CGI and Real Images for Semantic Segmentation on Autonomous Vehicles

William Sigala
Computer Science and Engineering
University of Texas at Arlington
Arlington, United States
william.sigala@mavs.uta.edu

Noah Wood
Computer Science and Engineering
University of Texas at Arlington
Arlington, United States
noah.wood@mavs.uta.edu

***Abstract*—In the field of autonomous vehicle driving, the vehicle's ability to perceive its surroundings is an essential feature that allows it to perform driving with autonomy. Being able to extract information from a vehicle's environment with cameras that provide pixel data such as patches of sidewalks, people, streets, and road conditions makes it possible. The primary method of gathering this information is by using Semantic segmentation. There have been several types of semantic segmentation models in recent years with varying performance with a focus on efficiency and accuracy. This method of feature extraction requires a considerable amount of computation which can lead to slower performance, a critical attribute due to the nature of autonomous vehicle driving, so it must be efficient enough to run in real-time. It also needs to be highly accurate to ensure that every decision it makes is based on reliable results given by the model.**

***Index Terms*—Semantic segmentation, autonomous vehicle, feature extraction, ground truth**

I. Introduction

The motivation behind this paper is to review and provide insight on the research done in the field of autonomous driving, specifically the detection of objects in a vehicle's environment. The most common technique used for this problem is by applying image segmentation, the process of labeling pixels of an image as certain objects, from a camera mounted on the vehicle. By using this technique, the vehicle can use the information gathered from this technique and make decisions based on the environment and its surroundings. For this paper, we have selected a variety of common datasets used for researching semantic segmentation and provide image data with labeled pixels and a popular semantic segmentation architecture that is suited for detecting and identifying objects in a vehicle's environment. The datasets we have chosen include CARLA (CAR Learning to Act) [6] and Cityscapes Image Pairs [7] which will be discussed in more detail later in the paper. Autonomous vehicles are a rapidly growing industry on the current infrastructure of the world. As such vehicles are put into the world, it is critical to understand not only how these vehicles track and see around them, but also the challenges computationally such methods like semantic segmentation has on these vehicles. In addition, the use of not only real images but also computer generated images are interesting ways to further the confidence and understanding of how such models are influenced by the training datasets.

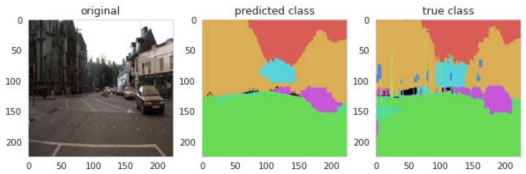


Fig. 1: Example of the semantic segmentation with autonomous driving from [2]

II. Literature Review and Related Works

Autonomous driving is a vastly growing field for the automobile industry. As a part of this is the expanding research for object detection utilizing semantic segmentation. Semantic segmentation is the process of calculating some classifying label associated with every pixel in the image. Object detection is not a new field in computer vision and machine learning. Various techniques from [1, 2, 3, 4] are applied to different but similar problems. Specifically, [2] delves into the object detection of autonomous vehicles with a big problem being the localization and mapping of detected objects. In the case of autonomous driving this is crucial because the estimated location of the objects potentially being people, cars, streets, etc are the building blocks of the information gathering in an automotive system. Many techniques can be used for the detection aspect of the autonomous driving system's information gathering. In addition, solutions with semantic segmentation are tough to get truly accurate as there are many influencing parts in any possible image. [1] discusses how semantic segmentation not only utilizes a classification method to identify the pixels in each object and particularly the computation constraints in doing so. A major challenge is the performance and computation speed of classification systems in autonomous driving systems while also maintaining a confident output. Semantic segmentation, while a good bit computational expensive outputs strong results making it a good choice for autonomous driving detection system. Semantic segmentation has various methods discussed in [3] with each again in simple terms, represent a classification of every pixel in an image. Various methods were used in the past, but recently a favored method is done using Convolutional Neural Network (CNN). [5] formed

Fig. 2: Sample of the computer generated images in the CARLA dataset

a basis for which [3] built from in the studies of convolutional networks for images that provide training model that train pixel to pixel with semantic segmentation. In such models exists a full image and pixel prediction from learning. This paper will attempt to extend the learning from these papers on the use of semantic segmentation and CNNs specifically in the realm of autonomous vehicle driving.

III. Analysis Design and Implementation

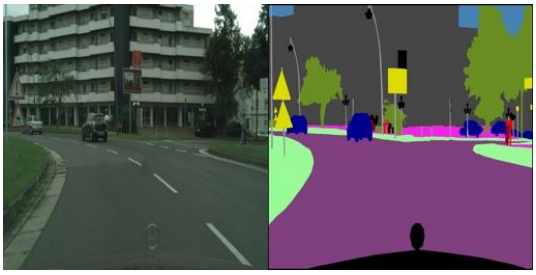
The model architecture selected for performing semantic segmentation on the datasets is U-Net, a fairly common convolution neural network architecture with many variations used for image segmentation. The goal of our experiment is to apply the same U-Net model on the datasets provided for semantic segmentation in an autonomous vehicle environment. Analysis will include an accuracy and computational speed assessment to further the understanding of the safety and performance of applying U-Net on separate datasets. It will provide some insight regarding the quality of data given by the datasets and whether real image or computer generated image data of vehicle driving footage influences the performance of the model.

A. CARLA (CAR Learning to Act)

CARLA [6] contains the ground truth pixel-wise semantic class labels for 28 video sequences from an urban driving simulator. From the 28 video sequences, it contains 10,767 frames in total, each video being recorded at 10 Hz, and with an average duration of 38.4 seconds. Of the videos recorded for this dataset, half of them were recorded in sunny weather, nine were recorded in rain, and the remaining five were recorded in cloudy weather. The recordings also include different driving conditions including moderate traffic, traffic jams, stopping at lights, etc... A sample of the simulation images that are being used is shown above. A key point to be noted is how much brighter and the difference of contrast between the images, this is a key aspect for why the dataset was chosen.

TABLE I: Five samples of CARLA semantic class RGB values and frequencies

Index	Semantic class	RGB values	Relative Frequency
0	Traffic Sign	[220,220,0]	0.05
1	Building	[70,70,70]	14.08
2	Fence	[190,153,153]	0.28
3	Other	[250,170,160]	0.29
4	Pedestrian	[220,20,60]	1.02



(a) Original

Fig. 3: The two images highlight the relationship between the original (left image) and ground truth (right image) pair in the Cityscapes dataset

B. Cityscapes Image Pairs

Cityscape Image Pairs [7] from the Berkeley AI Research group has 2975 images for training and 500 validation images. The images are 256x512 pixels. Each image contains the original photo on the left half alongside the labeled image on the right half. Unlike the CARLA dataset, these images are from real world places and as a result have particular qualities including the lighting and shadows being less contrasting than computer generated datasets as well as being what the model would actually be used for.

C. U-Net Architecture and Design

The architecture features an encoder for downsampling the input followed by a decoder for upsampling, each consisting of convolution blocks with skip connections (Fig. 4). The input will be a batch of images with shape ($\mathbf{BxCxWxH}$) where $\mathbf{B = 8}$ is the batch size, $\mathbf{C = 3}$ represents the number of channels the image has, and $\mathbf{W = H = 256}$ represents the width and height respectively. As the architecture implies, the output will match the input shape. We included dropout regularization with a probability of 20% between the encoder and decoder layers and before the last convolution layer as a means of preventing the co-adaption of neurons. For training, Mean Square Error (MSE) will be used as the criterion for computing the loss and Adam optimizer for updating the neurons. The model will be constructed with PyTorch and visualization tools such as matplotlib and pandas will be used.

A convolution block is defined by the following sequence of operations on a given input:

$\mathbf{ConvBlock}$(in_channels,out_channels=1,stride=1,padding=1)
- $\mathbf{in\_channels}$: the number of channels in the input image

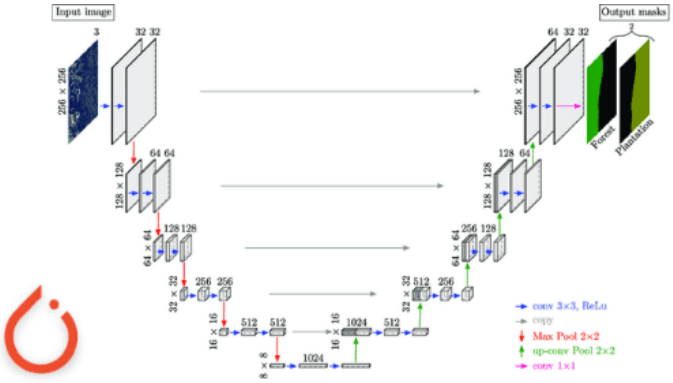


Fig. 4: Structure of U-Net Architecture that illustrates the ConvBlocks and skip connections of the encoder and decoder

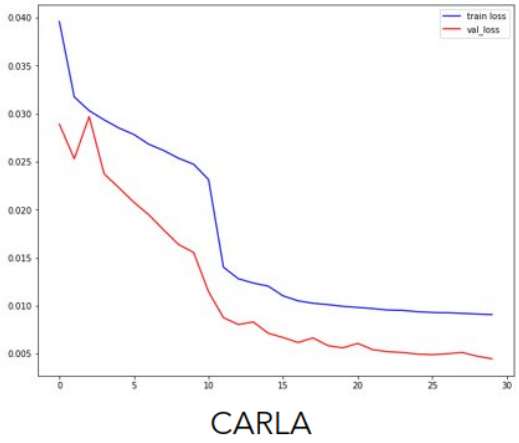


Fig. 5: U-Net model on CARLA dataset (training in blue and validation in red)

- **out_channels**: the number of channels produced by the convolution
- **stride**: the stride of the convolution by the given number
- **padding**: pads all fours sides of the image by the given number

- **Forward pass**:

- Conv2d(in_channel, out_channel, kernel, stride, padding), convolution
- BatchNorm2d(out_channel), batch normalization
- ReLU(inplace=True), activation
- Conv2d(out_channel, out_channel, kernel), convolution
- ReLU(inplace=True), activation

For the decoder, a deconvolution will precede each ConvBlock with matching input and output channels.

D. Data Preprocessing

Each of the datasets provide the original image and a masked image with the ground truth labels for every pixel. However, they are provided in different forms and will have to be loaded in independently for each. The images will have to match the input shape of the model.

- For the CARLA dataset, the data is given by two separate folders where one contains the original images and the other contains the ground truth segmentation map for each of the video sequences. In total there are 10,767 samples
- The Cityscapes dataset includes the original and ground truth segmentation map in a single image where the original is on the left and the ground truth to the right. The images will be split into two containers, one that contains all of the original images and the other containing the ground truths.

Once the data is properly separated by original and ground truth images, they will be resized to match the input shape of the model. There are an uneven amount of samples for each of the datasets with differences in video quality. In order to give a more balanced analysis, we will test the model with roughly the same number of samples for each of the datasets. As for the CARLA dataset, we will select one of the video sequences for this. There is a distinction to be made between the CARLA dataset and the other two in terms of quality. CARLA provides computer generated images from vehicle driving simulation, while the others are real images gathered from driving.

E. Feature Extraction

The primary libraries used for this implementation include PyTorch and OpenCV. The implementation after the preprocessing of the image datasets will take the images through the encoding section of using a set of filter sizes. Since the U-Net will also need to be decoded, the same filter sizes will be used for encoding as decoding. In terms of feature extraction, semantic segmentation can be a bit computationally expensive as each pixel in the image will need to be given its own classification. Feature will be encoded in the first half of the U-Net structure using the convolutional layers based on the filter sizes provides. These extracted feature will then be decoded up the latter half of the U-Net structure with the result being an output image matching the shape of the input image. See (Fig. 1 and Fig. 2).

IV. Experiment Results

The results we gathered from experimenting on the U-Net model can simply be visualized by the provided images and loss graphs.

A. Training experiments

For the loss graphs, we initially did not use dropout regularization which resulted in a validation curve that would not converge. We included dropout regularization in between the encoder and decoder, as well as before the output which resolved this issue. The size of the validation set and training set we used is 30% and 70% respectively, which influences the difference between each of the loss curves as see in the graphs. A learning rate of 0.01 with Adam optimizer and batch size of 8 was used.

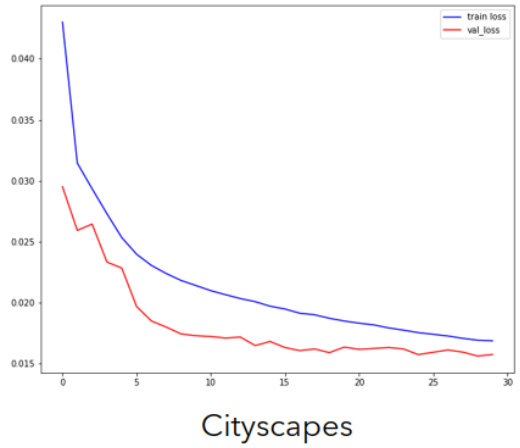


Fig. 6: U-Net model on Cityscapes dataset (training in blue and validation in red)

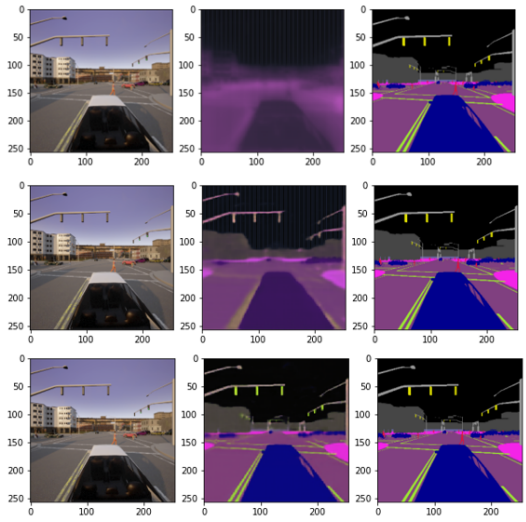


Fig. 7: CARLA images (original, predicted, ground truth) at epoch 1, 15, and 30

For both datasets, we found that training performance was roughly the same as the rate of convergence was more or less similar.

B. Predictions

The figures (Fig. 5, 6) displays the sample at the beginning, middle, and end of training the model for each dataset. The Cityscapes sample image (Fig. 6) appears to be more fuzzy in contrast with the CARLA sample (Fig. 5). The images from the CARLA dataset was able to produce a more sharp and pronounced segmentation map. The inference time for each of the datasets remained the same as expected, with an average time of 3 milliseconds for a forward pass running on an NVIDIA GTX 1080 graphics card.

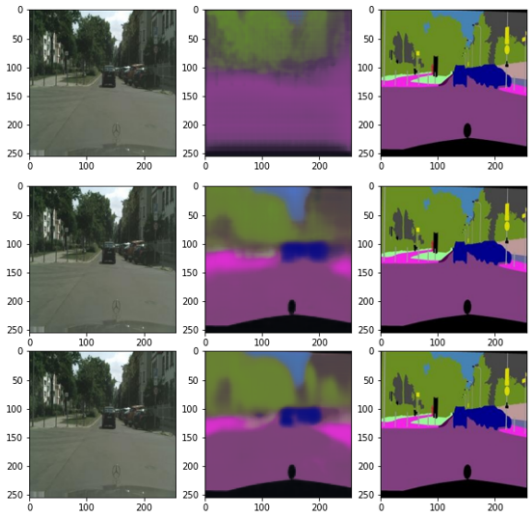


Fig. 8: Cityscapes images (original, predicted, ground truth) at epoch 1, 15, and 30

V. Conclusion

Labeling each pixel in the image is critical in autonomous driving as anything the vehicles ”sees” would need to checked and classified. Image Segmentation provides performance and accurate results. Training with computer generated images has the potential goal in allowing the model to be more robust, as well as showcasing a more applicable solution sample due to the difference of lighting, shading, and contrast that exists in computer generated images. This is crucial as vehicles can be in numerous different environments, lighting, and shaded areas. The CARLA dataset was able to provide an entry point to which we could explore the use of computer generated images and how they affect training and output of semantic segmentation. This in combination with the real images from the Cityscapes Image Pairs gave a clear way to distinguish the difference of using real versus computer generated images. After reviewing the results of the semantic segmentation model using real and computer generated data, it is clear that the model trained from the CARLA dataset was a bit more precise in the output than the Cityscapes dataset as shown from the output samples after training. Particularly the CARLA dataset model was able to detect smaller object such as the lights and signs significantly better while the Cityscapes model which struggled to capture some of such objects. Larger objects like the streets, cars, and surrounding environment were still detected well in both models. Possible reasons for this discrepancy is the fact the CARLA dataset is computer generated so things such as differences in lighting and shadows may be far less drastic than in real images used in the Cityscapes. This problems exists in other solutions as well, specifically [2] in which small objects were also tough to distinguish. The models are more fine tuned when objects have a greater contrast difference allowing a more distinguishable pattern from surrounding objects, particularly large objects like walls or trees. As for the machine learning architecture, U-Net was

able to provide a performant and decently accurate solution for object detection and image segmentation. This applied in an autonomous driving vehicle standpoint is sufficient to keep up with the real time usage that these vehicles require, and as a result satisfactory on that front as the inference time of a few milliseconds is capable of keeping up with the speed required. The other factor for safety and real application of the U-Net structure in autonomous driving is accuracy. As mentioned, there was a stronger detection found when using the computer generated images than real images, but overall, the output image segmentation gave correct output. A possible solution for this discrepancy however, is more training images. Particularly, more with images that have lower contrasting edges between the objects and more variable lighting so that the model will be able to better detect in such scenarios. Overall, this implementation of the U-Net model had a visually fair accuracy with room for improvement. It was able to meet the speed requirements and capture objects accurately, making the U-Net architecture trained with real and computer generated images a sound solution for the autonomous driving object detection problem with semantic segmentation while also showcasing the use of computer generated images in model creation.

REFERENCES

[1] R. N. Lazuardi, D. Sudrajat, N. Aulia and T. Adiono, "A System of Semantic Segmentation on An Autonomous Vehicle," 2019 16th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Pattaya, Thailand, 2019, pp. 786-789, doi: 10.1109/ECTI-CON47248.2019.8955214.

[2] A. Agafonov and A. Yumaganov, "3D Objects Detection in an Autonomous Car Driving Problem," 2020 International Conference on Information Technology and Nanotechnology (ITNT), Samara, Russia, 2020, pp. 1-5, doi: 10.1109/ITNT49337.2020.9253253.

[3] G. P.G., "Different Approaches for Semantic Segmentation," 2020 5th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2020, pp. 938-943, doi: 10.1109/ICCES48766.2020.9137966.

[4] T. Cane and J. Ferryman, "Evaluating deep semantic segmentation networks for object detection in maritime surveillance," 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Auckland, New Zealand, 2018, pp. 1-6, doi: 10.1109/AVSS.2018.8639077.

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

[6] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, & Vladlen Koltun (2017). CARLA: An Open Urban Driving Simulator. In Proceedings of the 1st Annual Conference on Robot Learning (pp. 1–16).

[7] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The Cityscapes Dataset for Semantic Urban Scene Understanding," in Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.