Semantic Segmentation using CNN architectures

Huma Ameer– Department of Computer Science, SEECS, NUST

***Abstract*— Semantic image segmentation is the task of classifying each pixel in an image from a predefined set of classes. Semantic segmentation is different from object detection as it does not predict any bounding boxes around the objects. We do not distinguish between different instances of the same object. Multiple techniques and models have been used to for the task of semantic segmentation. Pretrained image segmentation techniques have enabled the researchers to use the weights of pretrained models to train their models more efficiently. This paper shows the implementation of semantic segmentation in VGG and ResNet5 models are being used as a backbone and UNET is the baseline network. These models are trained on modified City Space dataset. The data set was divided into train, validation and test sets. Each model was trained using the same data. The final training accuracy of Vgg_unet model was 76% and the validation accuracy was 80% where as for the Resnet50_unet the model gives a training accuracy of 86% and validation accuracy of 88% respectively.**

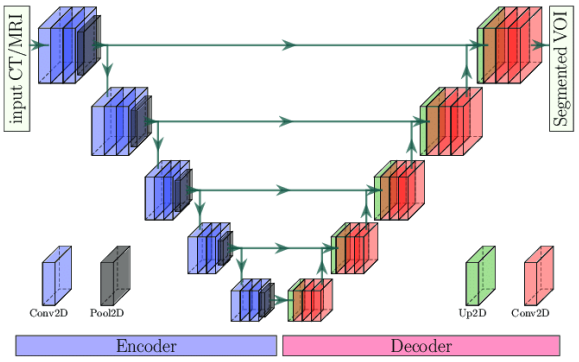
Keywords—semantic segmentation, CNNs, image classification, VGG, ResNet

I. INTRODUCTION

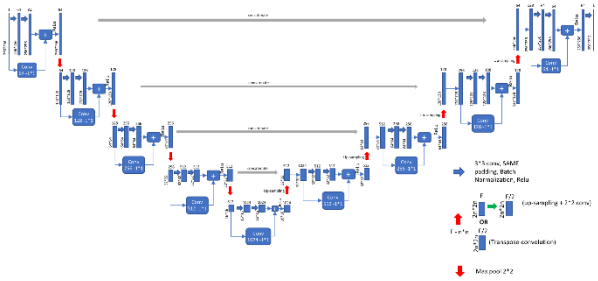
Techniques for image classification, data labeling and semantic segmentation have changed a lot with the pace of time. Over the period of time, due to advancements in the domain of technology there was an urge to learn representations in images at a very fine grain level. For this purpose, many images segmentation tasks and challenges were presented. In this paper, a modified version of the city space data set was used. The data set consisted of prepped train and test images along with annotated train and test images. There was a total of 12 classes in the given dataset. The training set was divided into train and validation with a split of .80 and .20. For the purpose of semantic segmentation **option 2.3** was implemented i.e we changed the backbones and the baseline were same. For the back bones we used VGG and Resnet where as for the baseline architecture we made use of UNet. With VGG_Unet we get a training accuracy of 76% and validation accuracy of 80% where as in Resnet_Unet we get training accuracy of 86% and validation accuracy of 88%. With changing the parameters for both models and with further we can achieve better accuracy for both models thus, yielding better results.

II. METHODOLOGY

In this paper, 2 different models were used for the semantic segmentation i.e. VGG concatenated with UNET and ResNet concatenated with UNET. The models were trained with different parameters ranging from number of epochs, their batch size, the validation batch size, number of steps per epoch and number of validation steps per epoch. The backbone network architecture along with modified architecture is given below.



Vgg_Unet Architecture - Figure 1



Resnet50_Unet Architecture - Figure 2

III. RESULTS

Results of the models are as follows.

A. *VGG_UNET*

In this model we used VGG 16 along with UNet. Model is trained using the following parameters

```
r=model.train(
    train_images =  "/content/A3_Dataset/images_train",
    train_annotations = "/content/A3_Dataset/annotations_train",
    input_height=384,
    input_width=480,
    n_classes=12,
    do_augment=True,
    val_images="/content/A3_Dataset/images_val/",
    val_annotations="/content/A3_Dataset/annotations_val/",
    augmentation_name="aug_all",
    epochs=50,
    validate=True,
    batch_size=2,
    val_batch_size=2,
    steps_per_epoch=146,
    val_steps_per_epoch=37,
)
```

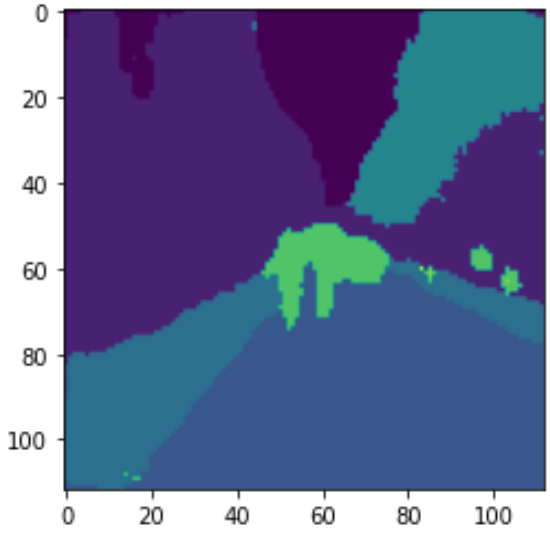
Dataset training and validation accuracy is 76% and 80% respectively.

The IOU Score for this model is given as follows

frequency_weighted_IW: 0.7005424464727552

mean_IU: 0.3445252628423781

class_wise_IU:array([8.95809997e-01, 6.95443542e-01,4.83345198e-03, 8.87583331e-01,3.28655930e-01, 6.57959667e-05,0.00000000e+00, .85571827e-03])



Vgg_Unet segmented image - Figure 3

Vgg_Unet superimposed image - Figure 4

B. Resnet_UNET

In this model we used Resnet along with UNet. Model is trained using the following parameters

```
r=model.train(
    train_images =  "/content/A3_Dataset/images_train",
    train_annotations = "/content/A3_Dataset/annotations_train",
    input_height=384,
    input_width=480,
    n_classes=12,
    do_augment=True,
    val_images="/content/A3_Dataset/images_val/",
    val_annotations="/content/A3_Dataset/annotations_val/",
    augmentation_name="aug_all",
    epochs=50,
    validate=True,
    batch_size=2,
    val_batch_size=2,
    steps_per_epoch=146,
    val_steps_per_epoch=37,
)
```

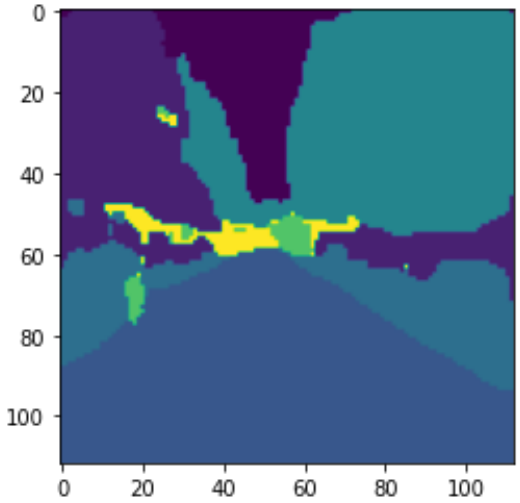
Dataset training and validation accuracy is 86% and 88% respectively.

The IOU Score for this model is given as follows

frequency_weighted_IW: 0.8000537126814479

mean_IU: 0.523038136702496

class_wise_IU: array ([0.92538519, 0.8375725 ,0.00212164,0.91047111,0.75586141, 0.83071235, 0.24583975, 0., 0.81904609, 0.25258023, 0.47300655, 0.22386081])



Resnet_Unet segmented image - Figure 5

Resnet_Unet superimposed image - Figure 6

IV. Discussion and Conclusion

Two models were implemented for the image segmentation task. Both the models had the same baseline but different backbones. One model was VGG16 encoded UNet while the other was Resnet50 encoded UNet. It can be concluded that Restnet UNet has better performance than VGG UNet.

V. Github link and references

https://github.com/hameer10/assignment_3

https://github.com/divamgupta/image-segmentation-keras