

# 1 Extract road features from Satellite

The aim is to identify road features from satellite imagery using the Massachusetts Roads Dataset [1], here I will be using keras and tensorflow for training model, have chosen Unet [2] segmentation model architecture for detecting the road features, here I have used the efficientnetb5 as the backbone for Unet (upon multiple experiments efficientnet gave the best result), also using features like EarlyStopping, ReduceLROnPlateau and ModelCheckpoint.

## 1.1 Dataset Preparation

After downloading and unzipping the dataset it was observed that the number of training input images were more than the number of training output images, hence had to delete the training input images where corresponding output image was missing. After this process had 804 training input and output images. Created the DatagenaratorF class to read input and output images ,resize, make them into batches and float 32 values. Here the image is resized to 512\*512 and 4 was chosen as the batch size. the training dataset is split to training and validation, keeping 754 images for training and 50 images for validation.

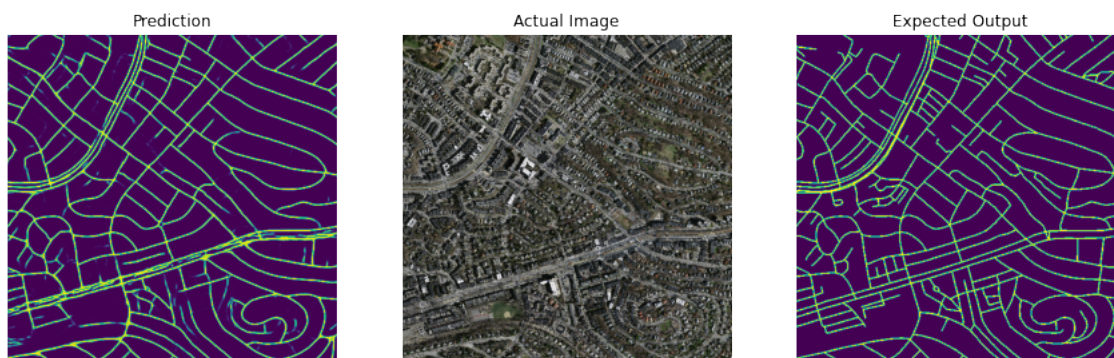
## 1.2 Model Preparation and Training

I am using segmentation modules package [3] which has a built-in Unet model implemented in Keras, here have selected **efficientnetb5** as the backbone and training for 40 epochs with early stopping enabled. choosing Adam optimizer, jaccard loss as the loss function and Intersection over Union(IoU) as the evaluation metric. ReduceLROnPlateau is used to reduce the learning rate by a factor of 10 if validation iou score doesn't change for 10 epochs.

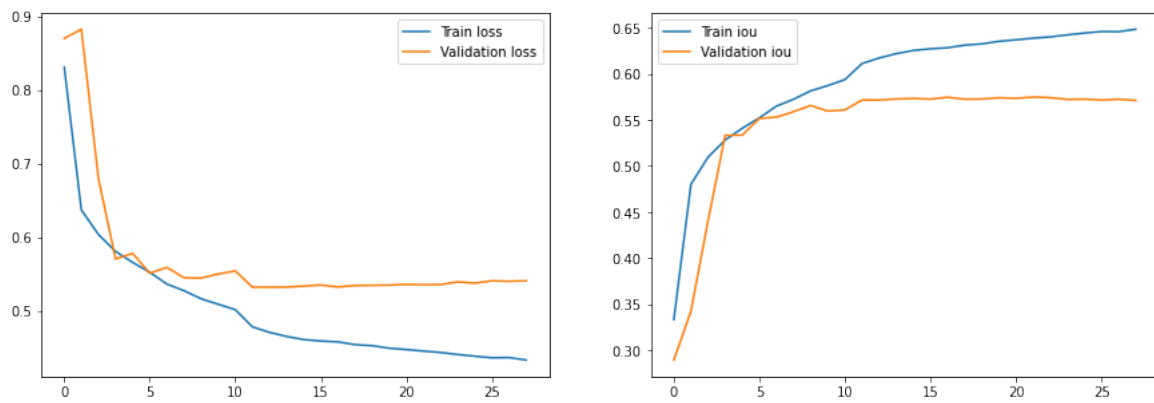
ModelCheckpoint is used to save the model weights evaluated against validation iou score.

## 1.3 Performance of Test Dataset

- the Iou score on test image is **0.54761**
- the Accuracy score on test image is **0.9628**



The prediction, actual and expected images



Viewing the training and validation loss and iou values over epochs

## 1.4 Further improvements on model score

I believe the model accuracy (iou score) can be further enhanced by below suggested methods

- Create more training images by augmenting the train dataset, by flipping, cropping, change the brightness etc. of the images
- use a larger resolution of image for training perhaps 768\*768 or 1024\*1024.
- Run for more epochs. Didn't try the above things as I have constraint on RAM and GPU memory, and also the training time would be very high