# Semantic Segmentation in Autonomous Vehicles

**Suraj Sapkal** [1] **Milan Shah** [1] **Akshata Pore** [1]

## Abstract

In Autonomous Vehicles (as well as in Robotics), it's important for the vehicle to understand the context of its environment e.g. to distinguish between pedestrians, other vehicles, traffic signs, and so on. To accurately discern these different kind of objects, one of the best approach is to use semantic segmentation. In this work, we implemented SegNet and its variants, inspired from U-Net and FDNet.

## 1. Introduction

Nowadays, autonomous vehicles are becoming more and more common and they are believed to be the normal way of transportation in the future. The navigation stack of these vehicles comprise of multiple stacks: perception (computer vision and sensor fusion), localization, mapping, planning, and control. It's the perception stack through which the autonomous vehicles understand their surrounding environment or say it's their eye through which they see the world around them.

Deep learning has recently achieved superior performance on many tasks such as image classification, object detection, and what not (Zhao et al., 2017). It is being heavily used in the perception stack to detect pedestrians, other vehicles, traffic signs, and so on. Though it generates the bounding boxes around multiple obstacles being seen by the vehicle, that information is not an accurate representation of the environment as it does not provide pixel-level information. The solution is semantic segmentation which predicts the class label for each pixel in the image i.e. whether a pixel belongs to person, rider, car, building, pole, and many more.

[1]Worcester Polytechnic Institute. Correspondence to: Suraj, Milan, Akshata <sssapkal@wpi.edu, mrshah@wpi.edu, aspore@wpi.edu>.

*CS/DS 541 Deep Learning Final Project*, Worcester Polytechnic Institute. Copyright 2020 by the author(s).

## 2. Related Work

When it comes to analyzing visual imagery, Convolution Neural Networks (CNNs) are most commonly used mainly because of their ability to learn relevant spatial features in the image. Prior approaches have been made to accomplish semantic segmentation task using the different version of CNNs e.g. in FCN (Long et al., 2015), first, some state of the art CNNs: AlexNet (Krizhevsky et al., 2012), VGGnet (Simonyan & Zisserman, 2014), and GoogLeNet (Szegedy et al., 2015) have been used to extract the low-level local features through supervised pre-training. Now, this output feature map has lower resolution. So, to generate the segmented image having the resolution as same as input image, the feature map is upsampled using simple Bilinear Interpolation.

However, the problem with this approach is that so many spatial information has been lost in the convolution part which can not be completely regained while upsampling. Because of this, the image is not accurately segmented.

To overcome this problem, generally encoder-decoder approach is used (Badrinarayanan et al., 2017). So, in our work, we have implemented that one and to further enhance the accuracy we have also implemented its variants, inspired from U-Net (Ronneberger et al., 2015) FDNet (Zhen et al., 2019).

## 3. Proposed Method

For semantic segmentation of an image, the aim is to represent the image in such a way that the Pixels belonging to the same class have similar intensity values. As mentioned in the previous sections, several architectures have been proposed to solve this problem. The results showed that the encoder-decoder based architecture outperformed the Convolution plus fully connected architecture. One of the reasons could be the way architecture processes the data. In the encoder part, we tried to represent the image in a lower-dimensional space (analogous to principal component analysis for the image). For the task of segmentation, the image shouldn't contain high-frequency data, texture, and morphological details except if it's in the boundary region of the image. In the encoder-decoder architecture, we can account for this boundary information by transferring the in-

formation directly from the encoder layer to the decoder layer. One of the distinguishing factors in the encoder-decoder architecture is how the model handles the process of upsampling. Our work is similar to SegNet. For getting a good understanding of architecture, a discussion on SegNet architecture is provided in the upcoming sections. Later, an explanation of implementing SegNet with Skip connection and SegNet with fully dense layers is described.

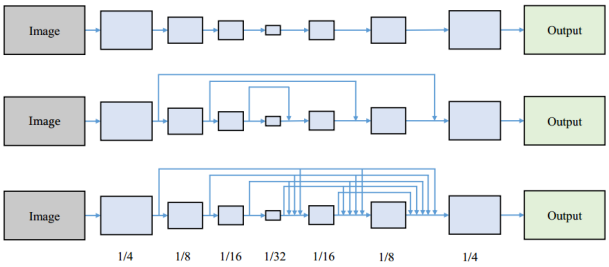


*Figure 1.* Different types of encoder-decoder structures for semantic segmentation. **Top:** basic encoder-decoder structure and SegNet using a multiple-stage decoder to predict masks, often results in very coarse pixel masks since spatial information is largely lost in the encoder module. **Middle:** Feature map reuses structures using previous feature maps of the encoder module, e.g. U-Net, achieves very good results in semantic segmentation tasks, but the potential of feature map reuse is not deeply released. **Bottom:** The fully dense networks, FDNet, using feature maps from all the previous blocks, are capable of capturing multiscale information, of restoring the spatial information, and of benefitting the gradient backpropagation.

### 3.1. SegNet - Vanilla version

The SegNet architecture adopts an encoder-decoder framework which is based on the 13 layers of VGG16 convolution neural network. The encoder and decoder layers are symmetrical and the upsampling operation of the decoder layers uses the max-pooling indices of the corresponding encoder layers. The novelty of this architecture is in the subsampling stage, translation invariance is achieved by Max-pooling. The subsampling and max-pooling also help achieve better classification accuracy but reduce the feature map size, this leads to lossy image representation with blurred boundaries which is not ideal for segmentation purpose.

### 3.2. SegNet - with skip connections

The SegNet architecture didn't perform very well when it comes to boundary delineation. We think one of the prime reasons for this is max-pooling indices as we lose crucial spatial information in the boundary areas. We think using the feature map from the previous layers will add some necessary information to handle the boundary information. Previous work on U-Net (Ronneberger et al., 2015) architecture verifies the points that using the feature maps from the encoder layer improved the segmentation performance significantly. Inspired by this architecture we have proposed the modified version of SegNet where apart from updating the indices of the max-pooling layer we are also using the feature map from the corresponding encoder region to create the feature map on decoder end.

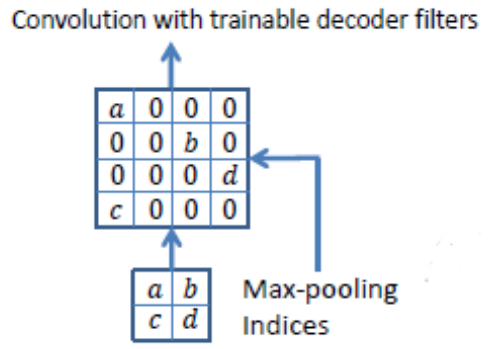


*Figure 2.* Decoding in SegNet - Vanilla version

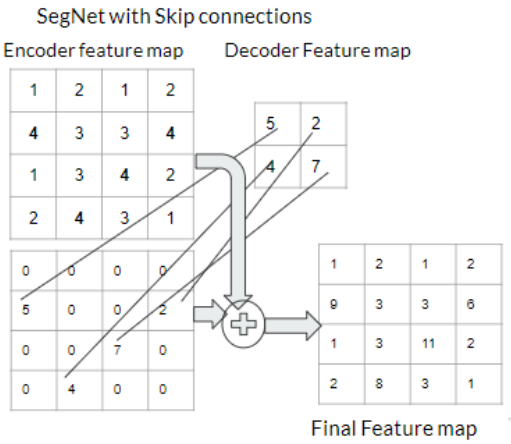


*Figure 3.* Decoding in SegNet with skip connection

### 3.3. SegNet - with Fully connected skip conections

Inspired by the performance improvement brought by the feature map reuse we proposed this architecture based on FDNet (Zhen et al., 2019). In this particular architecture while creating the decoder feature map rather than considering the max-pooling indices and the feature map of the corresponding encoder layer, the feature maps from all the previous layers are taken into account for the creation of decoder feature map. To account for the difference in the size of the feature map we are using transpose convolution and convolution to match the decoder feature map dimensions.

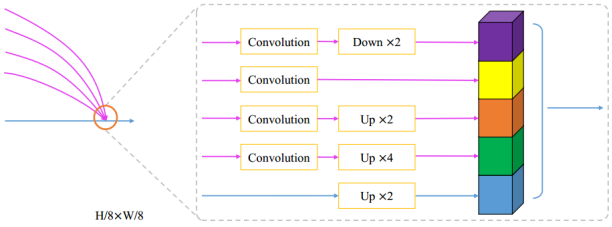


*Figure 4.* Decoding approach in FDNet. Through that, SegNet with Fully connected skip connections has been inspired

## 4. Experiment

### 4.1. Dataset

As our work focuses on autonomous vehicles, to get realistic results, we used Cityscapes (Cordts et al., 2016) dataset. There are many advantages of this dataset over other autonomous vehicle datasets, namely

- The Cityscapes Dataset focuses on semantic understanding of urban street scenes.
- Unlike other datasets, this has been generated by manually driving a vehicle in multiple different cities of Germany (and of few neighbor countries)
- It's very diverse i.e. several months (spring, summer, fall), good/medium weather conditions

### 4.2. Training

To get a good comparative analysis on the performance of all the proposed networks, we trained all the networks on an identical set of hyper-parameters and input data. For training data, we used 1000 images from the cityscape dataset. For tuning the hyper-parameters, we tried out different values of learning rate, batch size and optimization techniques. The final learning rate used during the training was 0.1, 0.01, and 0.001. The batch size was set to 2, 4, 8 and the model used Adam and Stochastic gradient descent for optimization. The training is carried out on a machine with Nvidia gtx1060 GPU and Intel i7 9th generation CPU. The training was carried out for 100 epochs and weights for each epoch was stored. Finally, the weights with minimum cross-entropy loss was considered for the model.

### 4.3. Testing

For testing the performance of the network, we used 500 images from the cityscape dataset. We use a pixel-wise cross-entropy function for determining the testing performance of the model. In order to get better insights on the comparative performance of the models, we tested the model under the same circumstances with cross entropy loss function.

## 5. Results

From the experiments performed described above, some of the results that is final segmented images are presented below. From the results achieved, it was observed that the Segnet with skip connections performed better that the other variations implemented, mainly due to resuse of feature maps from the encoder layers. An accuracy of 72.16 percent was achieved which was better than that of Vanilla Segnet and Fully Dense Segnet. The accuracy of Vanilla Segnet achieved was 60.73 percent whereas that of Segnet with Fully connected layers was achieved to be 71.76 percent which is slightly less than that of Segnet with skip connections.

As seen in the figures 5 and 6, segnet with skip connection was able to identify and segment the yellow taxi seen in the original image but Vanilla segnet failed to do so.

Segnet with Fully Connected Layers 7 gave clear discretisation of the labels.

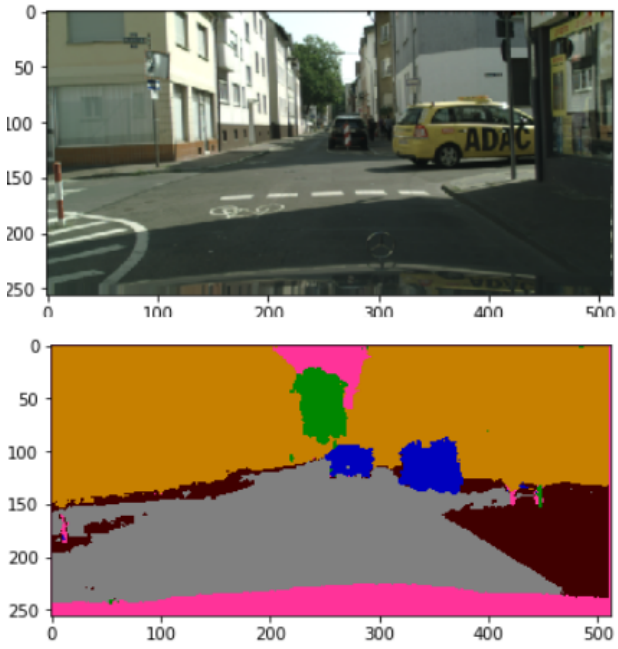


*Figure 5.* Original and Segmented output of Vanilla SegNet

## 6. Discussion

We found that utilizing the previous feature maps as much as possible improves the performance but the amount of improvement was not high enough. Also, it took at least 15 hours to train a single variant for 100 epochs on a local machine having 6 GB Nvidia GPU.

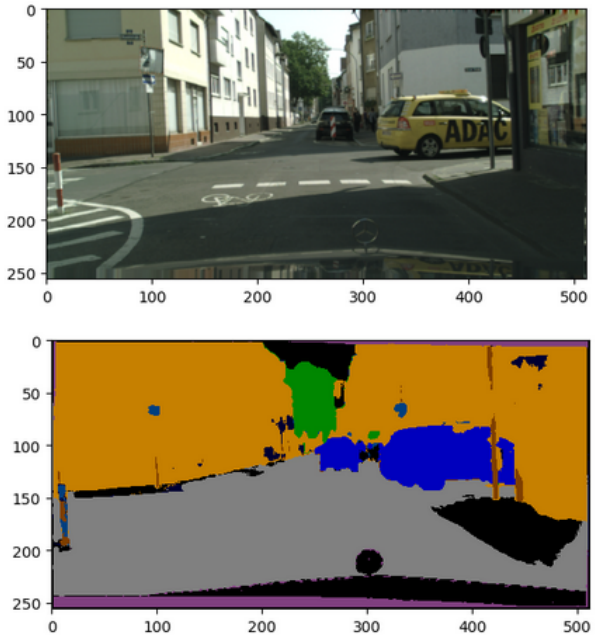

*Figure 6.* Original and Segmented output of SegNet with skip connections

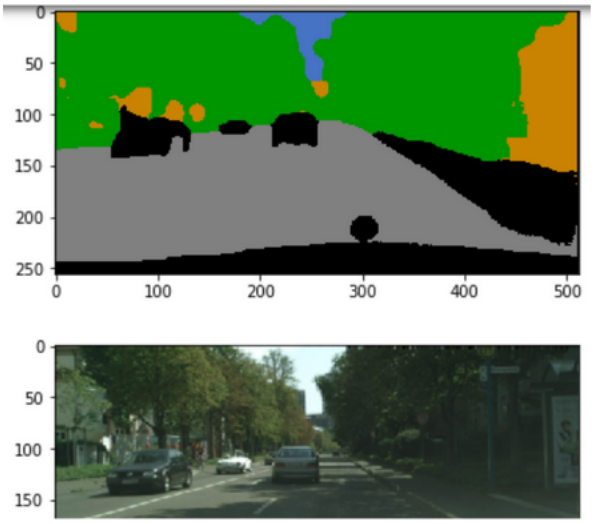

*Figure 7.* Original and Segmented output of SegNet with Fully Connected Layers

## 7. Conclusions and Future Work

As can be seen from the results, we can say that the level at which feature maps from the encoder are being used in the decoder greatly affects the output. Throughout these different approaches, we had kept the hyperparameters e.g. number of epochs, batch size, optimizer, etc same. However, in the future work, they can be explored to understand how do different variants behave for the various combinations of hyperparameters. Also, as mentioned earlier, it took too much time for training. So, another research that can be tackled is how to minimize the training duration.

## References


Badrinarayanan, Vijay, Kendall, Alex, and Cipolla, Roberto. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12): 2481–2495, 2017.

Cordts, Marius, Omran, Mohamed, Ramos, Sebastian, Rehfeld, Timo, Enzweiler, Markus, Benenson, Rodrigo, Franke, Uwe, Roth, Stefan, and Schiele, Bernt. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.

Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Long, Jonathan, Shelhamer, Evan, and Darrell, Trevor. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.

Ronneberger, Olaf, Fischer, Philipp, and Brox, Thomas. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Simonyan, Karen and Zisserman, Andrew. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Szegedy, Christian, Liu, Wei, Jia, Yangqing, Sermanet, Pierre, Reed, Scott, Anguelov, Dragomir, Erhan, Dumitru, Vanhoucke, Vincent, and Rabinovich, Andrew. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

Zhao, Bo, Feng, Jiashi, Wu, Xiao, and Yan, Shuicheng. A survey on deep learning-based fine-grained object classification and semantic segmentation. *International Journal of Automation and Computing*, 14(2):119–135, 2017.

Zhen, Mingmin, Wang, Jinglu, Zhou, Lei, Fang, Tian, and Quan, Long. Learning fully dense neural networks for image semantic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 9283–9290, 2019.