Affordable Self Driving Cars and Robots with Semantic Segmentation

Gaurav Bansal
gauravbs@stanford.edu

Jeff Chen
jc1@stanford.edu

Evan Darke
edarke@stanford.edu

Abstract

Image semantic segmentation is of immense interest for self-driving car research. Accurate and efficient segmentation mechanisms are required. We have evaluated Intersection over Union (IoU) metric over Cityscapes and KITTI datasets. We designed baseline softmax regression and maximum likelihood estimation, which performs quite poorly for the image segmentation task. We ran fully convolutional networks (FCN), which performed almost perfectly over the KITTI dataset. We found overfitting problem for the more complex Cityscape dataset. We conducted several experiments with regularization, dropout, data augmentation, image scaling and newer architectures. We are able to successfully mitigate overfitting by data augmentation. We also generated a confusion matrix and conducted error ablative analysis to get a deeper understanding of FCNs.

1. Introduction

Self driving car technology has a potential to revolutionize how we travel. It would not only make driving safer and more efficient, but also free passengers to utilize the driving time in various other productive activities. Traditional self-driving car design has dependence on mapping and localization. This approach requires expensive sensors such as LIDARs and high-precision GPS. An alternative approach is to conduct semantic segmentation on the camera data. Semantic segmentation, which means labeling images with pixel by pixel classification, will be able to perceive the surrounding environment. This includes both free roadspaces (roadways, pedestrian walkway etc.) and dynamic objects (vehicles, pedestrians, bikers etc.) [3, 4]. This approach is perhaps also closer to how humans perceive the driving environment. As opposed to LIDAR and High-Definition map based systems, vision based autonomous driving systems are much more affordable [4]. Further, semantic segmentation could also be incredibly useful for robot navigation because it is not always feasible to produce real time maps of indoor spaces (e.g., tables and chairs can be moved easily) [9]

2. Related Work

Semantic segmentation is a widely studied topic in Computer vision. The state-of-the art algorithms rely on deep learning based frameworks. In [1], FCNs are proposed as a first deep learning architecture for semantic segmentation. This architecture modified VGG-16 based Convolutional Neural Networks for segmentation task. Improvements were reported by using RESNET-50 based architectures [15]. The state-of-art results by Google uses a combination of RESNET-based architecture and dilated convolutions and is currently the best model on cityscape leaderboard for category IoU metric [18]. Researchers at Audi have modified Squeezenet to work for semantic segmentation [17], which improves inference time at the cost of accuracy.

3. Datasets

3.1. KITTI

Our first dataset is KITTI [5], which contains 289 images of 160x576 pixels with two classes: road and non-road. We divided the dataset into a training set of 231 images and a validation set of 58 images. In Figs. 1 and 2, we show a raw and labeled image in KITTI training set. In Fig. 3, we have plotted the frequency with which two classes (road and non-road) appear in the images.

3.2. Cityscapes

Our second and main dataset is Cityscapes[6], which has 5000 large images of 2048x1024 pixels with 30 classes in 8 categories. Examples of classes are car, truck, bus, which belong to the vehicle category; and road, sidewalk, parking, which belong to the flat category. The dataset is divided into 2975 training images, 500 validation images and 1525 test set images. We show the number of pixels present per category in Fig. 6, with flat, construction, and nature occupying the most number of pixels across all images.

4. Methodology

4.1. Metrics

We use intersection over union (IoU), averaged over each class as our primary metric. IoU is more robust than per

Figure 1: KITTI Image

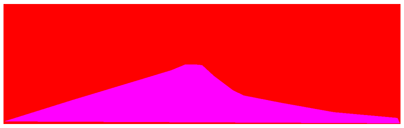


Figure 2: Annotated Image

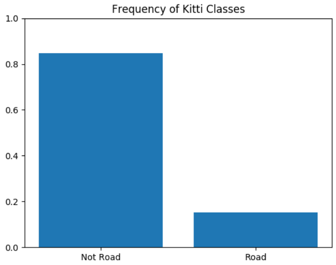


Figure 3: Class Distribution

Figure 4: Cityscapes Image

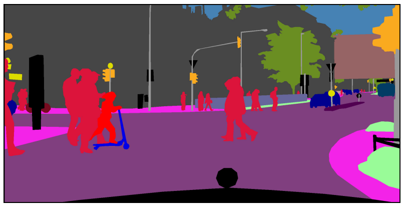


Figure 5: Annotated Image

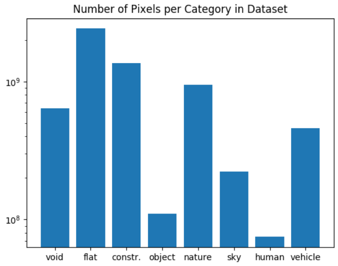


Figure 6: Pixel Distribution

pixel accuracy when there is a large class imbalance. For example, always predicting Not Road on KITTI would yield an accuracy of 85% but an IoU of only 42.5%.

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

4.2. Baseline Algorithms

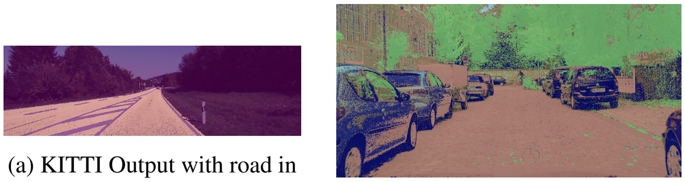
4.2.1 Softmax Regression

In this first baseline model, we predict the class of a pixel based on its color and the color of its neighbors within a window size of WxW. We train a softmax classifier to predict the class of the pixel at the center of the window based on the red, green, and blue components of each pixel within the window.

4.2.2 Maximum Likelihood Estimation

For this second baseline model, we assume that each class is associated with one or more color ranges. Therefore, we divide our color space into B buckets and estimate the class distribution of each. We define $b(x)$ as the bucket corresponding to the color of the pixel x. We then define our predictor as follows:

$$h(x) = \underset{y}{\operatorname{argmax}}\, p(y|b(x)) = \underset{y}{\operatorname{argmax}}\, p(y, b(x))$$



(a) KITTI Output with road in white, not road in purple

(b) Cityscapes Output has reasonable performance on trees, cars, and ground

Figure 7: Maximum Likelihood Estimate

where the joint distribution $p(y, b(x))$ is estimated by counting occurrences in the training set. The number of probabilities that we must estimate in this model for B bins and K classes is $O(BK)$

4.2.3 Discussion

Table 1 shows that our MLE algorithm significantly outperforms both the naive majority algorithm and softmax regression. The output of our MLE model (Fig. 7) classifies most pixels correctly with very tight boarders on each region, but regions are peppered with misclassifications. Furthermore, the model struggles with shadows and transparent objects, such as car windows. This indicates that color alone is insufficient to classify pixels. In an attempt to remedy these shortcomings, we will train a fully convolutional network (FCN) on KITTI and Cityscapes.

4.3. Fully Convolutional Network

We chose FCN-VGG16 [1] as it was a breakthrough deep learning architecture for end-to-end segmentation and is of-

Table 1: Baseline Algorithms Validation IoU

	KITTI	Cityscape
Always Predict Majority Class	42.5%	5.1%
Softmax Regression(W = 1)	12.8%	13.1%
Softmax Regression(W = 3)	16.7%	15.4%
MLE (B = 2^{21})	68.7%	31.6%
MLE (B = 2^{18})	68.5%	31.8%

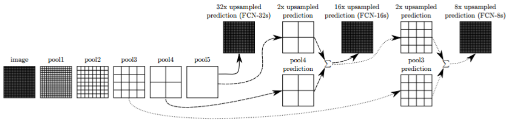


Figure 8: FCN Architecture

ten cited as a baseline for modern semantic segmentation algorithms such as PSPNet[14]and ResNet-34[15].

In Fig. 8, the architecture for a FCN is shown composed of 5 convolutional layers (each convolutional layer is followed by a pooling layer), upsampling (2x, 2x, 8x to preserve spatial dimension) and skip connections from layer 3 and 4 to retain fine-grain information from previous layers.

5. Experiments and Results

Our investigation starts by taking an existing implementation of FCN from Udacity [16]. We modified FCN-VGG16 architecture for various network configurations. Since IoU is not differentiable, we optimize categorical cross entropy loss as a proxy for IoU. We implemented early stopping by computing the validation loss after every training epoch and halting training if the validation error increases more than two times in a row. We implemented IoU metrics to evaluate the performance. We wrote our code using Tensorflow and ran simulations on Amazon AWS.

5.1. KITTI

5.1.1 Hyperparameter Tuning

We first conducted experiments to find the best values for various hyperparameters. We used validation loss as the criterion to select hyperparameters. We experimented with various design choices for learning rate, batch size, optimization algorithms; as shown in Figs 9, 10, and 11 respectively. We also ran experiments for keep probability for dropout used in VGG-16. We did hyperparamter tuning experiments with a image size of 160X576 pixels. We found that learning rate of .0001, batch size of 4, Adam optimization and keep probability of 0.6 to work best for our architecture.

5.1.2 KITTI: Initial Results

In Fig. 12, we plot the training set loss and validation set loss obtained by choosing best parameters from the experiments described above with input image size set of 320x1152. We have plotted validation loss versus training set size in Fig. 13. We can make two key observations from these plots: 1) Our network does not have variance problem as there is not a huge gap between training set loss and validation set loss. 2) The validation loss is still decreasing with training set size. This implies we could further decrease loss by having more data. We obtain 95.5% mean IoU score on the validation set (Fig. 14). We have plotted one output image from test set in Fig. 15. The pixels classified to be in road category are marked with green color.

5.2. Cityscapes - Initial Results

Cityscapes is an 8-class classification task (classes here refer to Cityscapes' 8 categories, not the 30 object classes). We used the same tuned hyperparameters from Section 5.1.1. Due to limited computing resources, we set the image to be 1/64 of original 2048x1024 pixels (resized to 1/8th on both height and width dimensions). Figs. 16, and 17, show that on Cityscapes FCN8 achieved an initial Val IoU of 59%, and a Train IoU of 68% with early stopping. There is a clear overfitting as the validation loss begins increasing while training loss continues to decrease. We tried regularization and additional dropout as first experiments. However, it achieved only marginal improvements in reducing overfitting.

5.3. Ablative Analysis

Network	Loss KITTI	IoU KITTI	Loss Cityscape	IoU Cityscape
FCN-8	.066	.954	.423	.595
Remove L3-skip	.073	.944	.455	.570
FCN-16	.077	.941	.438	.569
Remove L4-skip	.090	.932	.542	.509
FCN-32	.097	.926	.531	.489
L4-out	.085	.935	.428	.572
L3-out	.088	.935	.433	.573

Table 2: Ablative Error Analysis

We conducted ablative analysis to get a better understanding of our model. We removed layers from our network in this order: skip connection from layer 3, 3rd upsampling stage (i.e., now we upsample by 2x, followed by 16X - this is also called as FCN-16 in Long et. al [1]), skip connection from layer 4, 2nd upsampling stage (i.e., now we upsample directly by 32x - this is also called as FCN-32 in Long et. al [1]), conv5 layer (i.e. upsample layer 4 output by 16x), and conv4 layer (i.e. upsample layer 3 output by 8x). We show validation loss and mean IoU results for various configurations for both KITTI and Cityscapes datasets in Table 2. We observe from the results that layer 4 skip connection is most important.

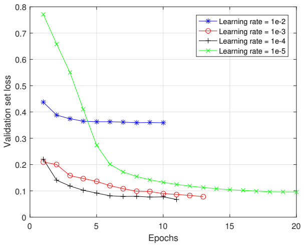


Figure 9: Learning Rate

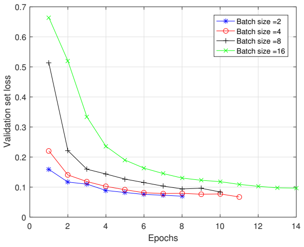


Figure 10: Batch size

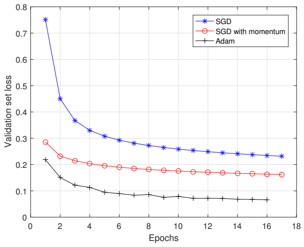


Figure 11: Optimization alg.

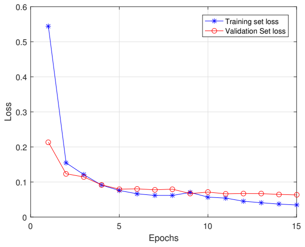


Figure 12: Loss vs Epoch

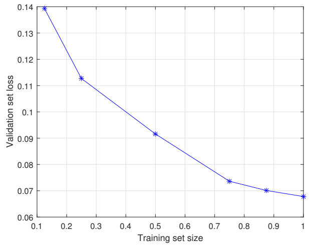


Figure 13: Loss vs Training set

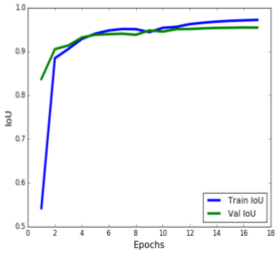


Figure 14: IoU vs Epoch

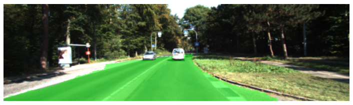


Figure 15: Output labeled Image

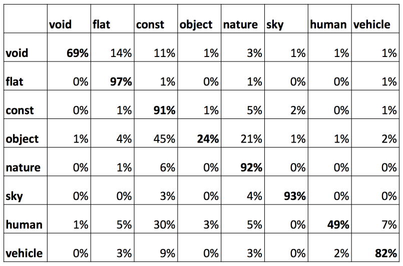
5.4. Cityscapes - Confusion Matrix

We are interested in which categories perform better and which perform worse. Figure 19 shows a confusion matrix with rows corresponding to the actual value of a pixel, and the columns being the predicted value. The human and object categories do far worse than others and significantly impact our mean IoU. We suspect this is due to two factors: 1) Figure 6 shows that human and object are the rarest examples in our training set and 2) humans and most objects, such as poles, tend to be very skinny. This means our IoU on these categories is very sensitive to noisy borders.

5.5. Cityscapes - Experiments

5.5.1 Data Augmentation

During training, we add random Gaussian noise to each image with zero mean and standard deviation of 5. To account for the different lighting conditions present in the Cityscapes validation set, we convert the images to HSL format and scale the lightness component by a uniformly random number and convert the image back to RGB format. Finally, with probability .5, we flip the image horizontally. We found augmentation to increase our final validation IoU by 3% and decreased our train IoU by 1% when training on 1/8 scale images.



	void	flat	const	object	nature	sky	human	vehicle
void	**69%**	14%	11%	1%	3%	1%	1%	1%
flat	0%	**97%**	1%	0%	1%	0%	0%	1%
const	0%	1%	**91%**	1%	5%	2%	0%	1%
object	1%	4%	45%	**24%**	21%	1%	1%	2%
nature	0%	1%	6%	0%	**92%**	0%	0%	0%
sky	0%	0%	3%	0%	4%	**93%**	0%	0%
human	1%	5%	30%	3%	5%	0%	**49%**	7%
vehicle	0%	3%	9%	0%	3%	0%	2%	**82%**

Figure 19: Confusion matrix of initial results

5.5.2 FCN-4 and FCN-2

Since we found that humans and objects are the primary source of our errors, and our model does a poor job at finding precise borders for these small entities, we propose extending the FCN-8 architecture with more gradual upsampling while preserving more fine grained features from early layers. In our FCN-4 model, we add an additional upsampling layer with a skip connection to pool2. In FCN-2 we add one more upsampling layer to FCN-4 with a skip connection to pool1.

	Val Loss	Val IoU
FCN-8	.437	58.6
FCN-4	.424	59.2
FCN-2	.430	59.6

Table 3: New Architectures performance improvement over 1/8 scale images

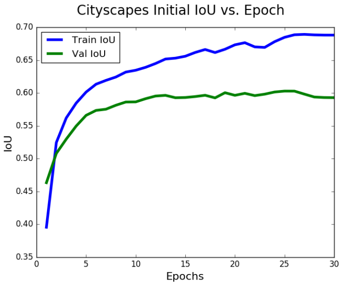


Figure 16: Cityscape IoU vs. Epoch

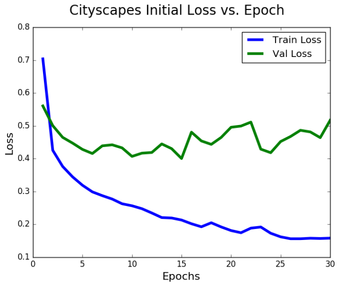


Figure 17: Loss vs. Epoch

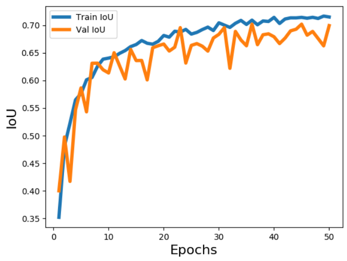
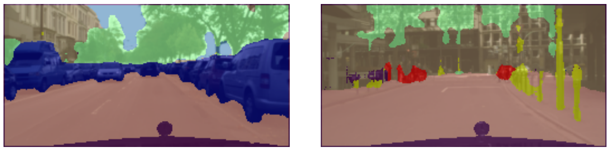


Figure 18: Final IoU vs Epoch



(a) Well segmented image

(b) Bad segmented image

Figure 20: Example output images

5.5.3 Scaling vs Cropping

The Cityscapes images are so large that our gpus can only process minibatches of size 1. Consequently, we were interested in the effect that training on downscaled or cropped images would have on our model's performance. We experimented downscaling images by a factor of 2, 4, and 8 using nearest neighbor interpolation. We also tried cropping 1/64th of each image randomly in each epoch. Both methods allow us to train a network in only a small fraction of the time required to train on full size images. Table 4 shows that cropping and scaling by 2 yields similar performance compared to training on full size images. However, performance degrades significantly as we scale images down further. Additionally, we find that random cropping mitigates our overfitting problem because every epoch we effectively train on brand new data.

6. Final Results and Discussions

We achieved the best correction to overfitting with the use of data augmentation by cropping images randomly as described in 5.5.3. Compared to our initial results, the ratio of Val Loss / Train loss was significantly reduced from 5.9 to 1.27. The final IoU performance can be seen in Fig. 18.

There is a difference of data being exposed to the model. Where as we omit pixels when scaling an image down, cropping actually exposes all pixels to the model during training. In some ways, this is not a 'fair' comparison since cropping introduces more data to the model - but this is the point of data augmentation. We compared the results of the cropping technique with 1/2 scale and Full Size Images (which take days to compute on AWS) in Table 4. It is clear that performance using the Cropping Technique achieves the same validation IoU as the training with full sized images, though there is less headroom as the model is now underfitting with a Train IoU of 72%. Fig. 20 show examples of well segmented and poorly segmented results. As explained with the Confusion Matrix, our model performs best with flat (roads, sidewalks), and sky categories and worst with object (poles, traffic lights) and human categories.

	Train Loss	Train IoU	Val Loss	Val IoU
Full Size	.07	.79	.55	.72
Scalex2	.08	.78	.60	.71
Scalex4	.23	.68	.39	.65
Scalex8	.28	.62	.44	.58
Crop	.26	.72	.33	.70

Table 4: Scaling and Cropping

7. Conclusion and Future Work

We started with baselines Softmax Regression and MLE for image segmentation. MLE performed reasonably with the 2-class KITTI dataset with a Val IoU of .69, but only achieves a Val IoU of .32 on the 8-class Cityscapes dataset. We did an in depth study of FCN8 architecture using pre-trained VGG16 weights and found almost perfect segmentation for the KITTI dataset, and overfitting on the Cityscapes dataset with the out-of-the-box model. Regularization didn't help much in our network, perhaps because we were only regularizing the weights in the FCN convolutional layers. The layers in our pre-trained VGG16 model are not regularized. Randomly cropping the training set images resulted in the best correction to overfitting, with a final Val IoU of .70 and Train IoU of .72.

Going forward, we plan to correct for underfit by introducing a deeper architecture (we attempted this, but were unfruitful). We also plan to introduce regularization to the VGG16 model weights as regularization should be able to reduce overfitting more than what we have seen. Finally, we would like to run the model on real-driving video data to test for classification performance as well as running time.

8. Contributions

Gaurav proposed the problem statement and running FCNs over Cityscapes and KITTI datasets. He developed the first code base for FCN. He conducted hyperparamter tuning and performance analysis over KITTI dataset. He also performed Cityscapes Ablative Error Analysis, Dropout and newer architecture implementation.

Jeff ran a separate code base to serve as a secondary check and plotted loss by Epoch. He performed regularization, different scale factor, introducing additional data, and training set size experiments. He also performed confusion matrix analysis.

Evan implemented early stopping and plotted loss by training set size, and loss by epoch. He also implemented data augmentation and random cropping.

All contributed to the writing to the report.

References

[1] J. Long, E. Shelhamer, and T. Darrell. Fully Convolutional Networks for Semantic Segmentation. CVPR 2015. 1, 2, 3

[2] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, Bernt Schiele. The Cityscapes Dataset for Semantic Urban Scene Understanding. CVPR 2016.

[3] Brody Huval, Tao Wang, Sameep Tandon, Jeff Kiske, Will Song, Joel Pazhayampallil, Mykhaylo Andriluka, Pranav Rajpurkar, Toki Migimatsu, Royce Cheng-Yue, Fernando Mujica, Adam Coates, and Andrew Y. Ng. An Empirical Evaluation of Deep Learning on Highway Driving. CoRR, abs/1504.01716, 2015. 1

[4] Mennatullah Siam, Sara Elkerdawy, Martin Jagersand, Senthil Yogamani. Deep Semantic Segmentation for Automated Driving: Taxonomy, Roadmap and Challenges. IEEE ITSC 2017. 1

[5] http://www.cvlibs.net/datasets/kitti/ 1

[6] https://www.cityscapes-dataset.com/benchmarks/ 1

[7] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, Bernt Schiele. The Cityscapes Dataset for Semantic Urban Scene Understanding CVPR - April, 2016

[8] Taigo M. Bonanni, Andrea Pennisi, Domenico Bloisi, Luca Iocchi, Daniele Nardi1. Human-Robot Collaboration for Semantic Labeling of the Environment.

[9] Abhijit Kundu, Yin Li, Frank Dellaert, Fuxin Li and James M. Rehg Joint Semantic Segmentation and 3D Reconstruction from Monocular Video. Computer Vision ECCV 2014. 1

[10] Joel Janai, Fatma Gu ney, Aseem Behl, Andreas Geiger. Computer Vision for Autonomous Vehicles: Problems, Datasets and State-of-the-Art.

[11] Yuki Furuta, Kentaro Wada, Masaki Murooka, Shunichi Nozawa, Yohei Kakiuchi, Kei Okada and Masayuki Inaba Transformable Semantic Map Based Navigation using Autonomous Deep Learning Object Segmentation

[12] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, Antonio M. Lopez. The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes.

[13] A Geiger, P Lenz, C Stiller and R Urtasun. Vision meets robotics: The KITTI dataset. The International Journal of Robotics Research. 2013

[14] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, Jiaya Jia Pyramid Scene Parsing Network CVPR 2017 3

[15] Zifeng Wu, Chunhua Shen, Anton van den Hengel. Wider or Deeper: Revisiting the ResNet Model for Visual Recognition. 1, 3

[16] https://github.com/udacity/CarND-Semantic-Segmentation 3

[17] Michael Treml, Jos Arjona-Medina, Thomas Unterthiner,Rupesh Durgesh, Felix Friedmann, Peter Schuberth, Andreas Mayr, Martin Heusel, Markus Hofmarcher, Michael WidrichBernhard Nessler, Sepp Hochreiter Speeding up Semantic Segmentation for Autonomous Driving 1

[18] Liang-Chieh Chen, George Papandreou, Florian Schroff, Hartwig Adam, Rethinking Atrous Convolution for Semantic Image Segmentation arXiv preprint arXiv:1706.05587. 1