Semantic Segmentation: CNN vs Random Forest

Shree Murthy

2024-02-19

Table of Contents

1. Introduction

Semantic segmentation involves classifying every pixel in an image to a specific class. For this report, I used the `CamSeq07` model which contains 202 images and masks to learn more about semantic segmentation. The goal of this project is to analyze how well a CNN and Random Forest model can perform on semantic segmentation. The CNN model was trained using the `U-Net` architecture and the Random Forest model was trained using the `RandomForestClassifier` from the `sklearn` library. Before I created the CNN architecture, I used the Random Forest model to decide if deep learning was required.

1.1 Random Forest Analysis

For the Random Forest (RF) model, I created a 70/30 train-test split and trained the model. Ever image and mask was paired together and pixels were flattened to be fed into the model. The model was trained on two estimators and a max_depth of three. When I initally started the training process, I set the estimators to ten and had no max_depth. This took the model 15 hours to train. I reduced the estimators drastically and added a max_depth to reduce the training time. This reduced the training time to 2.77 hours (9880 seconds). To me this was unaccepatable even though it makes sense. An RF model has data fed in one at a time and every pixel is an independent feature. However, the model's accuracy and IoU scores were good:

Train/Test	Accuracy	IoU Score
Train Data	0.8576	0.9172
Test Data	0.8295	0.9140

For both datasets, the IoU scores were above 0.9. This was a good sign that the model was able to classify the pixels well. While the accuracy is lower, those numbers are just provided to show that the model is not overfitting. The IoU score is the most important metric for semantic segmentation. Below is an example output of the model:

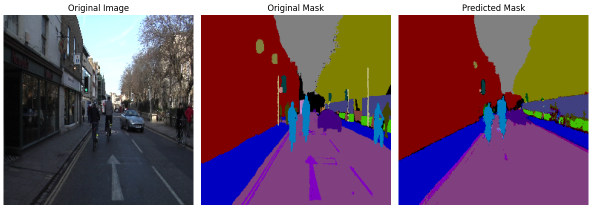


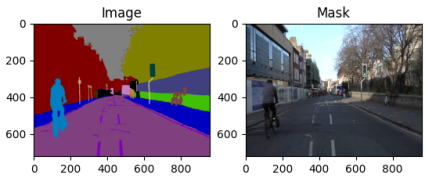
Figure 1: Random Forest Example

The predicted mask shows that the model was able to identify every section of the image. However, the segmentations aren't defined. For instance, the car is not segmented properly to showcase the entire shape of the car and the segmentation is not clear. The entire predicted mask has noise and is not smooth compared to the actual mask. This is due to the pixels being fed in without spatial relationships.

Thus, coupling the training time and the output analysis, using an RF model was not sufficient. I would rather use a CNN model to train faster to leverage faster analysis and better spatial understanding of the image.

2. Exploratory Data Analysis

There are 202 images and masks (101 each). Each image and mask are 720x960x3. The images with '(name)_L.png' are the masks. Furthermore, there is a `label_colors.txt` file that contains all the color codes for the various segments. There are 32 classes within the dataset: such as, buildings, trees, bicyclists, and etc. Since the `CamSeq07` dataset is derived from a video, the images are not independent of each other. To ensure no overfitting occurs, I will have to shuffle the pairs before splitting them. Below is a sample image and mask:



The masks are smooth and segments everything visible

within the original image. This is good to see because this will be needed to ensure the model is trained properly on all the visible classes. Overall, the images will have to be resized to ensure model is not overfitting and can train faster. Also, the masks will have to be encoded with the proper classes such that we can `argmax` the output to get the predicted mask.