Semantic Segmentation for Self-driving Cars using U-Net

Prasad Naik
Stevens Institute of Technology
pnaik3@stevens.edu

Abstract

This paper implements a U-Net based semantic segmentation system that segments images via CARLA (Car Learning to Act) self-driving simulator. The implementation includes designing a pipeline that semantically segments a scene captured in the simulator into 13 different classes. The performance of the model is evaluated using accuracy as the metric.

GitHub: https://github.com/naik24/Semantic-Segmentation-for-Self-Driving-Cars

1 Introduction

Over the past few years, deep convolutional networks have surpassed the current standards in numerous visual recognition tasks. Despite the long-standing existence of convolutional networks, their effectiveness was previously constrained by the limited size of training sets and network dimensions under consideration. Conventionally, convolutional networks have been employed primarily for classification tasks, assigning a single class label as the output for an image. However, in visual tasks, the sought-after output should encompass localization as well.

Semantic segmentation, also known as visual scene comprehension, involves attributing specific semantic labels to individual pixels within an image. Once completed, this procedure transforms the input image into a raster map. Semantic segmentation shares close ties with tasks like image classification, object detection, instance segmentation, and panoptic segmentation, all of which are widely used in the field of computer vision.

Semantic image segmentation has multiple applications, such as detecting road signs (Maldonado-Bascon et al. 2007), colon crypts segmentation (Cohen et al. 2015), land use and land cover classification (Huang et al. 2002). Also, it is widely used in medicine, such as detecting brains and tumors (Wei et al. 1997). Several applications of segmentation in medicine are listed in Dzung et al. (1999). In Advanced Driver Assistance Systems (ADAS) or self-driving area, scene parsing is of great significance, and it heavily relies on semantic image segmentation.

This paper aims to implement a U-Net based semantic segmentation system to classify an image into different classes.

2 Method

The following U-Net architecture is implemented in this paper.

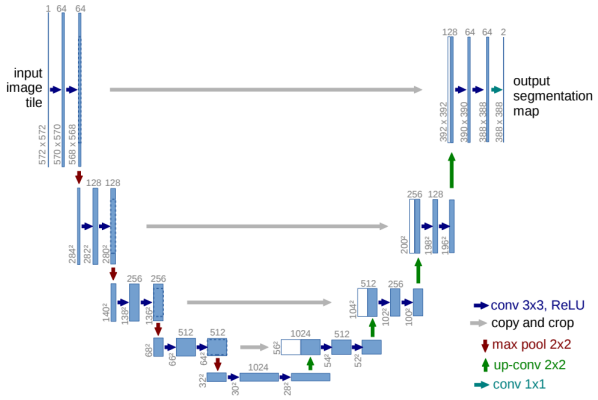

Figure 1 U-Net Architecture

The fundamental concept involves enhancing a typical contracting network by incorporating consecutive layers where up-sampling operators replace pooling operators. Consequently, these layers augment the output resolution. To pinpoint locations, the high-resolution features from the contracting path are merged with the up-sampled output. Subsequently, a sequential convolution layer is capable of refining the output more precisely based on this amalgamated information.

The up-sampling component boasts a substantial number of feature channels, enabling the network to disseminate contextual information to high resolution layers. Consequently, the expansive path exhibits a more or less symmetrical structure to the contracting path, resulting in a u-shaped architecture. Notably, the network excludes fully connected layers and solely utilizes the valid portion of each convolution, ensuring that the segmentation map only includes pixels with complete contextual information available in the input image.

The U-Net consists of four encoder blocks and four corresponding decoder blocks along with a bottleneck layer at the bottom.

3 Data

The dataset used in this paper was provided as a part of the Lyft Udacity Challenge. This dataset provides data images and labeled semantic segmentations captured via CARLA self-driving car simulator. An example of the input image and its segmented output is shown below.

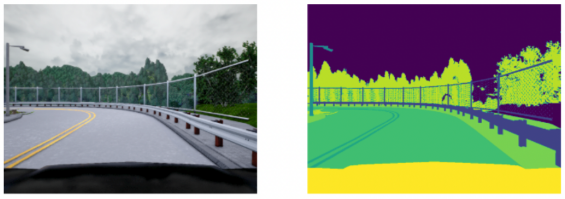

Figure 2 Example Input and Output

The dataset has 5 sets of 1000 images and corresponding labels, totaling 5000 images and their corresponding segmentation masks. Out of this, 3200 images are used for training the U-Net, 800 images for validation and 1000 images for testing. Accuracy is the metric used to determine the generalization of the model on different scenes capture from the CARLA driving simulator. Link for the data is: https://www.kaggle.com/datasets/kumaresanmanickavelu/lyft-udacity-challenge

4 Tools & Technologies

The entire project pipeline is implemented in Python programming language. The glob library (https://docs.python.org/3/library/glob.html) is used to extract data files from the sets of images and their segmentation masks. NumPy (https://numpy.org/) library is used to manipulate high-dimensional arrays. The Tensorflow (https://www.tensorflow.org/) library is used to prepare the data for the model. The Keras (https://keras.io/) framework is used to build, compile, train, and evaluate the model. The Matplotlib (https://matplotlib.org/) library is used to display images.

The entire project pipeline is implemented in Google Colab with a Tesla V100 GPU (https://www.nvidia.com/en-gb/data-center/tesla-v100/).

5 Experiments

The model is trained using three different optimizers – Adam, Stochastic Gradient (SGD), and AdaGrad. The number of epochs initially was same for all the optimizers. However, after training, it was observed that Adam requires the lowest number of epochs compared to SGD and AdaGrad to converge while AdaGrad requires the highest number of epochs in comparison. The learning rate of

0.01 was kept identical in all the cases. The sparse categorical cross entropy loss function was used to measure the loss.

The details of experiments are tabulated in Table 1.

Table 1 Details of Experiments

Optimizer	LR	Loss	Epochs	Metric
Adam	0.01	Sparse Categorical Cross Entropy	15	Accuracy
SGD	0.01	Sparse Categorical Cross Entropy	20	Accuracy
AdaGrad	0.01	Sparse Categorical Cross Entropy	30	Accuracy

Figure 3, 4, and 5 show the training and validation accuracy and loss of training the U-Net model with Adam, SGD, and AdaGrad optimizer respectively.

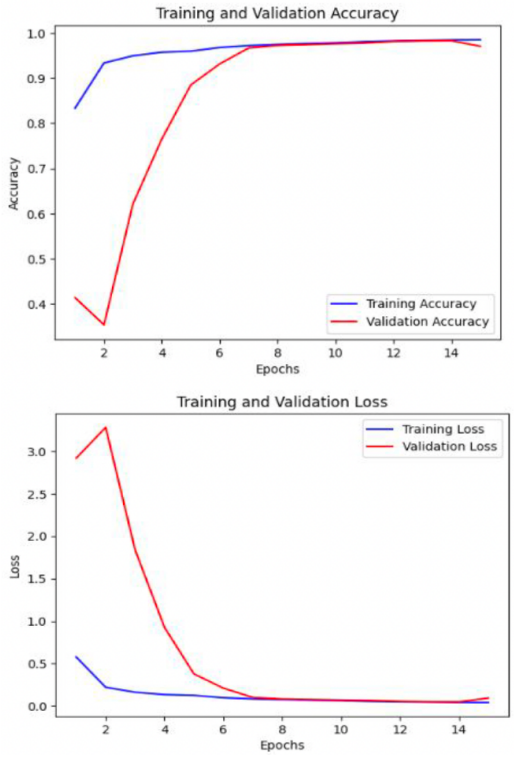

Figure 3 Accuracy and Loss Curves for Adam Optimizer

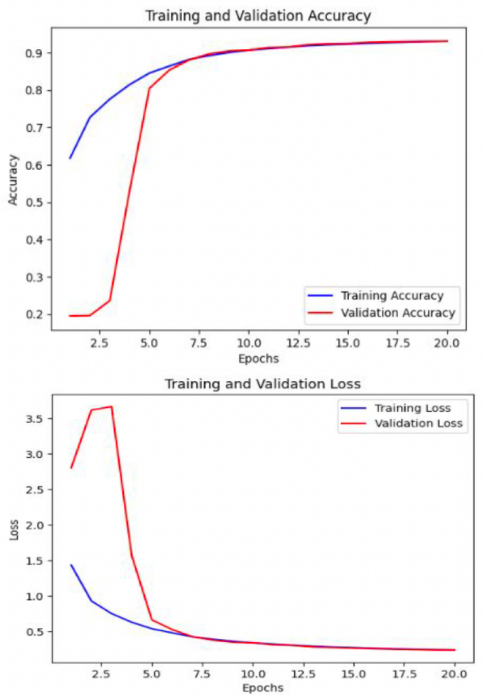

Figure 4 Accuracy and Loss Curves for SGD Optimizer

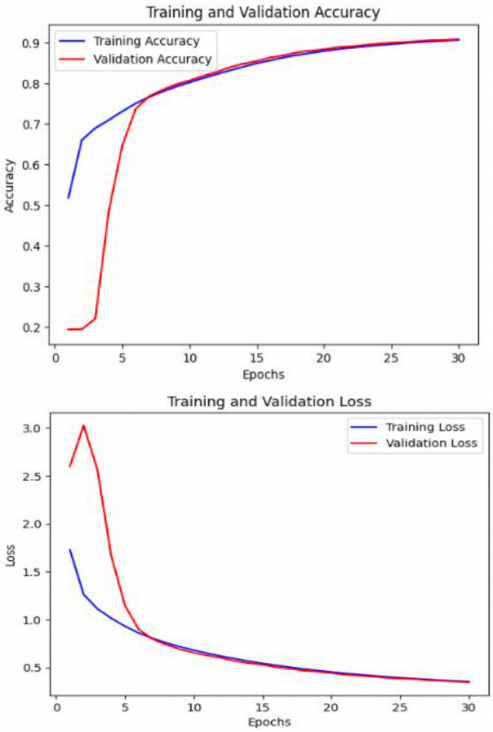

Figure 5 Accuracy and Loss Curves for AdaGrad Optimizer

6 Results

The model is tested using the test dataset. The testing is performed with three models trained using optimizers in Table 1.

Test accuracy and test loss for the optimizers is tabulated in table 2 below.

Table 2 Test Results

Optimizer	Test Accuracy (%)	Test Loss
Train	96.70	0.1074
Validation	93.09	0.2392
Test	90.30	0.3532

Figures 6, 7, and 8 show the results obtained using the three optimizers respectively. The figures show the input image, the true segmentation mask, and the predicted segmentation mask.

Figure 6 Predicted Mask (Adam)

Figure 7 Predicted Mask (SGD)

Figure 8 Predicted Mask (AdaGrad)

7 Conclusion

The U-Net performs very well on the CARLA driving simulator data. Thanks to Udacity and Lyft for providing the data in a structured file format which made it easy to retrieve and preprocess the data.

The model implemented in this paper can be used in not only self-driving applications, but also in biomedical and satellite imagery applications.

8 Future Work

Since the data provided is already processed by CARLA, the code in this paper does not implement processing raw dashcam images. Future work could include capturing raw dashcam images and post-processing these images to perform semantic segmentation.

The code pipeline implemented in this paper could be modified to perform semantic segmentation on videos instead of just images.

References

Ronneberger, O., Fischer, P., Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab, N., Hornegger, J., Wells, W., Frangi, A. (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. MICCAI 2015. Lecture Notes in Computer Science(), vol 9351. Springer, Cham. https://doi.org/10.1007/978-3-319-24574-4_28.

Emek Soylu, B.; Guzel, M.S.; Bostanci, G.E.; Ekinci, F.; Asuroglu, T.; Acici, K. Deep-Learning-Based Approaches for Semantic Segmentation of Natural Scene Images: A Review. Electronics 2023, 12, 2730. https://doi.org/10.3390/electronics12122730.

Liu, X., Deng, Z. & Yang, Y. Recent progress in semantic image segmentation. Artif Intell Rev

52, 1089–1106 (2019). https://doi.org/10.1007/s10462-018-9641-3.

Maldonado-Bascon S, Lafuente-Arroyo S, Gil-Jimenez P, Gomez-Moreno H, López-Ferreras F (2007) Roadsign detection and recognition based on support vector machines. IEEE Trans Intell Transp Syst 8(2):264–278.

Chan, R., Uhlemeyer, S., Rottmann, M., Gottschalk, H. (2022). Detecting and Learning the Unknown in Semantic Segmentation. In: Fingscheidt, T., Gottschalk, H., Houben, S. (eds) Deep Neural Networks and Data for Automated Driving. Springer, Cham. https://doi.org/10.1007/978-3-031-01233-4_10.