Report for Semantic Segmentation using SegFormer and CMP Facade Database

Abstract:

We propose a various approaches for semantic segmentation for given model, weights file and Façade database. For given problem statement we use same provide from github. We have used different Loss functions such as BCE, CrossEntropy Loss and L1 Loss for finetuning Auxiliary Network. These both we have use to fine tune and saved embeddings for 3 classes of Façade database. Also we Finetune model in such a way that we have 3 Architecture Encoder-Decoder and Auxiliary Net, where we have taken weights from provided files only for Encoder part. We are Fine tuning for other Networks.

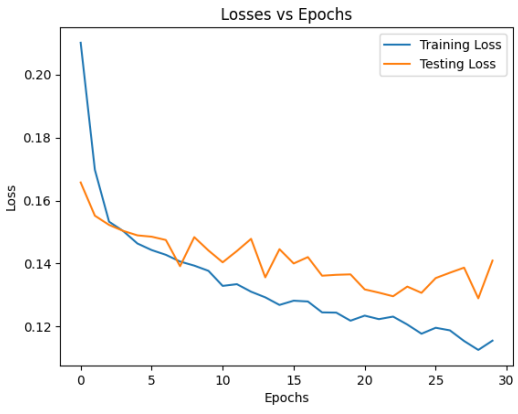
Implementations:

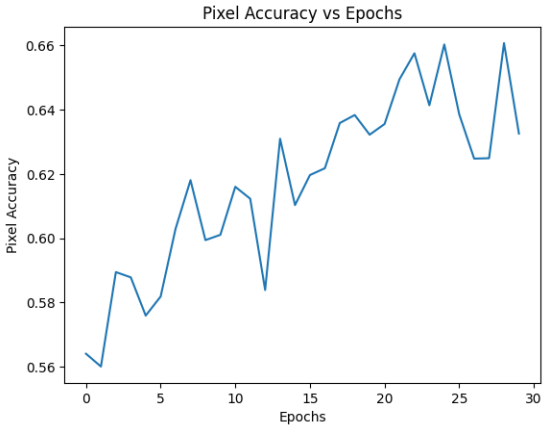
We have experimented with 2 different approaches which we differentiate in **E1, E2**. Now will see broadly about all experiments.

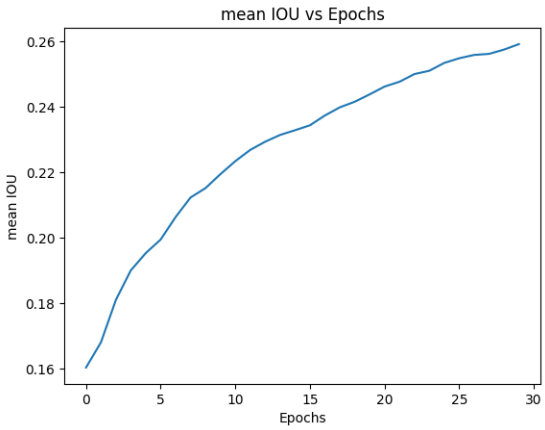
E1 – In this experiment we used BCE loss and L1 Loss for training the pretrain sequence of 194 classes. Here we also used Augmentations from Albumentations.ai where we used some specified methods to augment the image and mask itself i.e., ShiftScaleRotate, channel Shifting, Brightness, Contrast, Random Rotation, Image Transpose, etc for better learning data. We have trained for 30 epochs due to limited resources (even on Google collab). We have mapped Training loss, Testing loss, Mean IOU and Pixel Accuracy in below Fig .We have taken Adam optimizer with default hyperparameters.

Also, we have trained the Auxiliary Network which was updating its weights for 194 classes of which 3 classes from Façade database were common. So technically, those embeddings were also updated for that class and others would update in class 0 i.e., Class Unknown.

Lastly, we have observed good declination of Losses, which is 0.11 for training, 0.14 for testing. Also, we got 0.63 pixel accuracy and 0.26 Mean IOU at 30th Epoch.



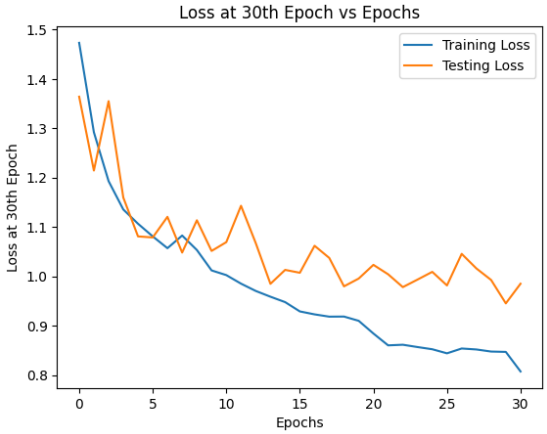


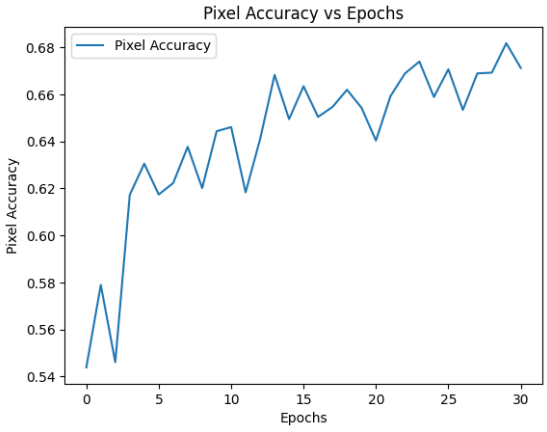


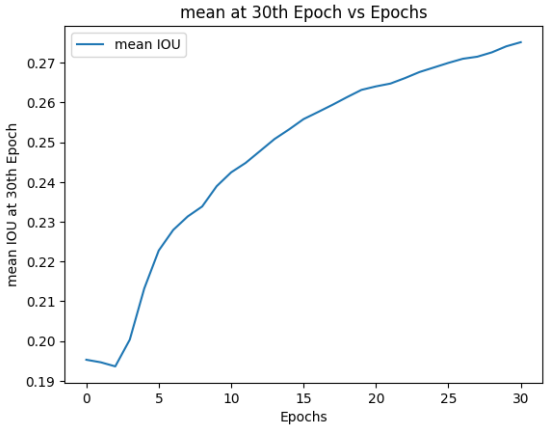
E2– In this experiment, We have train same model but different loss function i.e., CrossEntropy and L1 loss where we didn't require to make OneHot Encoding for label mask for training/testing purpose. The same augmentations techniques with some minute parameters tunning were performed for experimenting purpose. Here we have trained for 40 epochs but for comparing purpose will do till 30th epoch, the reason to train more was to observe the behavior for further plots. We have taken Adam optimizer with default hyperparameters.

Here also Auxiliary Network has trained as same concept in above method mention.

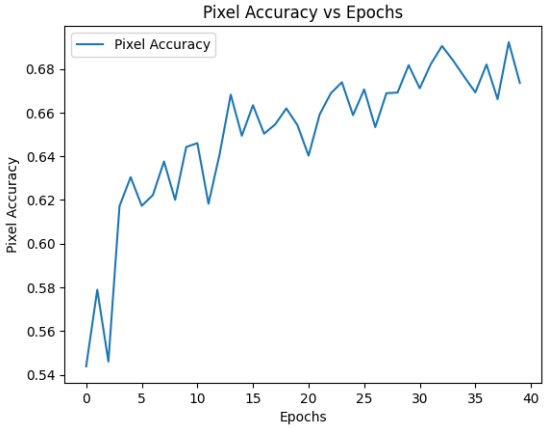
Lastly, we have observed good declination of Losses, which is 0.8 for training, 0.94 for testing. Also, we got 0.67 pixel accuracy and 0.278 Mean IOU at 30th Epoch.

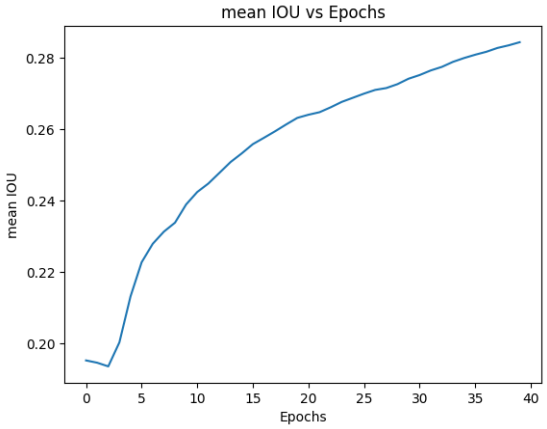






For further training we have observed improvements in meanIOU and pixel accuracy metrics.





Where pixel accuracy has reached 0.685 and meanIOU till approximately 0.29.

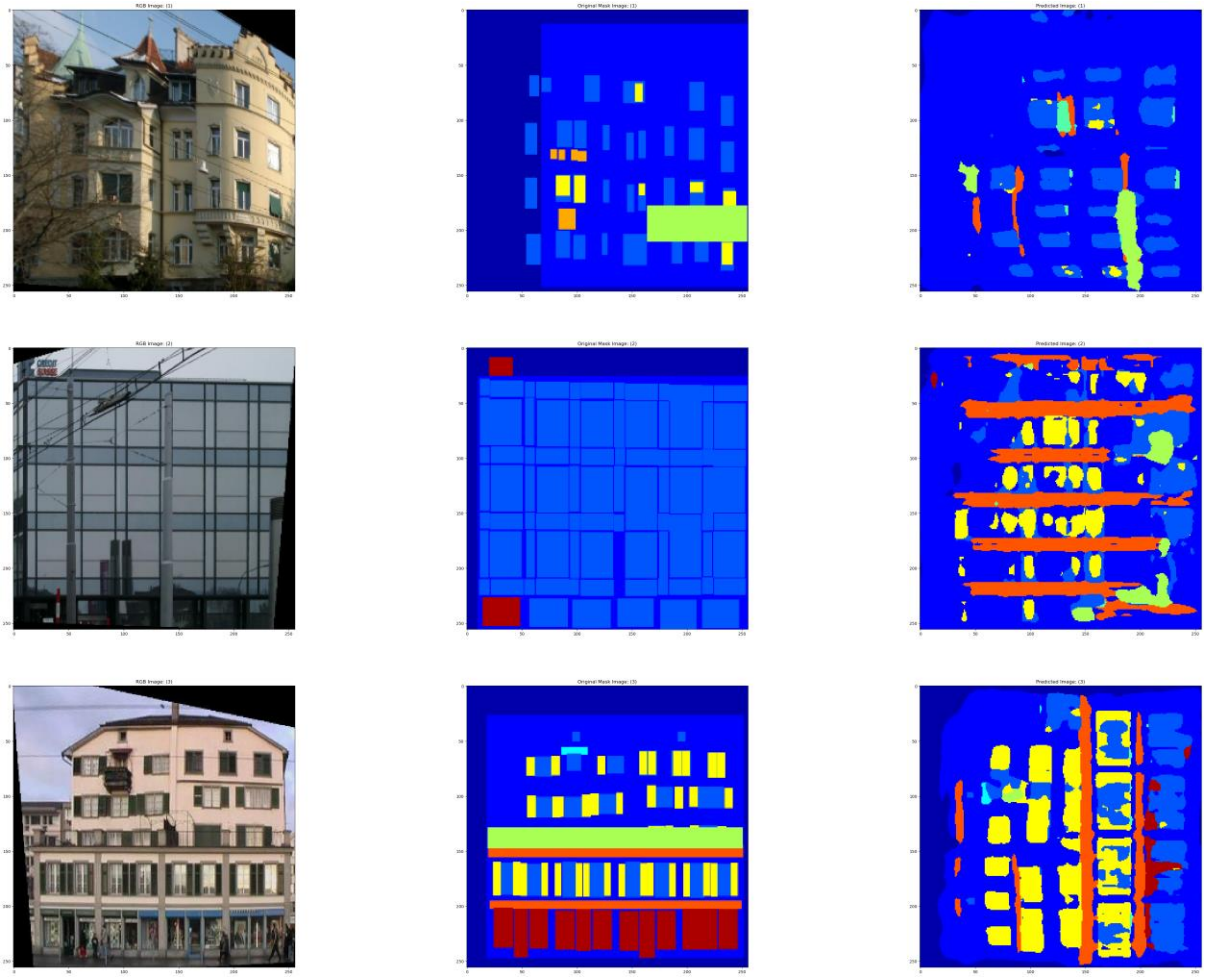
	E1	E2
TrainLoss	0.11	0.8
TestLoss	0.14	0.94
Mean IOU	0.262	0.278
PixelAccuracy	0.63	0.63

Comparison E1 vs E2

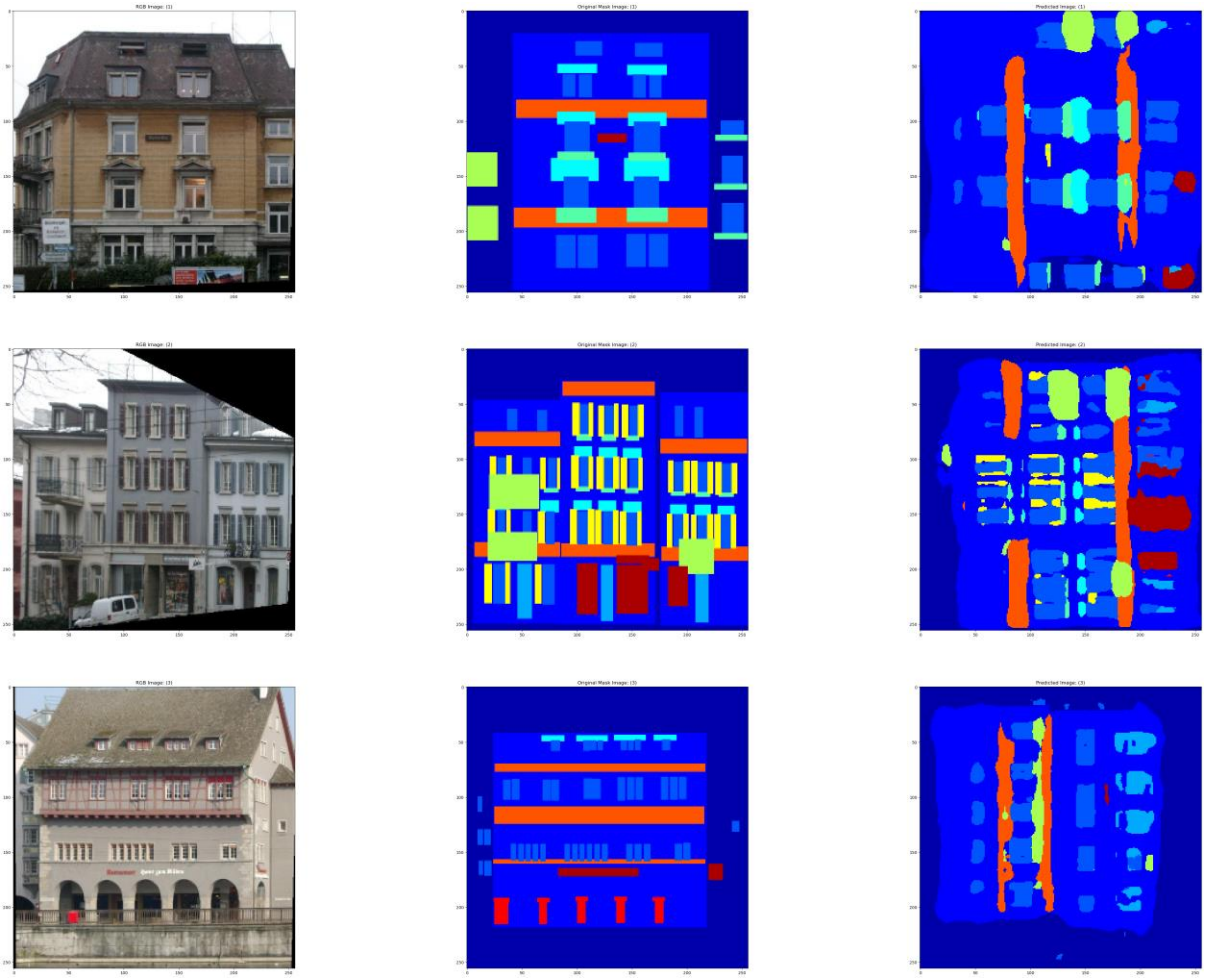
From above comparison table it is clear that meanIOU is greater for E2 than E1 even there is loss difference. We would state there can be a lot of improvements that can be dome in E2 to reduce Losses.

After this section will see Model outputs figure and there comparison.

Image Ground Predicted



E1 OUPUTS



E2 OUPUTS

Conclusion – We have two experiments with different Loss functions where we have brought descend meanIOU and pixel accuracy. We have seen smooth growth in the mean IOU score. Some of the improvements can be Stable training losses, training loss reduction at E2 experiment, learning rate can be slow for some starting epoch and then could be increase, addition of Penalty for decreasing gap between Training loss and Testing loss, greater training time would give a better result than this executed experiments.