Robust Semantic Segmentation

David Matos Rodriguez
Eindhoven University of Technology
Netherlands
d.e.matos.rodriguez@student.tue.nl

Abstract

Semantic Segmentation is the process of understanding and recognizing the objects in an image at the pixel level. Even though this is a challenging task in the computer vision field, it has been proven that good results can be observed with Convolutional Neural Networks(CNN) when images are taken in ideal conditions. However, the results turn out significantly worse once conditions are not ideal, namely rain, snow, or nighttime. Therefore, this paper aims at solving this issue by making a more robust CNN, primarily with hyper parameter tuning and data augmentation on the Synthia Dataset composed of synthetic images. By doing so, the experiments show a validation Mean Intersection over Union (MIOU) percentage increase in images in poor conditions when the network was augmented, as compared with the vanilla network which presented no augmentation.

1. Introduction

Autonomous vehicles have started to become a reality with the rise of new and different algorithms that approach problems in a different way. In order to make a vehicle drive by itself, it needs to be able to see what is surrounded by and to know what actions to take accordingly, just like any human would in a vehicle. Consequently, in order to simulate the human, cameras are placed in a vehicle to capture the world. Even though there are other approaches to autonomous driving, the main focus here will be the processing of the images taken by a camera, and the ability of a Convolutional Neural Network to recognize or segment objects around the vehicle. CNN are a type of deep learning that performs convolutions on an image, and is capable of detecting objects on an image.

As it has been experienced by everyone, cameras are great at capturing images under proper lighting conditions. With these conditions, CNNs are able to segment the objects—which is the process of recognizing the objects in the image at the pixel level. However, once conditions change, and rain/snow or nighttime appears, cameras become poorer at capturing what is in front, and the CNNs ability to segment the object also decreases drastically due to poorer conditions.

Furthermore, it is not possible to mandate an autonomous vehicle to only be driven during the day on a sunny day, and not under any other condition, because otherwise it may lead to crashes or accidents. Therefore, the need for a CNN to still segment an object under poorer conditions is a necessity. With this being said, there is a need for a robust CNN which can perform better under poorer conditions, while not degrading its performance on ideal conditions. The methods that were performed in order to achieve this are described in the following sections.

2. Related Work

Many attempts have been made at improving the performance of semantic segmentation using Convolutional Neural Networks. One of the earlier attempts was made by Long *et al*. [5] which consisted of doing convolutions on the entire image, some max pooling to downsample, to then up sample to match the original image. Using the entire image turned out to be computationally expensive, so then Girshick *et al*. [3] proposed using approximately 2000 regions around the image to detect possible objects. To further improve performance, Ren *et al*. [6] proposed the use of Region Proposal Networks(RPN), which greatly increased performance. To further decrease amount of parameters used, dilated convolutions were proposed by Chen *et al*. [2], which proves to be one of the state-of-the-art methods for semantic segmentation.

3. Methodology

One of the first items that the group became aware of was the selection of an appropriate dataset that could be used in such a way that it could be trained on images on ideal conditions, but also validated and tested on images with rainy or night conditions. That is why the Synthia dataset [1] proved to be the right tool for what was needed.

Table 1. Data split for experiments

Data Split		**Sequence**
Training		1,4,5 Summer
Validation	Good	20 % of Training
	Bad	4 Rain, 5 Fog
Testing	Good	2 Summer
	Bad	2 Night and Rain

Table 2. Description of experiments

Experiment #	Batch Size	Learning rate	Reg Loss	Size
1	10	0.001	No	1/4 Size
2	10	0.0001	No	1/4 Size
3	10	0.0001	Yes	1/4 Size
4	3	0.0001	No	Full Size

3.1. Initial Approach

It started by using 20 images as input to the network from the Synthia-Rand-Cityscape. After the input pipeline was sorted out with the appropriate batching of images and labels, the training was started with Adam optimizer and the sparse cross entropy softmax built-in function from Tensorflow. The first sanity check was to determine whether the loss was decreasing when training, which turned out to be the case. After that, the goal was to output the restored model from training with the corrected segmented images and prove whether the model was working as intended. To allow faster development times from training, the Google Colab tool was used.

3.2. Improvements

Furthermore, in order to allow for the experiments, several iterations were made at making a backbone architecture suitable for the project, such as Deeper Networks and more Shallow models, but the model did not seem to respond well to the architecture, and was not learning much. However, upon also trying Sandler et al [7], it was decided to focus more on other aspects of the network rather than only the architecture; thus, given time constraints, the Res-Net50 [4] was used which downsampled the image, to finally upsample it bilinearly to compare it with the labels. Other output strides in ResNet-50 were attempted, but there were persistent Resource Exhausted Errors, so an Output Stride of 8 was kept as the basis for our architecture.

Once the model turned out to work as intended with the input pipeline, optimization and restoring of the model, the following was to select how the datasets would be used. Synthia dataset also had sequences, so the sequences were split to validate our experiments which are shown in Table 1. With the data ready, several augmentation techniques were investigated which would make the network more robust against adverse conditions. In order to implement the data augmentation into the training, the built-in augmentation techniques from Tensorflow were used. These included included saturation, gamma, flip, and contrast in order to determine what effect augmentation has on the performance of the network.

With the augmentation techniques ready with the data, next step was the setup of Google Cloud Console rather than Google Colab given the advantages on computing power and GPU availability. The python scripts were modified to allow it to run on google cloud, and every training file was uploaded to the VM instance through the use of Google Cloud Buckets. Several models were restored with different hyperparameters such as varying dynamic learning rate, batch size, regularization use and the number of epochs being trained.

4. Experiments and results

Several experiments were made as described in Table 2. The experiments were made as to visualize what are the effects that batch sizes and different learning rates or regularization have on the network. Additionally, different augmentation techniques were used to describe its performance and Mean Intersection Over Union (MIOU) difference when compared with the Vanilla network.

As it can be seen in Figure 1, the experiment decoded to the original size of the image instead of 1/4 of the size presents a slight increase in MIOU accuracy. In contrast, Experiment 1 with the bigger learning rate, shows poor performance and shows no learning at all.

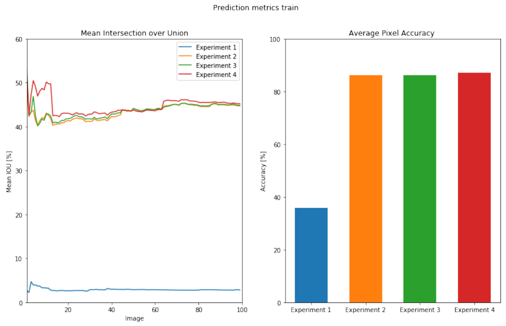


Figure 1. MIOU and Pixel Accuracy for Training Images with no augmentation

Additionally, in Figure 2, can be clearly seen how Experiment 4 with the original size image has the bigger increase in MIOU accuracy, and a clear difference as well for Pixel Accuracy for validation of the ”bad” images. Finally, with respect to the experiments, Experiment 4 yet again shows an increase on the MIOU Accuracy over testing images on ”bad” images, which in this case mean rain, as shown in Figure 4. Another clear performance gain can be seen in

MIOU accuracy of 45% over the training images with respect to the MIOU accuracy of 20 % over testing images which is expected due to being trained on these images and not on the testing images. Aside from Experiment 1, Experiment 2 and 3 show no clear difference in performance over any of the images.

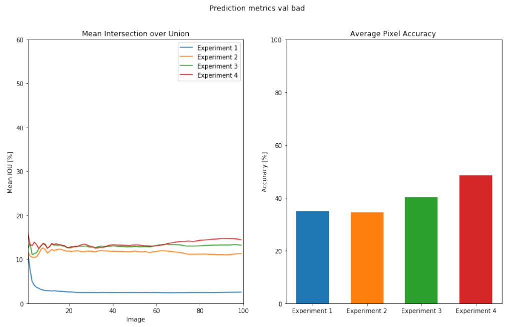
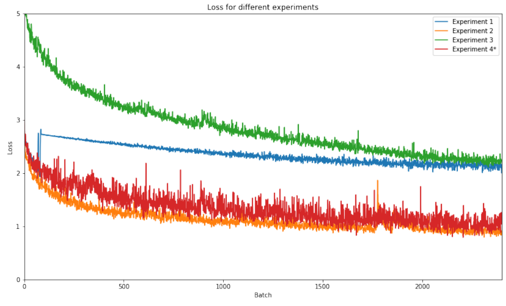


Figure 2. MIOU and Pixel Accuracy for Validation of ’Bad’ Images with no augmentation

Additionally, the losses in the 4 experiments with different batch sizes is seen in Figure 3. There is a clear trend in the loss decreasing, which proves the network is learning, except for Experiment 4 which stays roughly constant.



*batch = 3, so showing 3 epochs instead of 10 epochs

Figure 3. Losses for the four different experiments

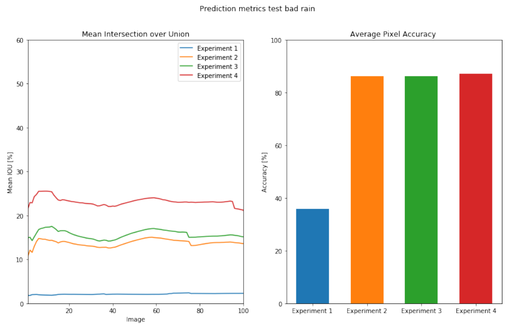


Figure 4. MIOU and Pixel Accuracy for Testing of ’Bad’ Images(rain) with no augmentation

Table 3. Overview of MIOU Accuracy for augmentation and no augmentation

	Vanilla	Flip	Saturation	Contrast	Gamma
Train	41.77	40.45	40.12	39.66	39.04
Val ’Good’	33.54	34.2	30.04	33.1	32.56
Val ’Bad’	12.0	11.87	12.84	11.97	10.65
Test ’Good’	27.6	28.78	25.75	26.64	26.37
Test ’Bad’ Night	13.9	12.81	13.86	14.29	10.58
Test ’Bad’ Rain	14.2	14.98	24.24	16.22	17.25

Aside from the experiments mentioned above, another set of experiments were performed but this time on a comparison with vanilla network vs augmentation techniques to show how a different augmentation affects the results on accuracy for different images. Every network was trained with images decoded to 1/4 of the size, batch size of 12 and learning rate of 0.0001 for 14 epochs.

In Figure 5,it can be seen that augmentation shows no result difference or increase in performance over the training images. In contrast, in Figure 6, Flip augmentation presents MIOU increase and pixel accuracy with respect to vanilla network, and also Saturation presents a decrease in performance of the network. Furthermore, there is a clear increase in MIOU accuracy as well on the testing ’bad’ images of rain with respect to the vanilla network as shown in Figure 7.

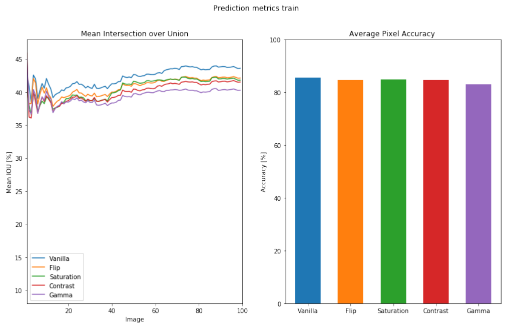


Figure 5. MIOU and Pixel Accuracy for augmentation vs no augmentation on ’Train Images’

Finally, Table 3 presents an overview of the MIOU accuracy for the different experiments presented here.

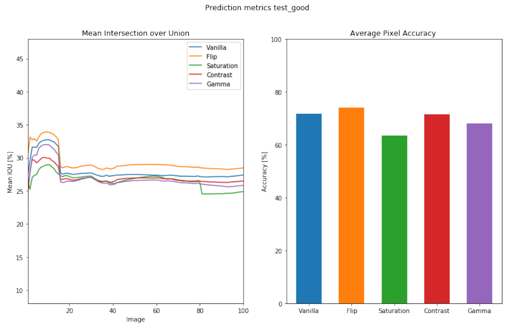


Figure 6. MIOU and Pixel Accuracy for augmentation vs no augmentation on 'Test good'

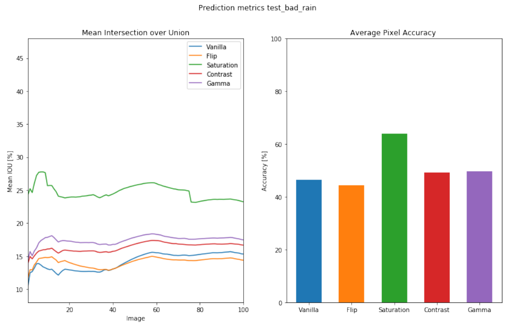


Figure 7. MIOU and Pixel Accuracy for augmentation vs no augmentation on 'Test bad rain'

As far as qualitative results go, Figure 8, shows the same image, but segmented under different conditions. It can be seen the difference in the segmentation results between a good summer image and a "bad" image. The street and lane markings are clear on the summer image, but there is a blob of everything on the image o the left.

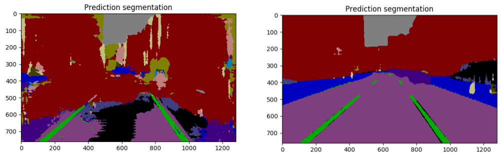


Figure 8. Sequence 4 Rain[left] and sequence 4 summer[right]

5. Conclusion

Segmenting and identifying objects in ideal conditions has shown to be a well resolved task, segmenting objects in poor conditions has proven to be more challenging. Consequently, the aim of the paper is to find out ways on which a network can be more robust against poorer conditions without degrading the performance of the network in good conditions.

As shown in the result sections, saturation and flip presented a considerable increase in MIOU accuracy in certain conditions, such as "Test Good" and "Test Bad Rain". On the other hand, augmentation made no difference on the train images. Although augmentation on the Synthia-Dataset resulted in several performance improvements over the Vanilla network, care has to be made as these images are Synthethic, and perhaps the increase could be explained by the way these "rain" or "night" images were generated.

Additionally, in order to draw more conclusions, it would help to do same experiments but on a set of real and more generalized images, such as CityScapes dataset. Images were trained on a Sequence, and there could be the possibility of the model "memorizing" the sequences.Finally, not all saturation techniques proved to increase performance, and even decreased performance in some aspects, so it indicates the proper augmentation technique varies depending on the type of image it sees.

References

[1] SYNTHIA Dataset. 1

[2] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. 2 2018. 1

[3] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. 11 2013. 1

[4] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. 12 2015. 2

[5] J. Long, E. Shelhamer, and T. Darrell. Fully Convolutional Networks for Semantic Segmentation. 11 2014. 1

[6] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. 6 2015. 1

[7] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 4510–4520. IEEE Computer Society, 12 2018. 2