Vision System for Autonomous Vehicles Image Segmentation using Deep Neural Networks

Chhavi Sharma[1]*,Prahasan Gadugu[1]*, Supriya Ayalur Balasubramanian[1]*

Abstract
Autonomous driving could redefine the automotive industry by decreasing traffic and reducing emissions, in turn leading to better health, increasing safety, and saving driving time for humans. With so many benefits, autonomous cars have forever been a lucrative research area. The vision of an autonomous vehicle as in essential and perhaps, one of the most important tasks towards eliminating human involvement in driving. Hence, as part of our Deep Learning final project, we propose to design a vision system for autonomous vehicles that helps semantically understand the foreground objects as viewed by a human driver by coloring them differently. The project extensively utilizes image segmentation techniques using various Deep Neural Network Approaches.

Keywords
Image Segmentation — Deep Learning — Convolutional Neural Networks — Fully Convolutional Networks — FCN8 — U-Net — Semantic segmentation — VGG16

[1] *Data Science, School of Informatics, Computing, and Engineering, Indiana University, Bloomington, IN, USA*
*Course Instructor: Minje Kim

Contents

Introduction 1
1 Related Work 2
2 Data Preprocessing 2
3 Image Segmentation Approach 2
3.1 Fully Convolutional Networks(FCN) 2
3.2 Traditional CNN vs. FCN 2
3.3 Encoder and Decoder Network 3
3.4 Building blocks of Segmentation 3
3.5 Key features of FCN 3
3.6 Evaluation Metrics 3
4 Models and Results 4
4.1 FCN8 - VGG16 4
4.2 U-Net - VGG16 4
5 Summary 4
6 Conclusion and Future Scope 5
Acknowledgments 5
References 5

Introduction

As the world continues to move deliberately toward a transportation system driven by autonomous vehicles, there are several benefits that driverless cars potentially promise. Higher levels of autonomy have the potential to reduce risky and dangerous driver behaviors. The greatest promise may be reducing the devastation of impaired driving, drugged driving, unbelted vehicle occupants, speeding and distraction. To accomplish autonomous driving, one of the main challenges lies in designing its vision system. This project is our tiny attempt in that area. Before proceeding, it is essential to understand and develop a logical approach to distinguish various objects in the foreground while designing the vision system for an autonomous vehicle. This is because a vehicle may come across various objects while driving like lanes, pedestrians, things, other vehicles, background etc. Each of them needs to be shaded with a distinct color to be distinguished. Thus, we can look at this problem which requires image segmentation which needs to be performed alongside semantic segmentation in order to completely understand what is going on in the scene forefront. The approach could progress like:

1. Classification: First, we perform classification to predict different classes for a whole input scene.

2. Localization: Next, we try to extract the spatial location of these classes.

3. Semantic Segmentation: Lastly, we make finer inferences by making dense predictions for every pixel and assigning a class to each pixel such that each class encloses pixels of its object or region.

1. Related Work

Let us review some neural networks that form the base of semantic segmentation systems and play an important role in computer vision:

1. AlexNet: This deep CNN competed in the ImageNet Challenge in 2012 and emerged as a winner with a test accuracy of 84.6%İt contained 5 convolutional layers, some followed by max-pooling layers, and 3 fully connected layers. It used non-saturating ReLU activation function giving an improved training performance.

2. VGG-16: Proposed in an Oxford paper, VGG16 achieves 92.7% top-5 test accuracy in ImageNet. It replaced large kernel-sized filters in AlexNet with multiple 33 kernel-sized filters one after another.

3. GoogLeNet: Emerging as new state-of-art, Google's entry in the ImageNet Challenge 2014, it won with an accuracy of 93.3%Ṫhe network used a CNN with a novel element called inception module. It used batch normalization, image distortions and RMSprop. Their architecture consisted of a 22 layer deep CNN but small convolutions to reduce the number of parameters.

Other popular but less efficient techniques for image segmentation owing to their rigid algorithms and human involvement:

1. Thresholding— The process of creating a black-and-white image out of a grayscale image by setting exactly those pixels to white whose value is above a given threshold, and setting the other pixels to black.

2. K-means clustering— K-Means clustering algorithm is an unsupervised algorithm and it is used to segment the interest area from the background. It clusters or partitions the given data into K-clusters or parts based on the K-centroids.

We have implemented the modern approaches such as FCN architecture with a combination of VGG and U-Net encoder networks the model designs are briefly discussed in the next sections.

2. Data Preprocessing

We are using the Cityscapes dataset[1], with fine annotations which consists of 2975 training, 500 validation, and 1525 testing images. Owing to the size of the image dataset and GPU capabilities required for our project, we have limited our train set to 802, validation set to 267 and a test set to 544 images.

Figure 1 below exhibits the data distribution and the pixel space according to the Cityscapes Data Overview paper:

The aim is to design a deep learning model that trains on foreground scenery and colors the objects to classify them into 4 classes - Road, Vehicle, Human and background(everything except the other 3 classes). The semantic color coding assigned are: **Road - Green, Vehicle - Red, Human - Blue and Background - Yellow**.

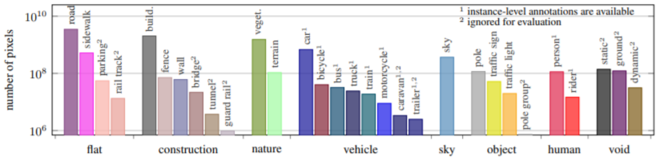

Figure 1. Pixel wise distribution

We converted the RGB model(describes color in terms of amount of Red, Green and Blue present) to HSV(Hue Saturation Value) model since it is a preferred approach in situations where color description plays an integral role. After initializing with random hue and saturation values, we reconverted it to RGB image and normalized it.

3. Image Segmentation Approach

As part of our research, we came across a paper on Fully Convolutional Networks(FCN) in Image Segmentation by Evan Shelhamer Jonathan Long and Trevor Darrell. Thus, we felt that FCN architecture was more relevant to our problem rather than implementing traditional CNN approaches.

3.1 Fully Convolutional Networks(FCN)

FCN is generally used for semantic segmentation. Firstly, it decompresses an image to 1/32th of its original size using convolution and maxpool layers, makes the class prediction at this level and then uses upsampling and deconvolution layers to resize the image to its original size. The final output layer has a large receptive field and corresponds to the height and width of the image, while the number of channels corresponds to the number of classes.

3.2 Traditional CNN vs. FCN

A fully convolutional CNN (FCN) is one where all the learnable layers are convolutional, so it does not have any fully connected layer unlike traditional CNNs that have just those. Since FCNs don't have any fully connected layer in your network, we can apply the network to images of virtually any size because only the fully connected layer expects inputs of a certain size. Moreover, a fully convolutional net tries to learn representations and make decisions based on local spatial input. Appending a fully connected layer enables the network to learn something using global information where the spatial arrangement of the input falls away and need not apply. It is also computationally faster than traditional CNNs.

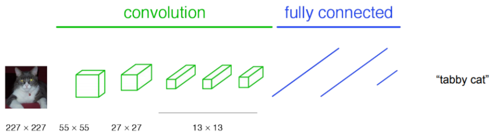

Figure 2. Traditional CNN

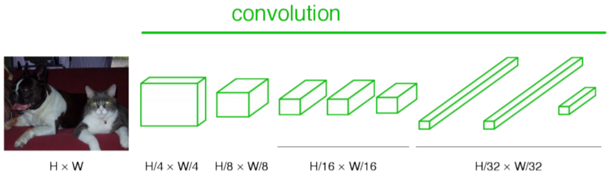

Figure 3. Fully Convolutional

3.3 Encoder and Decoder Network

The process to classify every pixel of the image into different classes based on some logic is called semantic segmentation. A general semantic segmentation architecture can be broadly thought of as an encoder network followed by a decoder network:
An encoder is usually a pre-trained classification network like VGG/ResNet which takes an input image and generates a high-dimensional feature vector. It aggregate features at multiple levels.

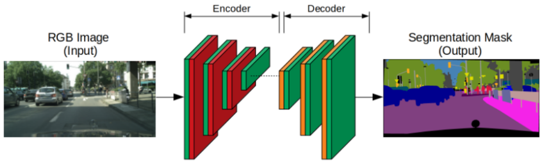

Figure 4. Encoder and Decoder Network

The task of the decoder is to take the high-dimensional feature vector generated by the encoder and semantically project the discriminative features (lower resolution) learnt by the encoder onto the pixel space (higher resolution) to get a dense classification. Thus, it decodes features aggregated by encoder at multiple levels. Unlike classification where the result of the very deep network is the only important thing, semantic segmentation not only requires discrimination at pixel level but also a mechanism to project the discriminative features learnt at different stages of the encoder onto the pixel space.[4]

3.4 Building blocks of Segmentation

To obtain a segmentation map (output), segmentation networks[3] usually have two parts and a third part called Skip connections:

1. **Downsampling path:** The goal is to extract and interpret the contextual/semantic (what) information. It may be done by max-pooling, average pooling or strided convolution.

2. **Upsampling path:** It is used to recover spatial information by precise localization (where). It may be achieved by un-pooling or deconvolution.

3. **Skip Connections:** A connection that bypasses at least one layer, It is used to fully recover/transfer the fine-grained spatial information lost in the pooling or downsampling layers by summing or concatenating feature maps from the downsampling and upsampling paths. In FCN, the skip connections from the earlier layers are also utilized to reconstruct accurate segmentation boundaries by learning back relevant features lost during downsampling. This helps combine context information with spatial information. They help traverse information faster in deep neural networks and pass information to lower layers so using it to classify minute details becomes easier.

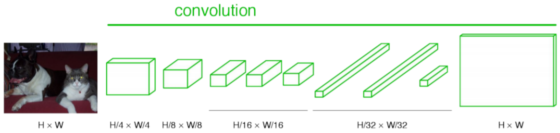

Figure 5. Upsampling

3.5 Key features of FCN

FCN transfers knowledge from VGG16 to perform semantic segmentation. The fully connected layers of VGG16 are converted to fully convolutional layers, using 1x1 convolution. This process produces a class presence heat map in low resolution. The upsampling of these low resolution semantic feature maps is done using transposed convolutions (initialized with bilinear interpolation filters). At each stage of FCN, the upsampling process is further refined by adding features from coarser but higher resolution feature maps from lower layers in VGG16. Skip connection is introduced after each convolution block to enable the subsequent block to extract more abstract, class-salient features from the previously pooled features.[2]

3.6 Evaluation Metrics

The following evaluation metrics are used to know how well a model performs in Image Segmentation:

Pixel to Pixel Accuracy: It is the percent of pixels in the image that are classified correctly.

$$PixelAccuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

However, it is not the best metric if we are dealing with an imbalance in classes. Considering our data is multi class and imbalanced, we used other coefficients to evaluate the models better.

Dice Coefficient: Dice Coefficient is 2 times the Area of Overlap divided by the total number of pixels in both images. The implementation is given below,

$$F1/Dice = \frac{2\times TP}{(2\times TP)+FP+FN}$$

Intersection over Union / Jaccard (IoU): IoU is the area of overlap between the predicted segmentation and the ground truth divided by the area of union between the predicted segmentation and the ground truth.

$$IoU/Jaccardmetric = \frac{TP}{TP+FP+FN}$$

Both the Dice and the IoU metrics are positively correlated.[5]

4. Models and Results

In this section, we discuss about the models that we tried and their methodologies.

4.1 FCN8 - VGG16

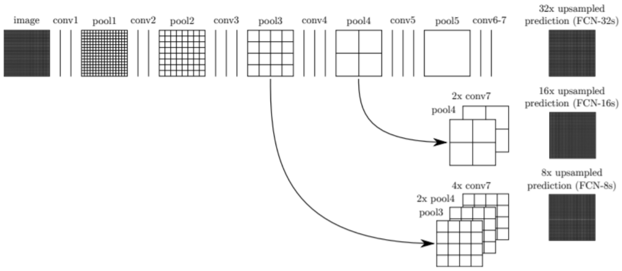
image
conv1
pool1
conv2
pool2
conv3
pool3
conv4
pool4
conv5
pool5
conv6-7
32x upsampled prediction (FCN-32s)
2x conv7
pool4
16x upsampled prediction (FCN-16s)
4x conv7
2x pool4
pool3
8x upsampled prediction (FCN-8s)

Figure 6. FCN-8 Architecture

As proposed in the paper, a pre-trained VGG16 is used as an encoder. The decoder starts from Layer 7 of VGG16. The last fully connected layer of VGG16 is replaced by a 1x1 convolution. FCN Layer-8 is upsampled twice to match dimensions with Layer 4 of VGG 16, using bilinear resizing (resize-convolution layers), Post that, a skip connection was added between Layer 4 of VGG16 and FCN Layer-9. FCN Layer-9 is upsampled twice to match dimensions with Layer 3 of VGG16, using bilinear resizing post which, a skip connection was added between Layer 3 of VGG 16 and FCN Layer-10. FCN Layer-10 is upsampled 4 times to match dimensions with input image size so that we get the actual image back and depth is equal to the number of classes. FCN-8 architecture is shown in **Figure 6** where pooling and prediction layers are shown as grids that reveal relative spatial coarseness, while intermediate layers are shown as vertical lines.

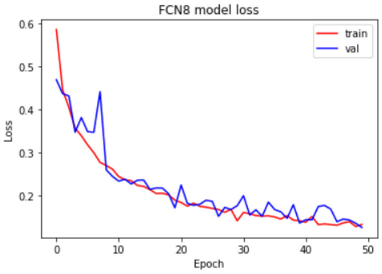

Figure 7. Model Loss for FCN8

4.2 U-Net - VGG16

The U-Net is based on the architecture of Fully Convolutional Network. However, it has some modifications which lead to improved/better segmentation results. When compared to FCN-8, it has 2 major differences: 1) U-Net is symmetric; 2) Skip connections concatenate the feature maps obtained from the upsampling path and the downsampling path instead of adding them. These skip connections aim to provide localized spatial information for the global information during upsampling. Owing to the symmetry in architecture, the U-Net network has a large number of feature maps in the upsampling path, thereby allowing transfer of information. In contrast, FCN architecture had feature maps in its upsampling path equal to the number of classes.U-Net uses Convolutional Transpose layers for upsampling. [6,7]

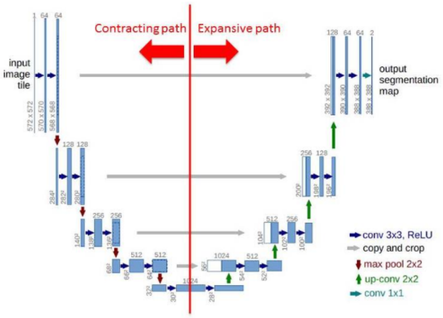

Figure 8. U-Net Architecture

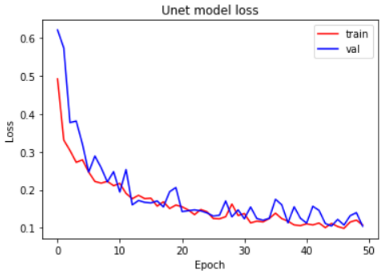

Figure 9. Model Loss for U-Net

5. Summary

As shown in **Figure 7** and **Figure 9**, we implemented two models– FCN-8 and U-Net where both U-Net and FCN8 converged at around 50 epochs.

Model Performance Summary			
Models	Pixel-Accuracy	mean-IoU Metric	Dice-coefficient
FCN-8 VGG16	95.93%	0.8552	0.9214
U-Net-VGG16	96.04%	0.8899	0.9412

The table above deocts the accuracies on the validation set. This table along with the segmentation images(**Figure 10**), clearly depict that although pixel accuracy for both models are comparable, U-Net performs better than FCN-8 with better Dice and IoU metic values of 0.9412 and 0.8899 respectively.

As seen above, FCN-8 predictions are coarser compared

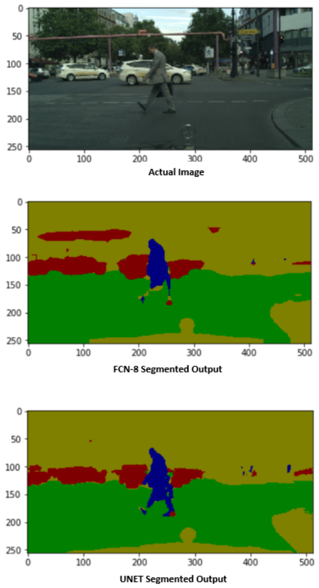

Figure 10. Segmented Images

to the images produced by U-Net. The person in the picture is being predicted better using U-Net as compared to FCN. Since this is the test set, we do not have ground truth values to exactly validate. However, we can validate on visual basis if the classes are getting predicted correctly.

6. Conclusion and Future Scope

With better GPU capabilities and hyper parameter tuning architectures, we would be able to achieve even better accuracies. We could not use the entire dataset though, if used, the results would have been much different and probably, better. We used only gtFine data, however, we could further train the coarser annotations for fine edge detection in future.

As a future scope, ResNet and pspnet can be used to implement and check which model would be the best bet to segment the images and give a better vision sytem for the autonomous vehicle system.

Acknowledgments

We thank Professor Minje for giving us the wonderful opportunity to explore the field of Image Segmentation and for the valuable knowledge and suggestions shared throughout the course. We also thank various sources that we took help from, without which this work would not have materialized. Last but not the least, we would like to thank all the students of ENGR-E533 Deep Learning Systems Fall 2019 Course for the essential information shared through Piazza discussions.

References

[1] *The Cityscapes Dataset for Semantic Urban Scene Understanding*.
Daimler AG R&D, TU Darmstadt, MPI Informatics, TU Dresden

[2] *Fully Convolutional Networks for Semantic Segmentation*.
Jonathan Long, Evan Shelhamer, Trevor Darrell

[3] *How to do Semantic Segmentation using Deep learning*.
James Le

[4] *Medium blog: ESPNetv2 for Semantic Segmentation*.
Sachin Mehta

[5] *Medium blog: Metrics to Evaluate your Semantic Segmentation Model*.
Ekin Tiu

[6] *Medium blog: Semantic Segmentation of Aerial images Using Deep Learning*.
Utkarsh Ankit

[7] *U-Net: Convolutional Networks for Biomedical Image Segmentation*
Olaf Ronneberger, Philipp Fischer, and Thomas Brox