Improvements on Robustness in Semantic Segmentation

*Final assignment for 5LSM0

Bo Yang 1796372

Abstract—**This essay aims to improve the robustness of convolutional neural networks(CNN) in semantic segmentation based on cityscapes dataset. Specifically to the two main aspects, which can influence the performance of CNNs significantly: robustness to degraded images and generalization for cities different from dataset, based on a predefined baseline, several solutions, including data augmentation, modification in structures, and other techniques have been taken to improve performance of model on these two aspects.**

Index Terms—**Quality degrade Robustness, CNN, GAN, Generalization, Segmentation**

I. Introduction

Semantic Segmentation is of great importance in automotive driving. For cityscapes, normally segmentation task can be performed relatively well by multiple CNN models for images with decent qualities, which means that the images are captured in good weather conditions, with good visibility and distinguishable objects. However, for bad weather conditions, like rain, snow, fog and night, which can be common in real lives, the visibility of objects in images shall be reduced, resulting variations in features and degrading in image qualities and eventually leading to worse performances. Similarly, the models trained with cityscapes set can have degraded performances on other cities not included in the dataset.

Several possible solutions have been proposed by the researchers. A video-segmentation approach is proposed, instead of conventional static image-segmentation methods. Video-segmentation approaches preserves temporal information from previous frames, enhancing robustness against possible noise in the current frame. these approaches can be especially effective for short-term disturbances. [1]. Another kind of solution is to use generative adversarial networks(GAN) and Cycle-GAN to generate images under different weather conditions and use these predicted data to augment the original dataset, then feed the enhanced data to the model, so that the robustness can be improved. [2] Or alternatively, use GAN to restore the image from bad weather conditions(rain for example). [3] Similarly, translating cityscapes data to different styles with CycleGAN [4] can make the model more generalized.

II. Baseline

A. Baseline: SegNet

According to my work in previous assignment, I chose SegNet as my baseline. SegNet was proposed in 2015, [2] it uses a classic encoder-decoder structure, suitable for multiple kinds of segementation tasks, including cityscapes.

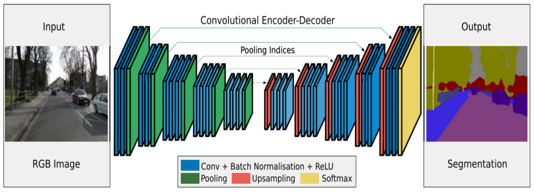

Fig. 1. The structure of SegNet proposed in [2].

The encoder on the left part are responsible for extracting the high-level features in the image,reducing the spatial dimensions. While the decoder layers use these features and try to recover the feature maps through trainable convolutional filter banks and upsampling layers.

A significant feature of SegNet is that it preserves pooling indices in the encoder layers during the downsampling process. These indices then will be used to upsample the feature maps in the corresponding upsampling layers. This process helps to preserve spatial information so that the segmentation can have a better performance.

Several attempts have been made to test its performance on validation set, degraded image quality and generalization. However, due to limitation of the attempts, baseline model was only tested once on image quality and twice on generalization. Thus 3 most representative attempts are presented in TABLE 1:

TABLE I
SegNet performance measured by Mean_Dice

Model Score	**Benchmarks**		
	Validation Set	***Image Quality***	***Generalization***
Attempt 1	0.27384	—	—
Attempt 2	0.39223	—	0.51067
Attempt 3	0.43668	0.26257	0.58671

Attempt 1 is the direct result without using any techniques(baseline). Attempt 2 and Attempt 3 used some data augmentation techniques and class weight vector stated in section III, and were trained on the original cityscapes dataset without using images generated by CycleGAN.

B. Problems of the baseline

The SegNet model can be effective and competitive, for normal cases. With some training techniques and data augmentation, the performance on the validation set was improved. Also, using images with a bigger size, the model can also produce better results since the quality of the data is improved. However, as is shown in the table, the model can be quite vulnerable to disturbances caused by bad weather conditions, leading to poor performance, even for attempt 3, in which case, the model was enhanced with augmented data and fed by images with high resolutions. Common techniques for improving performances, like tuning hyperparameters, data augmentation, changing loss functions…, could hardly make any further improvements, besides, due to limitation of memory of GPU, the resolutions of data cannot be further improved.

Besides, its encoder part using only pooling to downsample the feature maps, some valuable features might also be discarded.

Since bad weather conditions are similar to adding more unexpected noise to the original image, like variations in visibility, brightness, and contrast, which can be similar to some common augmentation techniques, so add more variations in the training data can be helpful. However, based on my attempts, further data augmentation for SegNet can hardly further improve the performance, actually, the model had low accuracy on greatly augmented data, thus it is believed that the normal SegNet model is not complex enough for noisy data in bad weather conditions. Also, models with higher complexity can handle harder prediction problems and wider range of variations in styles if trained properly, which can be suitable for generalization problem.

C. General ideas for solutions

Eventually, two general directions were decided according to my experiments on baseline model:

- Increasing the model's complexity. Models with higher complexity can be more capable to handle harder problems and larger dataset.
- Adding more variations in training data, which are specific to the problems: bad weather conditions and generalization. The model can be more likely to have better performance if fed with data more related to the cases present in the problem.

III. Improvements to baseline and solutions taken

A. Using CycleGAN to create more samples

Instead of using conventional data augmentation techniques, like cropping, resize, rotation…, one can make the data more specific to the problem. It is possible to translate normal images to images under specific weather conditions using CycleGAN, a special variation of generative adversarial network(GAN) [2].

In the experiment, generated images under 3 specific conditions are added to the dataset, along with their originals. These 3 conditions are rain, snow, and night. The new dataset now has a larger training set, with more variations in weather conditions, allowing the model to extract more general information robust against bad weather conditions. I think this can be a proper solution to the bad weather conditions, for that this shares the same idea with normal data augmentation, but more specific. By adding relevant instances to dataset, the model can extract the features of classes in different weather conditions and make the information extracted more generalized and representative, so that the robustness against degrading image quality, caused by bad weather conditions can be improved.

[2] proposed an effective way to generate images under different weather conditions as training data using CycleGAN and GAN.

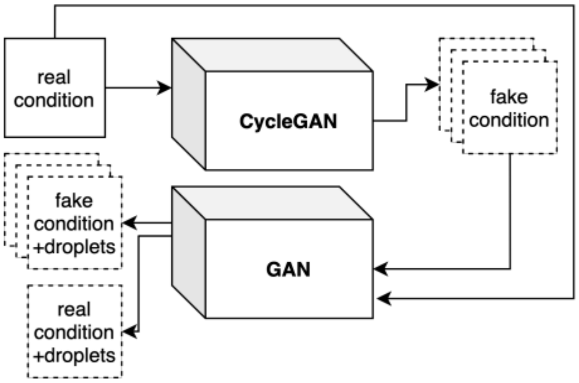

Fig. 2. The method schematic proposed in [2].

According to [2], this method consists of two stages. In the first stage, The CycleGAN, composed by N models, receives real images as input, and outputs the generated images with N different weather styles.In the second stage, another GAN model can be used to add rain drops to both the generated images and real images, making the outputs more similar to conditions in real life.

Fig. 3 shows a image under 3 bad weather conditions generated by the methods in [2] and actually used in the training process.

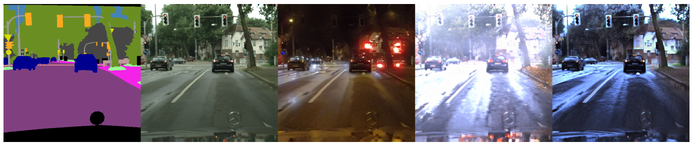

Fig. 3. GroundTruth, Original image, night, rain, snow

B. Increasing model complexity: DeepLabV3plus with Xception

Since the dataset was greatly expanded, as stated in section II.B, SegNet might not be complex enough to learn all necessary features, and also, due to the limitation of feature extraction methods used in SegNet, some feature information (low-level for example) cannot be preserved. I chose to use a more complex and powerful model: DeepLabV3plus.

This model was proposed in 2018 [5], also having a encoder-decoder structure like SegNet. However, DeepLabV3plus have different feature extraction and transfer methods and more flexible convolution algorithm. It introduces the skip-connection part between the encoder and decoder, allowing both global and local features to be extracted and transferred. Besides, several new techniques, including dilated convolution and spatial pyramid pooling, enable the model to capture feature information and context in multiple scales and handle variations of objects in different conditions. This network has been proved to be a powerful and effective model.

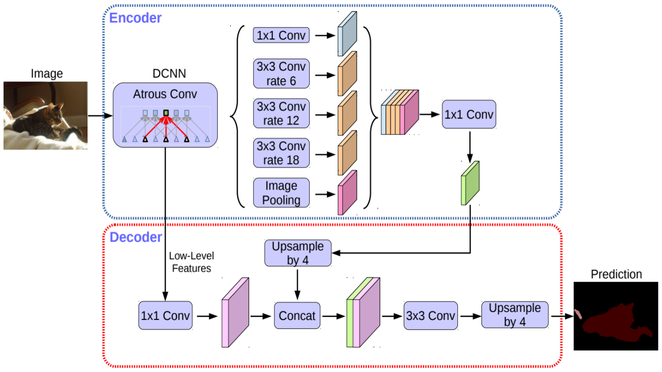

Fig. 4. The general structure of DeepLabV3+ proposed in [5].

It is noted that the DeepLabV3+ requires a backbone network in DCNN to extract features. Several options available: MobileNet, ResNet and Xception.

There is a trade-off to be made here:

- MobileNet: A series of lightweight neural networks, has the lowest computational complexity, suitable for tasks with limited storage and memory in devices.
- Xception: A new network showing efficiency in computation and also maintaining a good performance in feature extraction.
- ResNet: A set of complex neural networks, with a good performance in feature extraction, can be relatively computational expensive.

For this task, I chose Xception as backbone, for that it can maintain a balance between feature extraction capability and computation cost.

[5] proposed a modified Xception structure, shown in Fig. 5. This modification makes the network have a better performance.

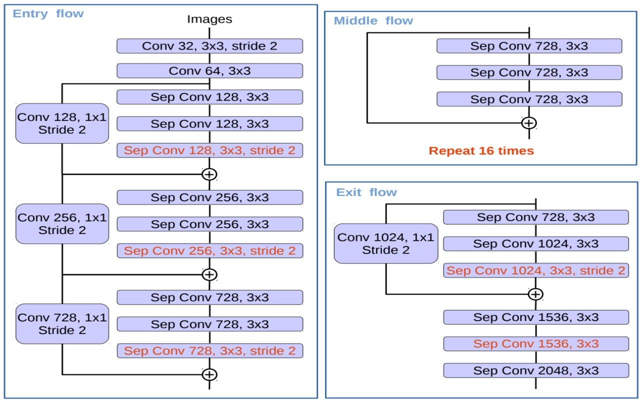

Fig. 5. modified Xception proposed in [5].

C. Conventional Data Augmentation

After the previous two methods, some common techniques of data augmentation were also applied:

- Horizontal flip: Flip the image horizontally. The probability is 0.8. Used in SegNet.
- Vertical flip: Flip the image vertically. The probability is 0.8. Used in SegNet.
- Random resize crop: Crop a random part of the image and resize it to the original size. The probability is 0.8. This is very effective in segmentation tasks, for that it can produce samples with objects in different scales. Taking longer time of training. Useful in SegNet, not used in DeepLabV3+ due to limited time.
- Color jittering: Randomly adjust the contrast, brightness, saturation and hue. Useful in DeepLabV3+ for that this method can be similar to creating different weather conditions.

D. Class weights specific to cityscapes dataset

This method is specific to the cityscapes dataset, for that it is unbalanced. The 19 classes to be predicted have different frequencies in the data. Some of the classes can appear more frequently and have larger sizes in the images, like roads and sky. while some classes have small sizes and are less common, like traffic signs. In order to compensate that, a weight can be assigned to each class. Similar methods have been applied in [6]. The weights can be computed based on the median frequency balance [6].

In the assignment, a weight vector was assigned to the cross-entropy loss function to compensate for the imbalanced distribution of classes.

E. Results of implementation

Based on the solutions proposed in this section, two attempts were taken in the competition server and shown in TABLE II.

Combine TABLE I with TABLE II, it can be concluded that Better performances in attempt 4 and 5 were contributed by DeepLabV3+, which is more powerful and advanced model

TABLE II
DeepLabV3+ with other solutions measured by Mean_Dice

Model	Benchmarks	
Score	***Image Quality***	***Generalization***
Attempt 4	0.32853	0.61708
Attempt 5	0.3323	0.6186

than SegNet, with additional training data generated by CycleGAN.

The significant improvements in Image Quality, (from 0.26257 in attempt 3, TABLE I to 0.3323 in attempt 5 TABLE II) indicates that the extra data provided by CycleGAN can be especially effective in this benchmark. As for the generalization, the DeepLabV3+ model also showed better performance.

Also, for the SegNet, data augmentation techniques can also produce considerable improvements in performance, however, due to limitation of model complexity, further improvements can hardly be made through pure data processing without modifying network structures.

In conclusion, all the solutions proposed in section III can be effective in improving performance. Some of them are especially useful for certain benchmarks.

IV. Limitations & options for further improvements

A. Generated data not representative enough

I only added images under 3 bad conditions: snow, night and rain. However, there are also other bad weathers conditions (like fog) leading to degrading qualities in images, which can also be used during the evaluation process. I think the model will have better performance if I can feed it with data under more bad conditions and from different cities. However, due to the limitation of time and computation resources, I could only use the data provided in [2]. Different data mean training the CycleGAN and GAN in different ways, which can take much time and computation resources.

Similarly, for generalization problem, I only used the data generated on the basis of the original cityscapes dataset, without changing the city styles. The performance in generalization shall be improved if I can use GAN to produce data with different city styles and use them to train the model.

B. New backbones available

Backbones serve as feature extractor in DeepLabV3+ structure, there are some other new networks which can serve as the backbone. For example, efficientNet [7] , showing better performance than Xception and having much fewer parameters. Using more powerful networks as backbones can surely improve the performance.

C. Limited data qualities

The data provided in [2] have a fixed size:512x512, which were acquired by cropping from the larger size(512x1024), instead of directly resizing the original images. This indicates that there can be differences in the contents of data from the original images. Besides, the resolutions of these data were also reduced.

The model will have a better performance, if fed with the data acquired in same way as the evaluation process, using resizing directly instead of cropping. Besides, higher resolutions in the data can also be helpful.

D. More augmentation methods available

Due to the limitation of time and computation resources, I didn't use some augmentation techniques when training DeepLabV3+. For instance, Random resize crop is very effective in segmentation task, which greatly improved the performance of SegNet(shown in TABLE I). However, when training DeepLabV3+ with expanded training set, this augmentation method will require much longer training time and resources, thus it was not implemented in the training process of DeepLabV3+. Similarly, if I have more training time, rotation, flip can also be added to improve the performance.

E. Additional training tricks available

Using pretrained weights for the backbone network would also improve the performance for that by doing so the backbone network can be more capable of extracting features. This method was not considered at first.

Besides, other training techniques, like weight decays, learning rate decays and optimizing the downsample factor for DeepLabV3+ would also be effective.

References

[1] A. Pfeuffer and K. Dietmayer, "Robust semantic segmentation in adverse weather conditions by means of fast video-sequence segmentation," 2020.

[2] V. Mușat, I. Fursa, P. Newman, F. Cuzzolin, and A. Bradley, "Multi-weather city: Adverse weather stacking for autonomous driving," in *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pp. 2906–2915, 2021.

[3] H. Porav, T. Bruls, and P. Newman, "I can see clearly now : Image restoration via de-raining," 2019.

[4] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

[5] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," 2018.

[6] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[7] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," 2020.