Overview of the project

The title of our project is ‘Road Network Extraction Based on Deep-learning and Transfer Deep-learning Models Using Satellite Images’.
This project is helpful in various ways to update the maps or to use such data in emergencies such as disaster planning, any military emergency, evacuation planning, or resource allocation. If any new roads are built it will also help to detect those.
Considering the state of the art for this project, already there have been numerous studies done on using ResNet, U-Net, DeepLabV3+, DenseNet, and VGG. There are various preprocessing and post-processing techniques are also included in that.
In the way we are implementing the system, a project is divided into three steps, we will preprocess the data with existing technologies and implement a solution by using preprocessing methods such as Data Augmentation, Data Segmentation, and Image Enhancement on the dataset. Secondly, we will implement different CNN models. We are aiming to use a model fusion of Resnet50 and VGG, where we have concatenated the features extracted from both the models and predicted the output on the test image dataset, and as we will be exploring 3 different preprocessing we will implement the model for each preprocessing technique, where we can analyze models with different preprocessing based on the accuracy. Post-processing techniques that we are implementing are Edge Detection, Morphological operations, and Conditional Random Fields.
To summarize the work we are doing:
We have implemented 3 preprocessing techniques in 2 performed by Bhushan 1 by Deep, 1 fusion model(ResNet and Vgg) by Bhushan,1 U-Net Model by Deep, and 3 post-processing techniques in that 2 performed by Deep and 1 by Bhushan and get our results to compare with existing technologies.

Approach

The approach we have implemented for road extraction consisted of important three steps. At first, we preprocessed the training images which consisted of satellite images and the corresponding mask of that image, using three different preprocessing techniques, which are Image Augmentation, Image Enhancement, and Image Segmentation. Secondly, we used the two different models to train on each preprocessed technique. The models we used are a model fusion of ResNet50 and Vgg19, where we extracted the features from both the transfer deep learning model and concatenated their extracted features. The third important step includes the use of post-processing techniques on predictions of test images, where we have used three different post-processing techniques, which are Edge Detection, Morphological Operations, and Conditional random field. For a better understanding of the flow of approach, refer below diagram.

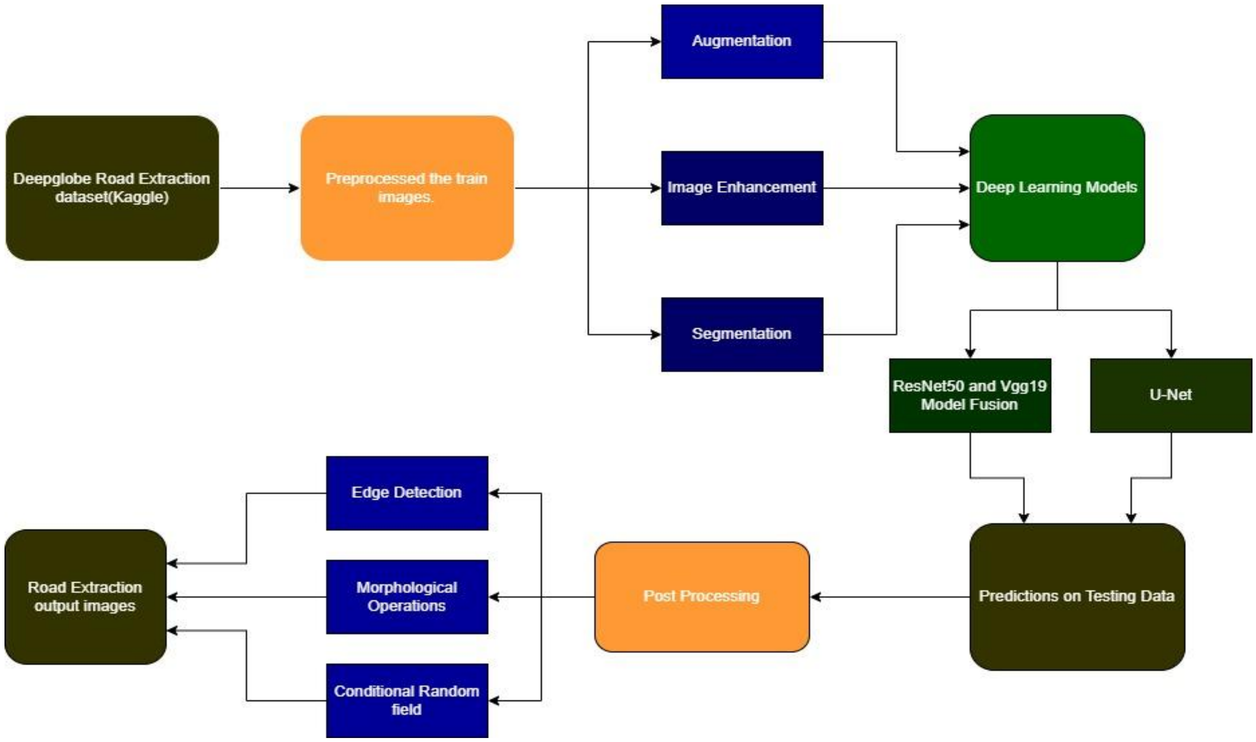


Fig: Block Diagram

We wanted to explore how different preprocessing and post-processing help to affect the output of images from models. Also, while reading some research papers we came across the U-net model as an efficient model to train on road extraction problems. The idea behind implementing model fusion of ResNet50 and Vgg19 was to explore how multimodal works on a specific task. ResNet architectures are known for deep architectures with residual connections, which helps in learning from a vast amount of data without overfitting. Whereas, Vgg19 works better in capturing texture and fine details. The fusion of models makes use of the strength of both models and potentially leads to better performance. Initially, we thought to implement model fusion in such a way that we extract features from ResNet50 and provide the extracted features as input to the Vgg19 model, but we came across many errors in the whole process. Then we completed the fusion process by concatenating the features extracted from both models and then training accordingly.
Referred online resources for concatenating the features of pre-trained models fusion, also referred to Kaggle notebook for defining U-net model. Independently coded sections include, augmentation using ImageDataGenerator, training the model, and testing the model, For other parts, have referred to some online resources such as geeks for geeks, StackOverflow, Analytics Vidhya, and Medium.

Experimental Protocol

We have used the Satellite Road images dataset from Kaggle. It consists of 1 metadata CSV file, 1 class_dict CSV file, and 3 folders namely train test and valid. The train contains 12452 images, including masking and satellite images, while the test contains 1101 and valid contains 1243 satellite images.

First, we trained our models on the training dataset, and then to compare the results we needed to mask the test dataset and compare the results. Currently, we are getting good results on Unet with augmentation as preprocessing and edge detection as post-processing, And also for Unet with image enhancement as preprocessing and Edge detection as post-processing.

Since this dataset contains the maximum number of images, we preferred this over another. In addition to that, it also contains the masking images in the training dataset which are helpful in the model training and preprocessing part. From which model gets an accurate idea for training purposes.

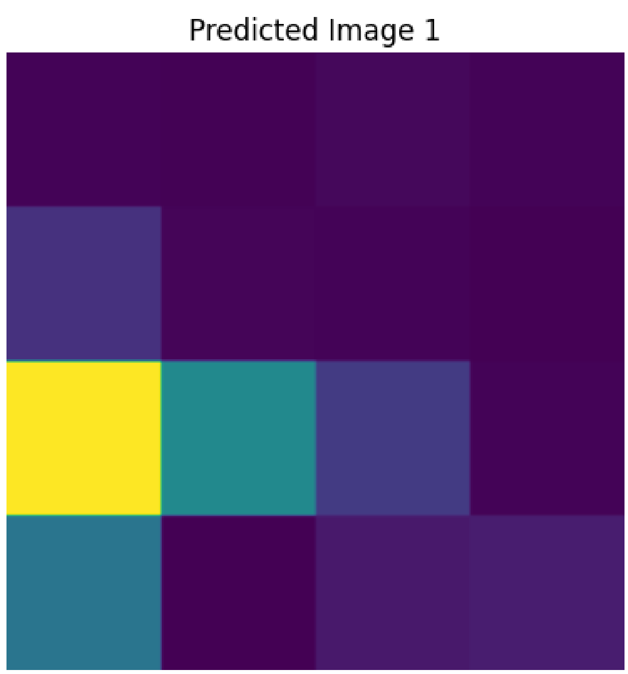
For evaluating the success we are using testing and validation datasets, Initially, we also thought of using F1, Precision, and Recall, but till now we have implemented only the testing datasets.

This project is very computationally heavy since we have to train a large dataset of 3.86 GB. Additionally, we also trained Unet with 7 layers and model fusion consisting of Resnet and VGG, which are very computationally heavy models. Hean runtime required for each part that is preprocessing, postprocessing, and model training is over 60 manhours per person.

Results

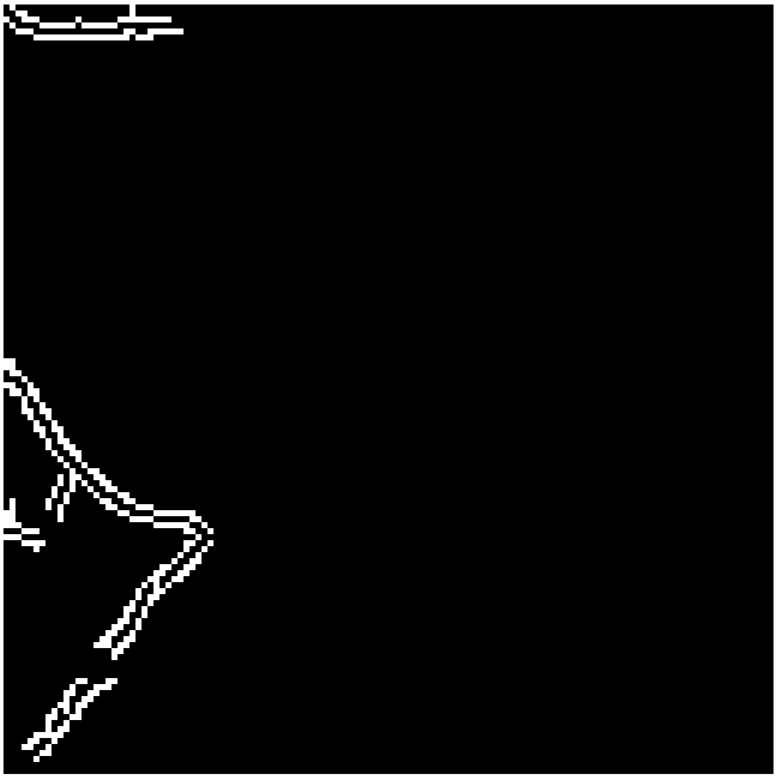
From the preprocessing model implementation and post-processing, we have achieved these results. From the image itself, it is better to understand.

After implementing Image enhancement on Unet and training the model, we predicted the testing images



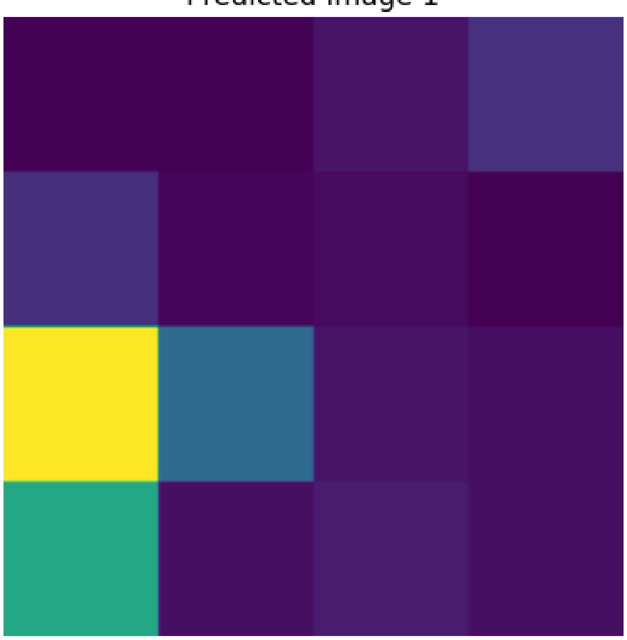
After applying post-processing on it we received an output like this

Edge-detected Image 1

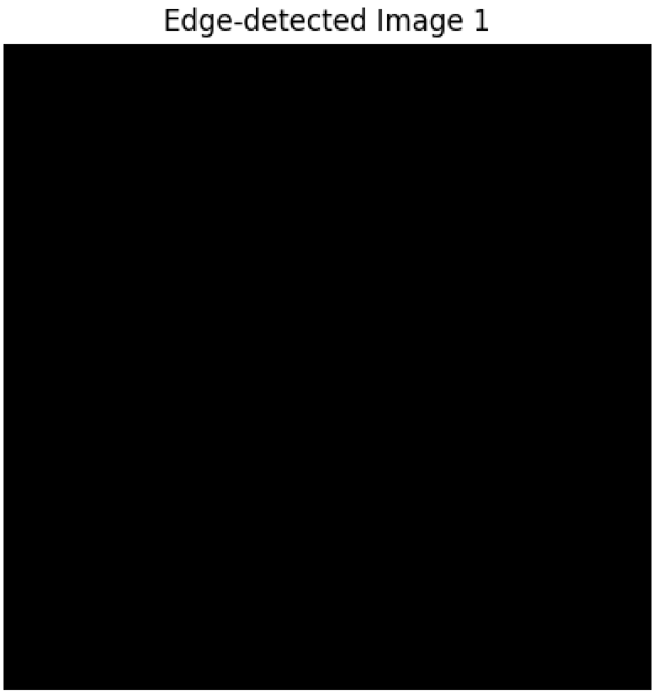


After applying the Image enhancement on the fusion model and training the model, we tested on test dataset
We got output as

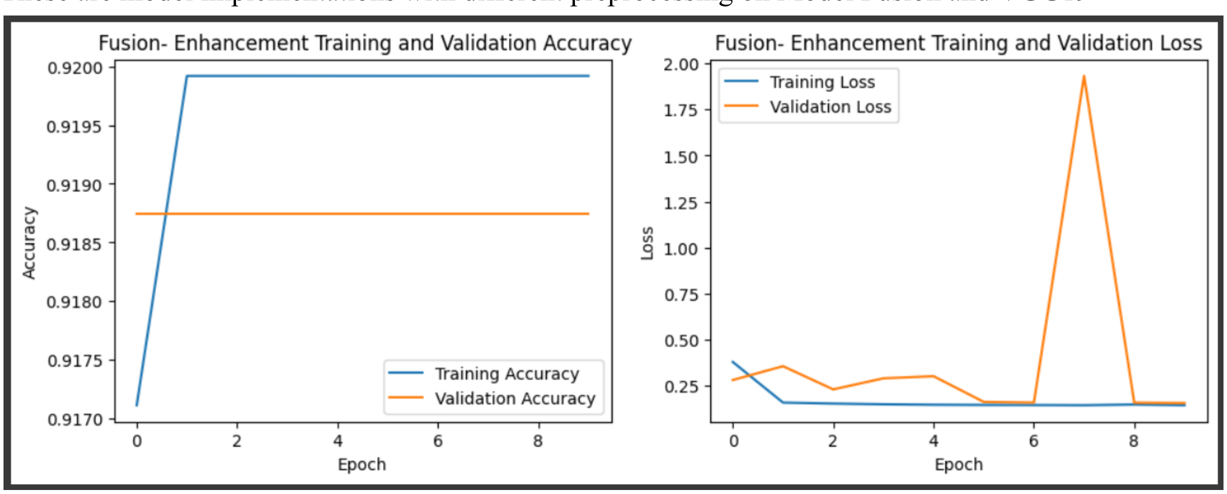
Predicted Image 1



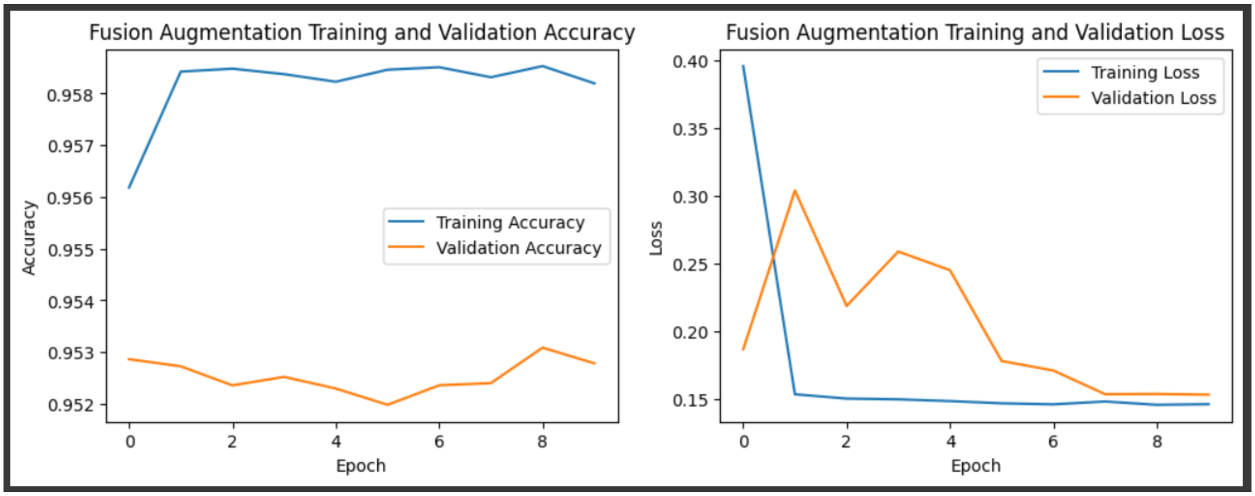
And after applying edge detection as post-processing technique on that we received



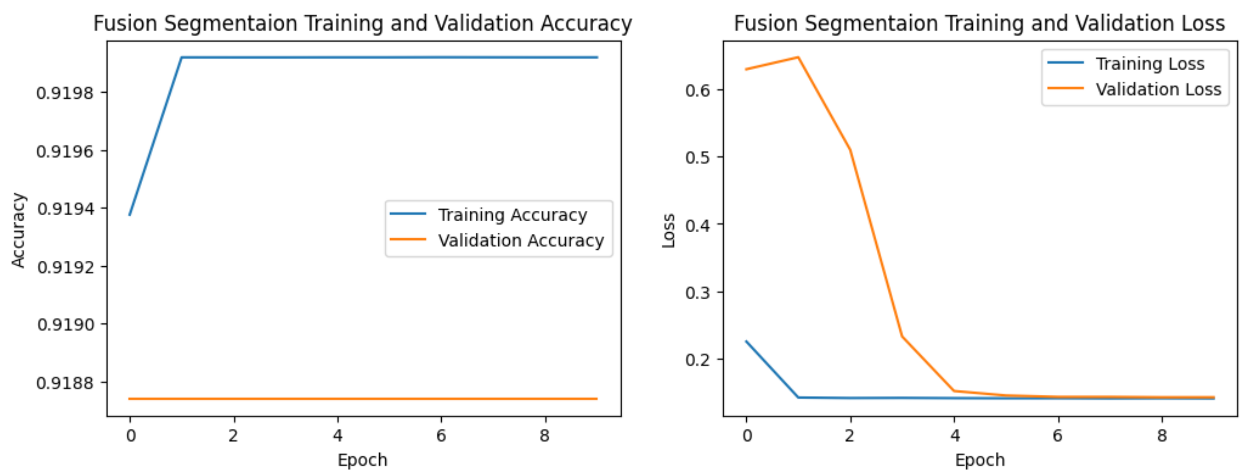
These are model implementations with different preprocessing on Model Fusion and VGG19



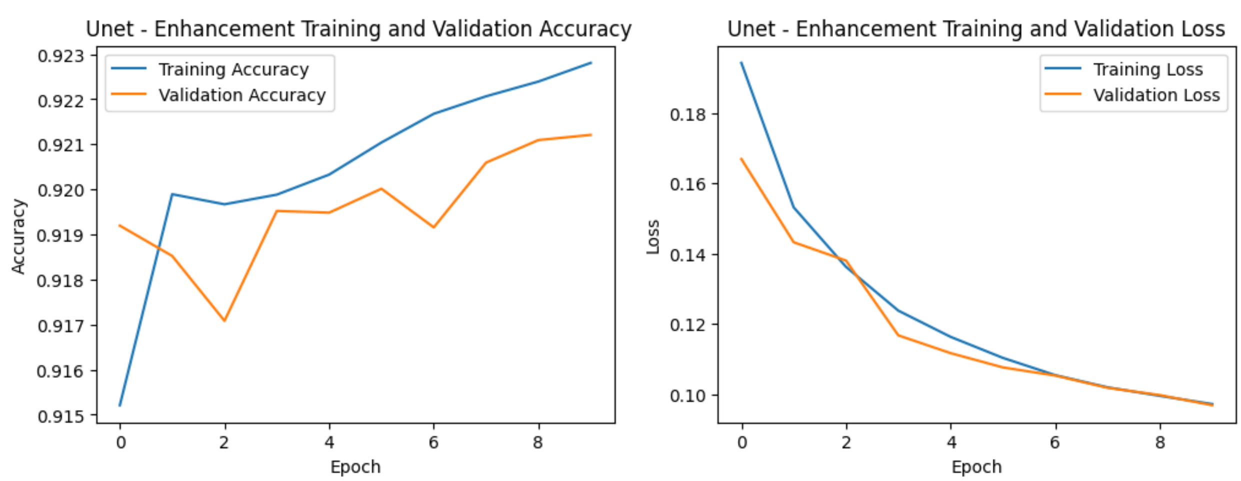
This image indicates that Model Fusion provides an accuracy of 92% on the training dataset and 91.87% on validation for Image Enhancement as preprocessing.



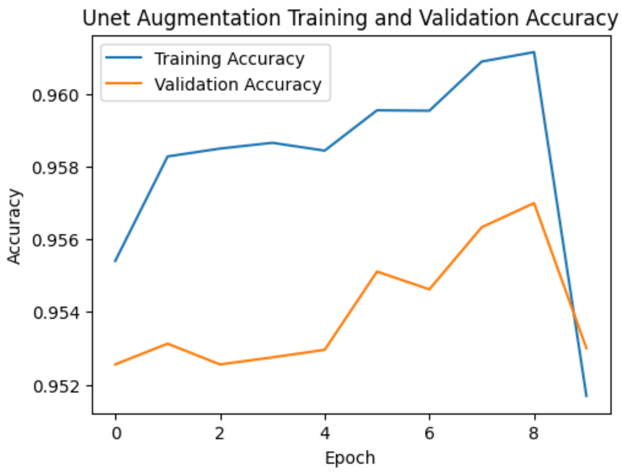
This image indicates that model fusion gives 95.8% training accuracy and 95.3% validation accuracy, This is done on Image augmentation as a preprocessing technique and we received better results than image enhancement.

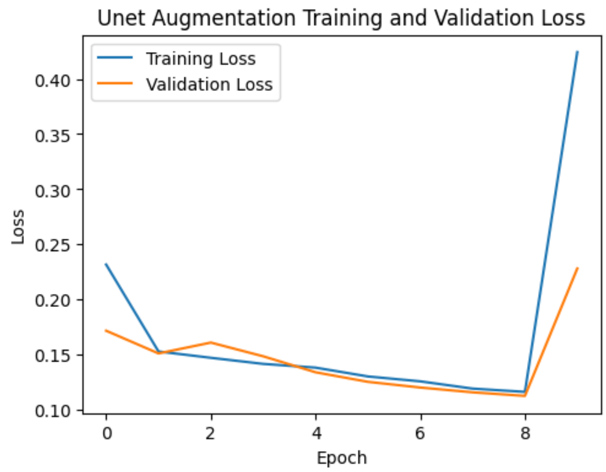


We can see with Image segmentation as a preprocessing technique we get 91.99% accuracy on the training dataset and 91.88% accuracy validation dataset.

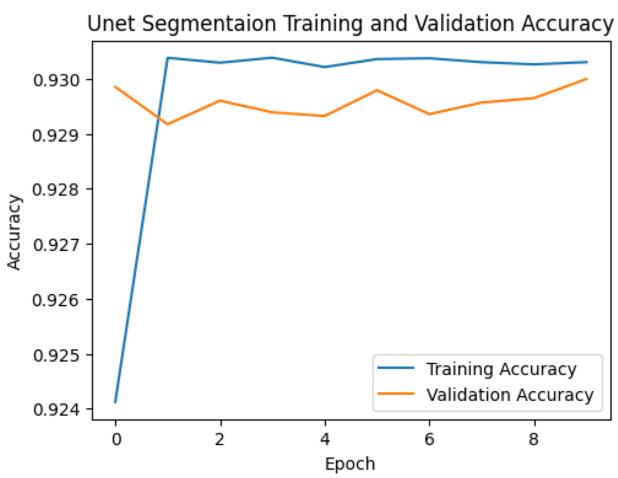


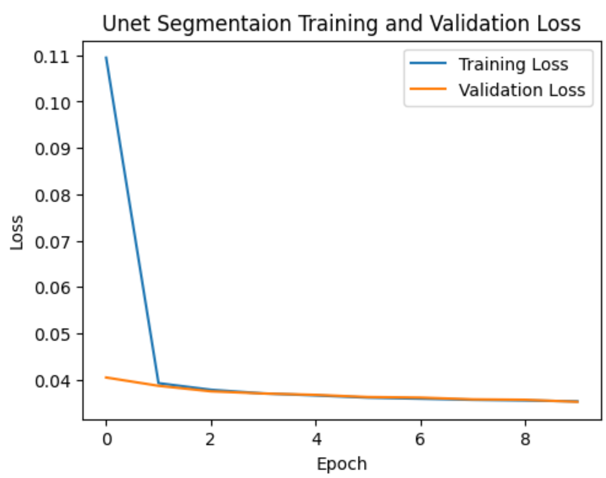
We have tried image enhancement as a preprocessing technique and passed the output data to the Unet model, we received 92.3% on training accuracy and 92.1% on validation accuracy.





In this image data augmentation is implemented and passed to Unet then we got a training accuracy of 95.2% and a validation accuracy of 95.3%. Which is better than image enhancement but in the last 2 epochs it decreased by over 0.8% and loss increased a lot.





After applying image segmentation and passing it to Unet we received the following results, training accuracy of 93.00% and validation accuracy of 92.98%.

Analysis

In the analysis of the algorithm we got to know that Unet is working better even after post-processing even though we tried using model fusion which consists of resnet and VGG. That is performing better till the model training part but after postprocessing, it is giving blank images.

Discussion and Lessons Learned

We have tried a new approach to solving a problem by implementing a model fusion along with VGG to compare the results, It turns out that model fusion is working great until the point of post-processing because for post-processing we are getting blank images for model fusion while correct images for Unet Learning the usage of preprocessing and post-processing techniques from a user point of view does not do much but from a model point of view, it is a great help to understand the data more efficiently and train the model on that data. Hence even though there might be little improvement in the model we will implement it in the model.

Here we learned that model fusion is working great till the model training ad on the training dataset. Still, it does not provide correct output after the post-processing methods like edge detection, morphological operations, and conditional random fields. And Unet is providing better output compared to model fusion on the testing dataset.

Computational resources required for running and implementing the codes are very high hence it is taking a lot of time to run. There are a lot of implementations like every model requiring correct data as input since it trains on that.

Bibliography

Paper 1:
Demir, I., Koperski, K., Lindenbaum, D., Pang, G., Huang, J., Basu, S., ... & Raskar, R. (2018). Deepglobe 2018: A challenge to parse the Earth through satellite images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (pp. 172-181).
Paper 2:
Shen, W., Chen, Y., Cheng, Y., Yang, K., Guo, X., Sun, Y., & Chen, Y. (2021, July). An improved deep-learning model for road extraction from very-high-resolution remote sensing images. In *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS* (pp. 4660-4663). IEEE.

References:

https://medium.com/
https://www.analyticsvidhya.com/
Model Fusion:
https://datascience.stackexchange.com/questions/86769/error-after-merging-two-deep-learning-models-vgg16-and-resnet50
https://www.geeksforgeeks.org/residual-networks-resnet-deep-learning/
Unet:
https://www.kaggle.com/code/devesudev/mini-project-road-map-from-satellite-images
https://www.kaggle.com/code/ekremzturk/satellite-image-road-segmentation
https://www.geeksforgeeks.org/u-net-architecture-explained/
https://discuss.pytorch.org/t/u-net-output-has-256-channels/161949
Segmentation:
https://realpython.com/python-opencv-color-spaces/