# Level 1 Autonomous Driving

Tom Dörr[1], Theodor Cheslerean Boghiu[2], Mohammed Asif Chand[3], and Vlad Paul Cosma[4]

[1]tom.doerr@tum.de [2]theo.cheslerean@tum.de [3]mdasifchand@gmail.com [4]vlad.cosma@tum.de

## Semantic Segmentation

- Definition: Semantic segmentation means assigning a class label to each of the computed segments example: Road, traffic lights, lanes. Examples: Fully CNN(2015), SegNet(2015), ENet(2016), LinkNet(2017).
- Semantic Segmentation is used to detect road in the absence of clear road-lane separation
- Present developments in Semantic Segmentation on autonomous vehicles

  SqueezeNet(2016) was able to demonstrate the increase in accuracy of AlexNet using 50 times less parameteres by tweaking with the architectures

  ENet(2016) could replicate the Semantic Segmentation on real time

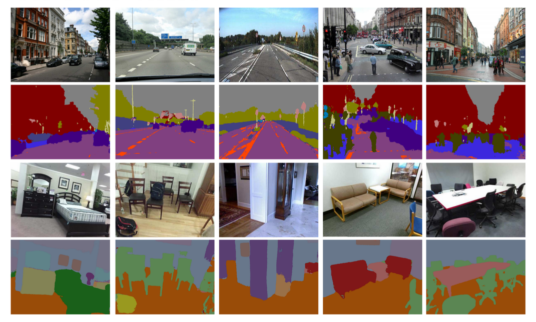

**Figure 1:** Segnet predictions on road scene and indoor scenes

## Udacity Simulator [4]

Udacity provides a free and easy to use platform for testing and gathering data in autonomous driving tasks. It encompasses the following set of features:

- Compatible with both Windows and Linux systems(compared to GTA V)
- Built-in

  Data Collection Mode

  Autonomous Driving Mode
- Customizable through Unity

The data gathering is done by capturing the screen output and the steering angle input from a user. This data is saved in the form of the screenshots, each one annotated with the steering angle value, which can then be used for training.

## Cityscapes Dataset [3]

The cityscape dataset is used for the semantic segmentation part of our project. It encompasses the following features:

- 30 class definitions related to city driving
- Various weather conditions
- 25000 annotated images
- Metadata including: GPS coordinates, Odometry data of ego vehicle, Preceding and trailing video frames

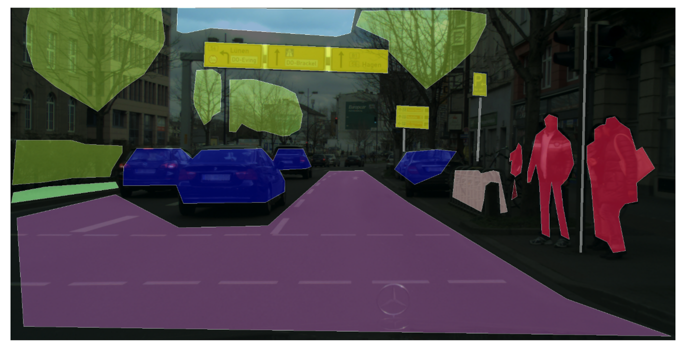

**Figure 2:** Cars - blue, Pedestrians - red, Road - purple, Sidewalk – light green, Vegetation – green, Traffic signs - yellow, Guard rail – light brown

## Segmentation Architecture [2]

- The architecture selected for the problem definition was taken as SegNet(2015) because of it's simplicity
- Transfer training of the weights is performed and then fine tuned to attain the segmentation of the road.

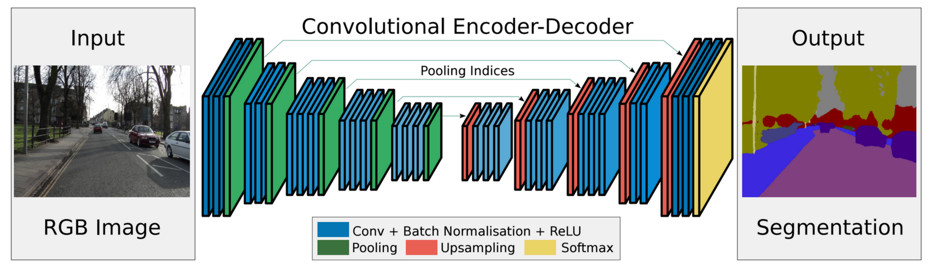


**Figure 3:** SegNet does not include any fully connected layer hence it is only convoluted. A sparse map of features is produced by the decoder which upsamples input produced by the encoder using encoders transferred pool indices. Then performs convolution with a trainable filter bank to densify the sparse feature map. Softmax classifier takes the final decoder output for pixel wise classification

## Udacity Dataset

The udacity dataset is created from the udacity car simulator. It associate via a timestamp 3 images taken at an offset (left, center and right) with steering angles, brake and acceleration. Our dataset contains 8036 frames of the simulation game along with odometry information. These images have been resized and modified in the following way randomly in order to increase the model's prediction capacity:

- Randomly chosen from an offset (left, center, right)
- Brightness changed
- Cropped
- Shadow mask added
- Flipped (left becomes right and right becomes left)
- Translated (shift the image vertically and horizontally)
- RGB 2 YUV (YUV is a color space more closely related to human perception)

All of the preprocessing is done using the python opencv library.

**Figure 4:** Left, Center and Right Image

## Nvidia Architecture

A modified version of the Nvidia model is used[1].
We're leaving out one fully connected layer and cut the number of Convolutional feature maps in halve. We still have the last 3 fully connected layers and also have the five convolutional layers of the original network. As an activation function we are using ELU.

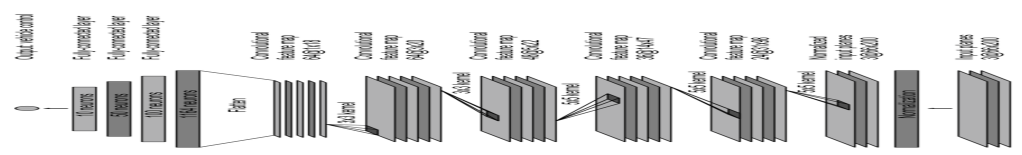

**Figure 5:** Cars - blue, Pedestrians - red, Road - purple, Sidewalk – light green, Vegetation – green, Traffic signs - yellow, Guard rail – light brown

## Segmentation Training Results

## Segmentation Testing Results

## Driving Training Results

## Driving Testing Results

## References

[1] https://devblogs.nvidia.com/parallelforall/deep-learning-self-driving-cars/ [2] SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation by Vijay Badrinarayanan et.al [3] https://www.cityscapes-dataset.com/ [4] https://github.com/udacity/self-driving-car-sim