs better than MultiResUNet in most (3/4) of the dataset.

Chapter 6

Visual comparison of the models

The visual comparison of the models are shown in Table 6.1. From the table we can see that the proposed UNet performs better than MultiResUNet in modified Pix2Pix dataset. Similarly, it also performs better in jimutmap dataset and zanzibar openAI building dataset. In case of Massachusetts roads dataset we can see that MultiResUNet performs better than the proposed UNet mostly due to the deeper connections between the encoder and the decoder.

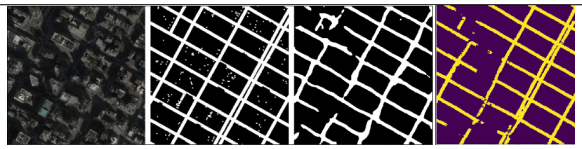
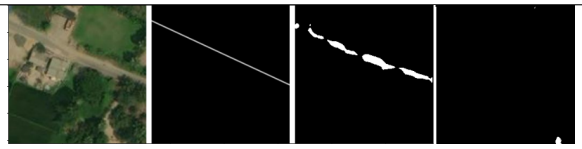
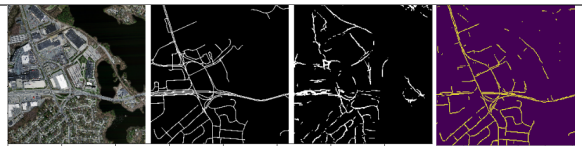
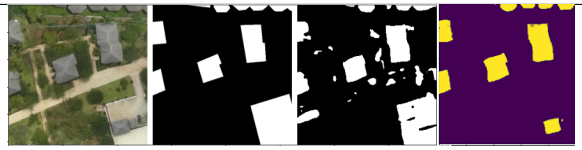
	Image	GT	Our	MRUNet
pix2pix dataset				
jimutmap dataset				
Massachusetts Road dataset				
Zanzibar OpenAI Building dataset				

Table 6.1: A table showing visual comparison of the predicted results.

Chapter 7

Detecting Ships from above

We have proposed a 12 layer deep convolutional neural network model to predict whether an image is a ship [14] or not? The problem is simple, but extremely time consuming. We divide an image to a set of grids and we scan through them. The model is shown in Figure 7.1. There are few false positives in the images, and we get more or less decent results.

We have trained the model using Nadam optimizer with a learning rate of 0.001, and a loss function of binary Cross entropy. The model is trained for 1000 epochs and we get a loss of 0.0640, an accuracy of 0.9802, a validation loss of 0.0457 and a validation accuracy of 0.9889. The training graph of accuracy is shown in Figure 7.2 and loss is shown in Figure 7.2.

If this was a satellite video feed, we could subtract frames and get the moving ships easily by tracking them. Some of the results are shown in Figure 7.3, 7.4, 7.5 and 7.6.

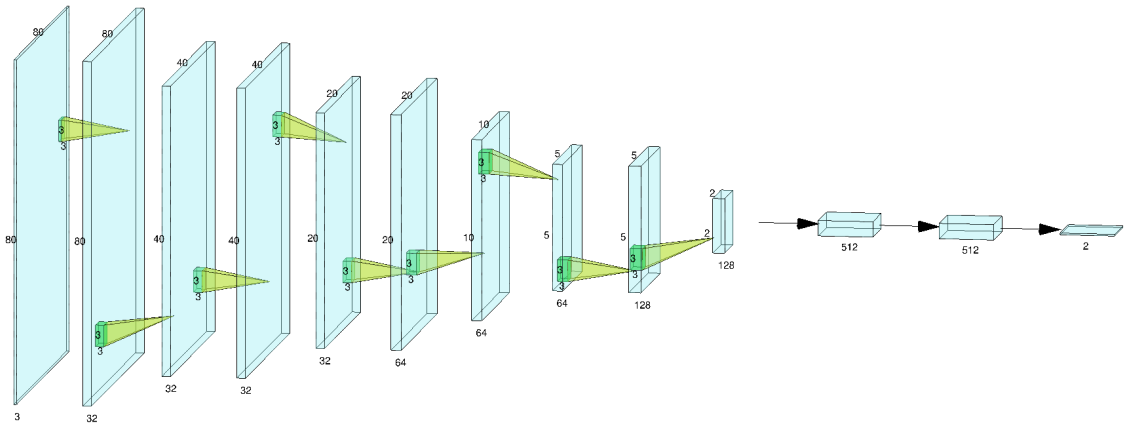


Figure 7.1: Model for detecting ships.

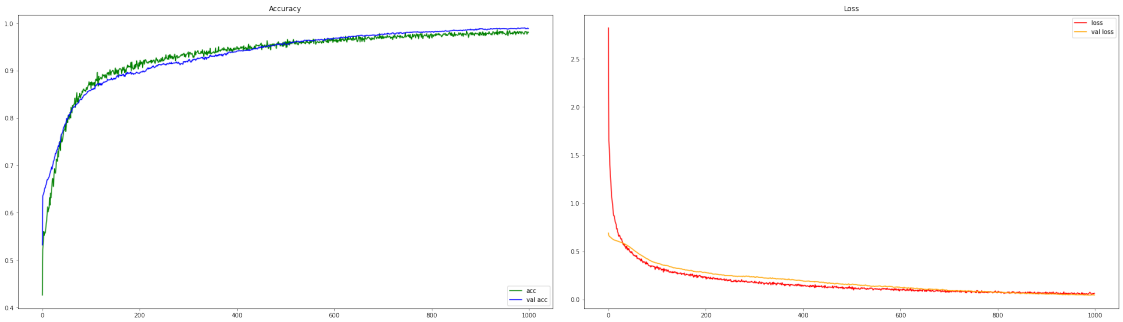


Figure 7.2: Accuracy and Loss for the proposed model trained for 1000 epochs.

Figure 7.3: Ships detected from Los Angeles Bay area.

Figure 7.4: Ships detected from Los Angeles Bay area.

Figure 7.5: Ships detected from San Francisco Bay area.

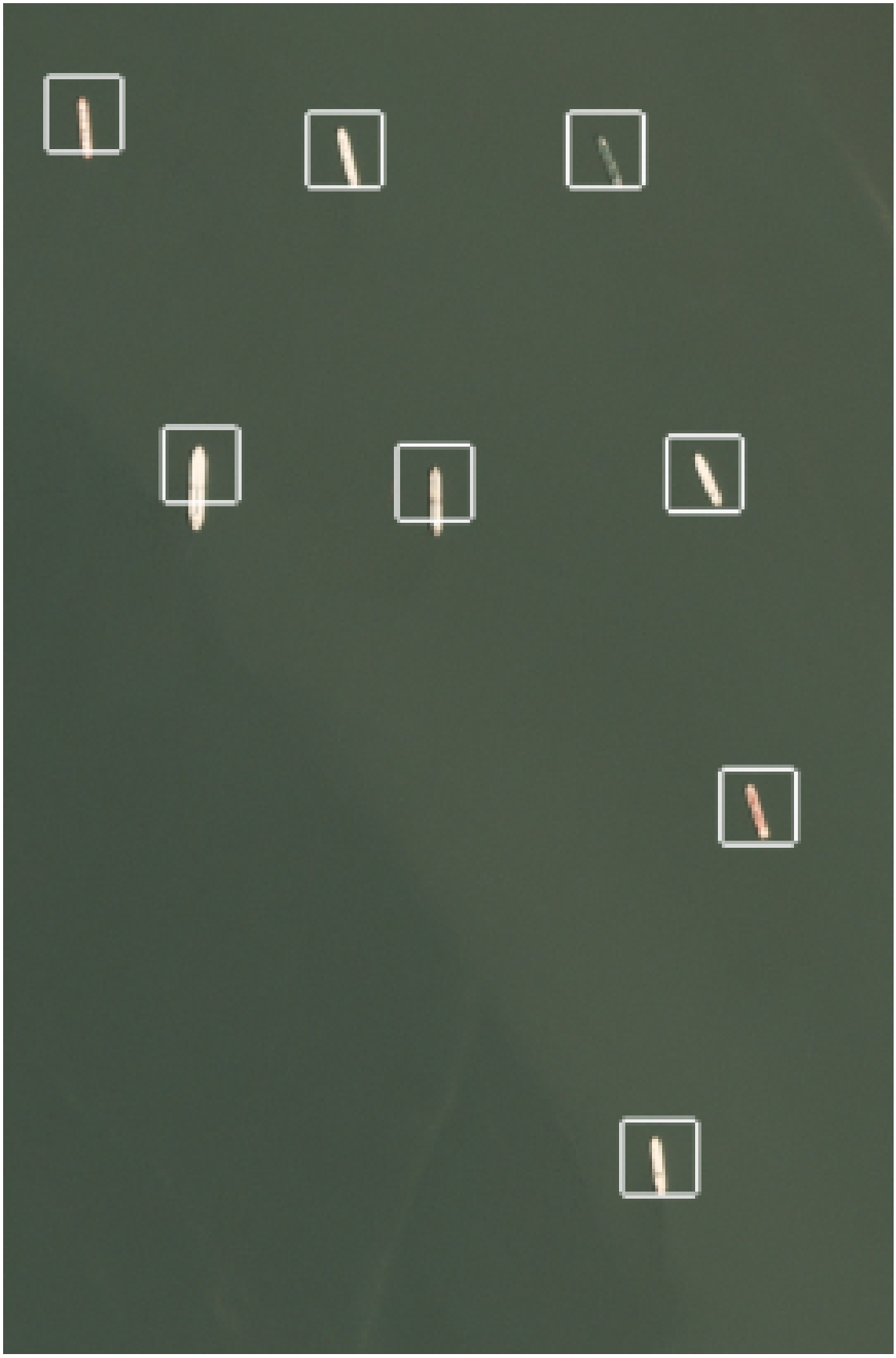


Figure 7.6: Ships detected from San Francisco Bay area.

Chapter 8

Mask RCNN

8.1 About Mask RCNN

Mask RCNN [15] outputs object **bounding boxes**, **classes** and **masks** for an image which is given as input. It uses a convolutional backbone, which is **Feature Pyramid Network (FPN)** for preserving features at different scales. The backbone forms a part of Faster R-CNN backbone. **Region Proposal Networks (RPN)** contains bounding boxes, known as anchors, which helps to detect objects faster. It also uses **ROIAlign** to align the features at different scales using **Bilinear Interpolation**, which helps to **remove location misalignment** caused due to **ROI pooling**.

The illustrative architecture of Mask RCNN is shown in Figure 8.1.

8.2 Mask RCNN - Results

We have performed the test of detecting and masking buildings on the test set of Incubit data challenge, and the results are shown in Figure 8.2 and Figure 8.3.

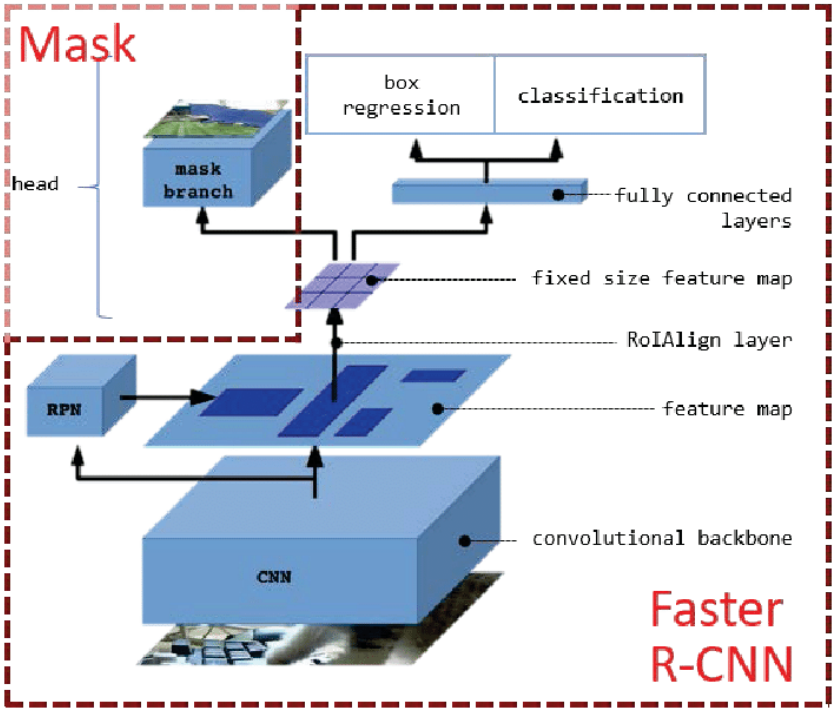


Figure 8.1: Mask RCNN (Picture Courtesy: Research Gate)

Figure 8.2: Buildings detected and segmented from Mask RCNN.

Figure 8.3: Buildings detected and segmented from Mask RCNN.

Chapter 9

Conclusions

We have seen that different style of U-Net architectures perform differently for datasets, i.e., they are dataset dependent. Our main goal was to design a robust type of model which performs well on different datasets when trained, and it performs well on 3 out of 4 dataset. Our proposed U Net variant model gives better dice coefficient and recall on jimutmap, pix2pix and zanzibar openAI dataset, which is a great feat. Our segmentation model may be used in medical datasets also, since this performs well on a range of satellite segmentation datasets. The only modification that can be done to ship segmentation dataset is by passing the extracted images to GPU for faster processing. We have used Mask RCNN to detect buildings, which could be also done for ship detection, and would have given faster results. Current naïve ship detection model is only giving false positive to those stationary samples which looks like ship, and in the case of video feeds, we might get over this problem.

Appendix A

Appendix

A.1 Softwares Used

We have used Keras, Tensorflow and Python3 as our primary tool. Mask RCNN was taken from Matterport's open source implementation.

Bibliography

[1] Wikipedia contributors. *Image segmentation - Wikipedia, The Free Encyclopedia,* . `https://en.wikipedia.org/wiki/Image_segmentation`2020. last accessed on 27.10.2020.

[2] Wikipedia contributors. *Haversine formula - Wikipedia, The Free Encyclopedia,* . `https://en.wikipedia.org/wiki/Haversine_formula`2020. last accessed on 27.10.2020.

[3] J. B. Pal and P. Kahn. *jimutmap,* . `https://github.com/Jimut123/jimutmap`2019.

[4] V. Mnih, *Machine Learning for Aerial Image Labeling,* PhD thesis, (2013).

[5] S. Das. *Zanzibar OpenAI Building Footprint Mapping,* 2019.

[6] V. Iglovikov and A. Shvets. *TernausNet: U-Net with VGG11 Encoder Pre-Trained on ImageNet for Image Segmentation,* 2018.

[7] O. Ronneberger, P. Fischer, and T. Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation,* . In N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015* 234Cham2015. Springer International Publishing.

[8] N. Ibtehaz and M. S. Rahman, *MultiResUNet: Rethinking the U-Net architecture for multimodal biomedical image segmentation,* Neural Networks **121** (2020) 74.

[9] G. Chhor and C. B. Aramburu. *Satellite Image Segmentation for Building Detection using U-net,* . 2017.

[10] A. Khalel and M. El-Saban, *Automatic Pixelwise Object Labeling for Aerial Imagery Using Stacked U-Nets,* CoRR a **bs/1803.04** (2018) [1803.04953].

[11] B. Bischke, P. Helber, J. Folz, D. Borth, and A. Dengel. *Multi-Task Learning for Segmentation of Building Footprints with Deep Neural Networks,* . In *2019 IEEE International Conference on Image Processing (ICIP)* 14802019.

[12] P. Kaiser, J. D. Wegner, A. Lucchi, M. Jaggi, T. Hofmann, and K. Schindler, *Learning Aerial Image Segmentation From Online Maps,* IEEE Transactions on Geoscience and Remote Sensing **55** (2017) 6054–6068.

[13] S. Ohleyer and E. Paris-Saclay. *Building segmentation on satellite images,* . 2018.

[14] user rhammell. *Ships in Satellite Imagery,* 2018.

[15] K. He, G. Gkioxari, P. Dollár, and R. Girshick. *Mask R-CNN,* . In *2017 IEEE International Conference on Computer Vision (ICCV)* 29802017.

Department of Computer Science
Ramakrishna Mission Vivekananda Educational and Research Institute
Kolkata - 711202, India