Improving semantic segmentation under severe weather conditions : based on Bilateral Segmentation Network

Yancheng Cai
School of Information Science and Technology
Fudan University
220 Handan Road
19307140030@fudan.edu.cn

Lesi Chen
School of Data Science
Fudan University
220 Handan Road
19307130195@fudan.edu.cn

Zhiyang Liang
School of Data Science
Fudan University
220 Handan Road
19307130184@fudan.edu.cn

Abstract

For this mid-term project, our team uses the Bilateral Segmentation Network(BiSeNet) to handle the semantic segmentation task. Semantic segmentation requires both rich spatial information and sizeable receptive field. In real application scenarios such as autonomous vehicles or agricultural real-time monitoring, robustness to the complex and changeable environments is highly demanded, too. However, modern approaches usually compromise spatial resolution to achieve real-time inference speed, which leads to poor performance. In this project, we handle this dilemma with a Bilateral Segmentation Network Version 1 (BiSeNetV1) and we also implement other famous networks(like segnet,unet and pspnet) for comparison. The architecture of the network makes a right balance between the speed and segmentation performance. We get 76.2% Mean IOU in the original Cityscapes dataset with speed of 128 FPS. And after changing the 19 categories into 2 categories(the road and the background), we achieve 97.4% Mean IOU with speed of 104 FPS in the modified Cityscapes dataset, which is surely ideal for recognizing roads for autonomous driving. Furthermore, we find that it also has similar excellent performance in dense fog scenes, with 96.5% Mean IOU with a speed of 125 FPS in the modified Foggy-cityscapes dataset.

1 Introduction

The research of semantic segmentation, which amounts to assign semantic labels to each pixel, is a fundamental task in computer vision. It can be broadly applied to the fields of augmented agricultural monitoring, autonomous vehicles, and video surveillance. These applications have a high demand for efficient inference speed for fast interaction or response.

To meet the high demand for both the enough spatial information and real-time inference speed, the Bilateral Segmentation Network (BiSeNet [9]) proposes two parts: Spatial Path and Context Path. As their names imply, the two components are devised to confront with the loss of spatial information and shrinkage of receptive field respectively. The design philosophy of the two paths is clear. For Spatial Path, it stacks only four convolution layers to obtain the feature map, which retains affluent spatial details. In respect of Context Path, it appends a global average pooling layer on the tail of ResNet18[3], where the receptive field is the maximum of the backbone network. We will introduce both of the paths more specifically in the following parts.

Inspired by the video about autonomous vehicles in class, in which the sensor of the cars

only divides the road from any other kinds of things. So after finishing the basic task of semantic segmentation in the Cityscapes[2] dataset, we do some modifications to the dataset: changing the original 19 categories into 2 categories(the 'road' and the other 18 into 'background'). More than that, we also try it out on the Foggy-cityscapes[7] dataset in order to validate its robustness to severe weather conditions and figure out whether it has the capability to tackle the rapidly changing real environments. In other words, we want to know whether it is able to be well adaptable and respond quickly to external changes.

2 Related Works

For comparison, we implement many famous networks used for semantic segmentation. Most of these methods are designed to encode more spatial information or enlarge the receptive field. In this section, we will extract their common features and introduce them to the readers.

2.1 Spatial information

The convolutional neural network (CNN) encodes high-level semantic information with consecutive down-sampling operations. However, in the semantic segmentation task, the spatial information of the image is crucial to predicting the detailed output. For example, PSPNet[11] use the dilated convolution to preserve the spatial size of the feature map. Global Convolution Network utilizes the 'large kernel' to enlarge the receptive field.

2.2 Context information

Semantic segmentation requires context information to generate a high-quality result. The majority of common methods enlarge the receptive field or fuse different context information. For example, PSPNet applies a 'PSP' module which contains several different scales of average pooling layers.

2.3 U-shape method

The U-shape structure can recover a certain extent of spatial information. The original FCN[4] network encodes different level features by a skip-connected network structure. Some methods employ their specific refinement structure into U-shape network structure. For example, U-net[6] introduces the useful skip connection network structure for this task. However, in the U-shape structure, some lost spatial information cannot be easily recovered.

2.4 Real-time segmentation

Real-time semantic segmentation algorithms require a fast way to generate the high-quality prediction. SegNet[1] utilizes a small network structure and the skip-connected method to achieve a fast speed. E-Net[5] designs a lightweight network from scratch and delivers an extremely high speed.

Although these methods can achieve a real-time inference speed, they dramatically sacrifice the accuracy to the efficiency with the loss of the low-level details.

3 Bilateral Segmentation Network

In this section, we will introduce the Bilateral Segmentation Network, the network we use to handle our mid-term task. Unfortunately, the official code of BiSeNetV1 is not made public available, so we get another one from the Github, which is a little different from the official network. And we also did some modifications to it. First, we will illustrate the two parts of the network. Then, we will show the whole network architecture of the community version of BiSeNetV1.

3.1 Spatial path

In the task of semantic segmentation, some methods try to preserve the resolution of input image to encode enough spatial information with dilated convolution, while others attempt to

capture sufficient receptive field with pyramid pooling module, atrous spatial pyramid pooling or 'large kernel'. These indicate that both the spatial information and the large receptive field are of great significance for achieving high accuracy. However, it is nearly impossible to meet the two requirements simultaneously, especially for the real-time semantic segmentation.

In order to solve the dilemma decribed above, the Spatial Path is proposed to preserve the spatial size of the original input image and encode affluent spatial information. The Spatial Path contains four layers. Each layer includes a convolution, followed by batch normalization and ReLU. It encodes rich spatial information due to the large spatial size of feature maps.

3.2 Context path

While Spatial Path encodes affluent spatial information, the Context Path is designed to provide sufficient receptive field. To enlarge receptive field, some methods have taken advantage of the pyramid pooling module, atrous spatial pyramid pooling or 'large kernel'. Unfortunately, these operations are computation demanding and memory consuming, which results in low speed.

To get large receptive field and efficient computation simultaneously, the Context Path is proposed. It uses lightweight model and global average pooling to provide large receptive field. In this work, the lightweight model, like ResNet18, can downsample the feature map fast to obtain large receptive field, which encodes high level semantic context information. Then we add a global average pooling on the tail of the lightweight model, which can provide the maximum receptive field with global context information. Finally, we combine the up-sampled output feature of average value and the features of the lightweight model.

We also propose a specific **Attention Refinement Module** to refine the features of each stage, as is shown in Figure 1. It uses the convolution, batch normalization and ReLU to capture global context and computes an attention vector to guide the feature learning. This design can refine the output feature of each stage in the Context Path. It integrates the global context information easily without any up-sampling operation. Therefore, it demands negligible computation cost.

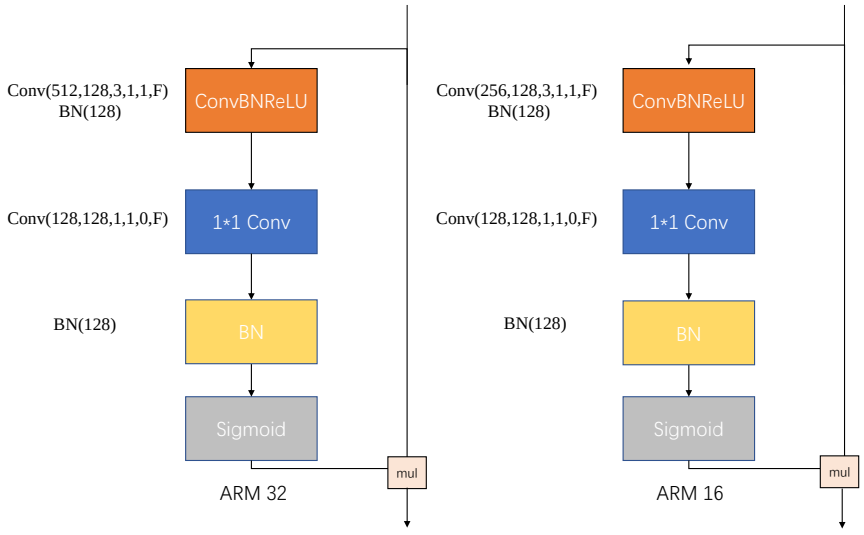


Figure 1: The Attention Refinement Module

The parameters are given in the format as follows:
Conv(inputC,outputC,kernel_size,stride,padding,bias)
BN(outputC)

3.3 Network architecture

With the Spatial Path and the Context Path, we propose BiSeNet for real-time semantic segmentation. We use the ResNet18 model as the backbone of the Context Path and four convolution layers as the Spatial Path. And then we fuse the output features of these two paths to make the final prediction. It can achieve real-time performance and high accuracy at the same time. First, we focus on the practical computation aspect. Although the Spatial Path has large spatial size, it only has four convolution layers. Therefore, it is not computation intensive. As for the Context Path, we use a lightweight model to down-sample rapidly. Furthermore, these two paths compute concurrently, which considerably increase the efficiency. Second, we discuss the accuracy aspect of this network. In our paper, the Spatial Path encodes rich spatial information, while the Context Path provides large receptive field. They are complementary to each other for higher performance.

Finally, we use the **Feature Fusion Module** to combine the feartures we get from the Spatial Path and the Context Path, as is shown in Figure 2. The features of the two paths are different in level of feature representation. Therefore, we can not simply sum up these features. The spatial information captured by the Spatial Path encodes mostly rich detail information. Moreover, the output feature of the Context Path mainly encodes context information. In other words, the output feature of Spatial Path is low level, while the output feature of Context Path is high level. Therefore, we propose a specific Feature Fusion Module to fuse these features.Given the different level of the features, we first concatenate the output features of Spatial Path and Context Path. And then we utilize the batch normalization to balance the scales of the features. Next, we pool the concatenated feature to a feature vector and compute a weight vector, like SENet. This weight vector can re-weight the features, which amounts to feature selection and combination.

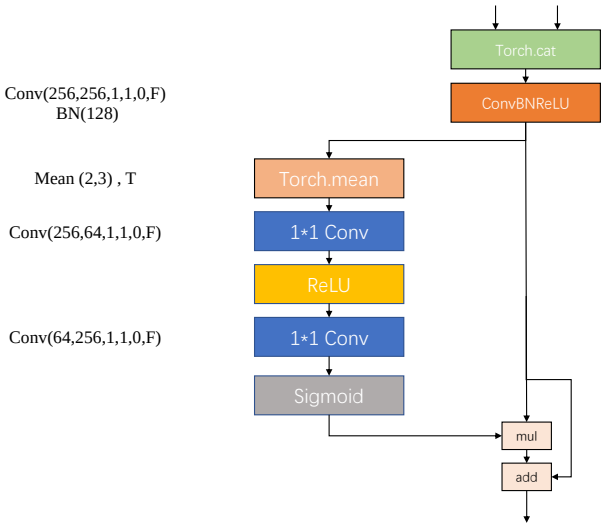


Figure 2: The Feature Fusion Module

The parameters are given in the format as follows(Things mentioned above will not be repeated):
Mean(dim,keepdim)

The Figure 3 shown below is the whole view of the Bilateral Segmentation Network.

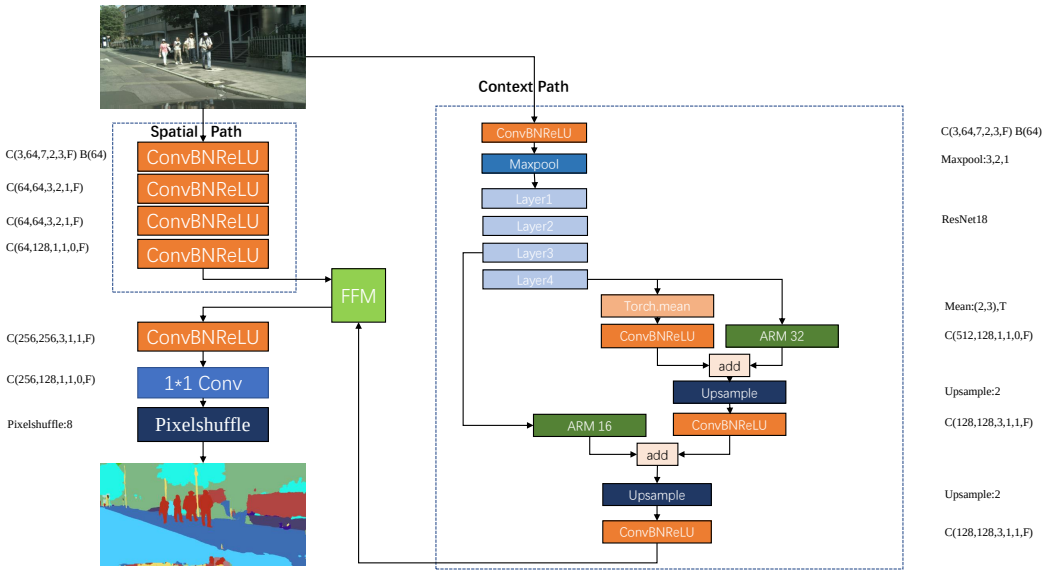


Figure 3: The Whole Network Structure

The parameters are given in the format as follows(Things mentioned above will not be repeated):
Maxpool(kernel_size,stride,padding)
Upsample(factor)
Pixelshuffle(factor)

4 Training details

In this section, we train and evaluate our network on two different datasets. Also, we will do some modifications to the datasets in order to train our own network more efficiently. We first introduce the datasets, then we will try our best to show the details of our training. Finally, we show our results including Mean IOU and speed in the Cityscapes dataset as finishing the basic task of this project.

4.1 Datasets

Cityscapes: The CityScapes is a large urban street scene dataset from a car perspective. The dataset includes 5000 manually selected images from 27 cities, which are collected with a similar weather condition. These images are annotated with dense pixel-level image annotation. However, because the labels are manually done, there are many mistakes in the Cityscapes dataset. We remove some images of the dataset to avoid the mistakes in order to get better performance. After finishing the basic task of semantic segmentation, we restrict the label to 'background' and 'road' to explore the autonomous driving issues that our team is interested in. In the end, we have 2264 images as our traing set and 392 images for our validation set. These images all have a resolution of 1,024*2,048, in which each pixel is annotated to the pre-defined 2 classes.

Foggy-cityscapes: Foggy-cityscapes dataset simulates the foggy weather based on the Cityscapes. The pixel-level labels of Cityscapes can be inherited by the foggy-cityscapes such that we can do the same modifications to the Foggy-cityscapes dataset as we do to the Cityscapes dataset. In the our experiments, we have 2264 images as our traing set and 392 images for our validation set.

4.2 Data augmentation:

We apply the mean subtraction, random horizontal flip, and random scale on the input images to augment the dataset in training procedures. The scales contains {0.75, 1.0, 1.5, 1.75, 2.0}. Finally, we randomly crop the images into fixed size for training.

4.3 Loss function

OHEM method:OHEM[8] is short for the 'online hard example mining'. The core idea of the algorithm is: according to the loss of the input sample, mine out the hard example, which means the samples that have a greater impact on the classification and detection, and then apply the selected samples in the stochastic gradient descent training. Generally, the algorithm has two advantages:
1) The problem of unbalanced data categories does not need to be solved by setting the ratio of positive and negative samples. This online selection method is more targeted;
2) As the scale of dataset increases, the algorithm improves more obviously. Especially, it's suitable for situations where the batch size is small, but there are many examples of each image.

We implement this by setting a threshold of loss, when the loss is less than the threshold we set, it is a positive sample; when the loss is greater than the threshold, it is a negative sample. We set it as 0.7.

Auxiliary loss: We also implement auxiliary loss function to supervise the training process. We not only set a loss function for the output of the entire network, we also set two loss functions to supervise the out put of the Context Path. That's the output of ResNet18 Layer3 and Layer4, we denote them as feature_16 and feature_32. All of these three loss functions are Cross Entopy Loss, as Equation (1) shows. Moreover, we use the parameter α to balance the weight of these three loss, as Equation (2) shows. Finally, we set α to 1. The joint loss makes optimizer more comfortable to optimize the model.

$$Loss = \frac{1}{N}\sum_{i} -log(\frac{e^{p_i}}{\sum_j e^{p_j}}) \tag{1}$$

$$L(X;W) = l_p(X;W) + \alpha l_{f\_16}(X_{f\_16};W) + \alpha l_{f\_32}(X_{f\_32};W) \tag{2}$$

We denote l_p as the principal loss of the concatenated output. And L is our joint loss function.

4.4 Model training

Warm-up strategy: At the beginning of training process, we apply a small learning rate so that the network has enough time to get familiar with the data. As the training progresses, the learning rate gradually increased. To a certain iteration, the training is performed at the set initial learning rate, and then we begin our main training strategy. We set the $warmup\_step$, when the $step$ is less than it, the $learning\_rate$ is equal to $initial\_rate * \frac{current\_step}{warmup\_step}$ When the warmup finished, we begin our main strategy.

Also, the warm-up strategy has two advantages:
1) It helps to slow down the early overfitting of mini-batch in the initial stage of the model, and keep the distribution stable.
2) When the network is very deep and complex, it is of great help in maintaining the stability of our model.
We choose the initial warm-up learning as 1e-5, the warm-up step as 1000.

Poly strategy: We use the poly learning rate strategy in which the initial rate is multiplied by $(1 - \frac{iter}{max\_iter})^{power}$ in each iteration with a certain power. We set the initial learning rate at 1e-2, the power at 0.9.

Other details: We use mini-batch stochastic gradient desent(SGD) with batch size 2, momentum 0.9 and weight decay 5e-4 in training. Our max iteration is 80000.

4.5 Results

First we show our loss curve by tensorboard in Figure 4. We use blue to represent the training set loss, and red to represent the validation set loss. From the curve we can see that our model finally converged, and the overfitting is not obvious.

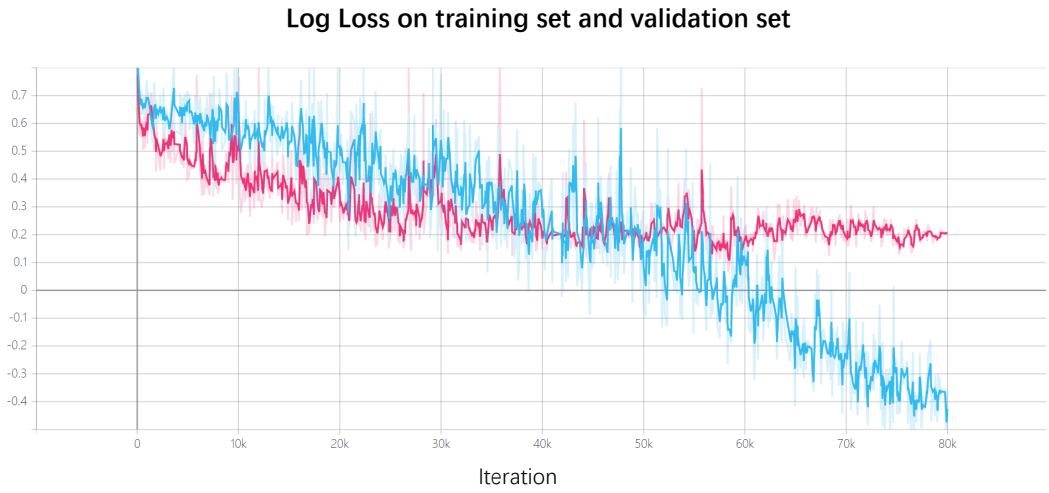
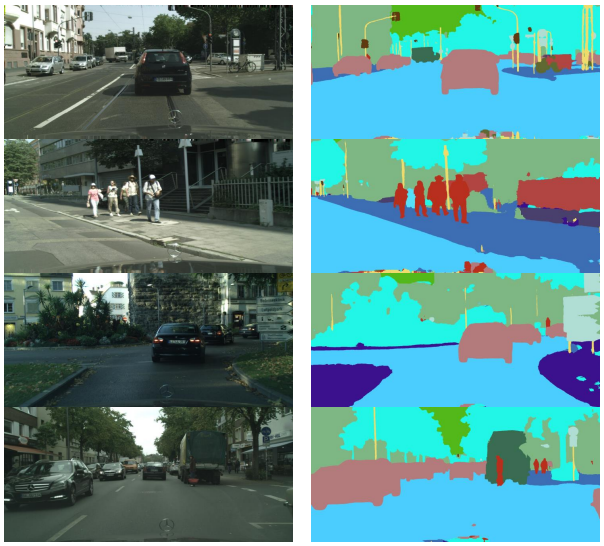


Figure 4: The Loss Curve

Then we visualize some of our images in the original Cityscapes dataset, as is shown in Figure 5. We achieve 76.2% Mean IOU in the original Cityscapes dataset with speed of 128 FPS, which is pretty good.



(a) Input Image (b) Output by BiSeNet

Figure 5: The Result of BiSeNet

5 Extra Experiments

In this section, we do the modifications we mentioned in the introduction part to the two datasets and explore the topics on autonomous driving which our team is interested in.

Next, we describe our speed strategy in comparison with other methods in detail. Finally,we try to change the structure of our network to investigate the effects of each component of our proposed approach .

5.1 Foggy Environment

We relabel the dataset into 2 categories: the 'road', and other 18 into the 'background'. **And now we just calculate the IOU with 'road'.** Similarly, we train our network on the two datasets, and the results are shown in Figure 6.

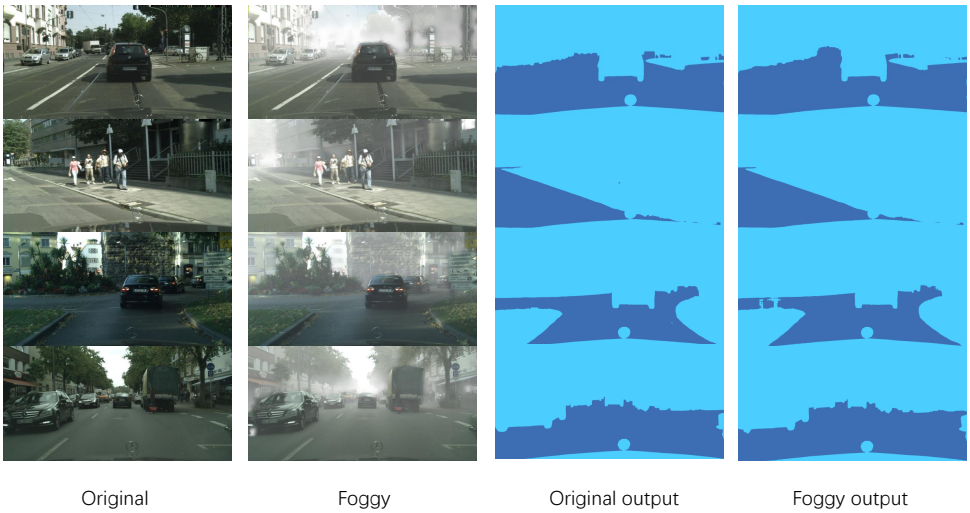


Figure 6: The Result in both datasets

We achieve 97.4% Mean IOU with speed of 104 FPS in the modified Cityscapes dataset, which is surely ideal for recognizing roads for autonomous driving. Furthermore, we find that it also has similar excellent performance in dense fog scenes, with 96.5% Mean IOU with a speed of 125 FPS in the modified Foggy-cityscapes dataset.

However, such tests alone are not enough to prove the robustness of our network. We will discuss the advantages of our network compared to other networks in the next subsection.

5.2 Comparison study

We implement some famous networks which have high performance in semantic segmentation. The accuracy and speed comparison of our method against other methods on Cityscapes dataset is shown in Table 1.

Table 1: In the Cityscapes Dataset

Method	Mean IOU (%)	FPS
SegNet	84.8	62
FCN8	90.5	134
PSPNet	80.8	23
U-Net	77.2	66
ICNet [10]	86.3	12
BiSeNet	97.4	104

The accuracy and speed comparison of our method against other methods on Foggycityscapes dataset is shown in Table 2. We can see that compared to normal weather, the performance of each network is more or less degraded in dense fog. However, the overall network speed has not changed much, or even increased. We speculate that the thick fog is artificially added, which may cause the loss of the spatial information of the picture, and the time required for calculation becomes shorter.

Table 2: In the Foggycityscapes Dataset

Method	Mean IOU (%)	FPS
SegNet	89.4	52
FCN8	80.9	119
PSPNet	79.5	27
U-Net	78.5	76
ICNet	85.1	14
BiSeNet	96.5	125

It is worth noting that it is difficult for a network to have high speed and high accuracy at the same time. So the architecture of BiSeNet is indeed a milestone in semantic segmentation.

In order to be able to understand these data more intuitively, we also visualized some result pictures, as are shown in Fig 7 for Cityscapes and Fig 8 for Foggycityscapes. We have three things to note:
1)For a large area like the road, even a small change in Mean IOU will make a huge difference.
2)If the road interruption is detected by mistake, it is fatal for autonomous driving, because it may cause the car to fail to drive normally on the established route, resulting in low traffic efficiency.
3)Robustness to complex weather is also very important, because self-driving cars are needed at any time, especially in bad weather conditions, when we are more willing to sit in vehicles.

Although our network has a certain performance loss, its performance in dense fog is basically the same as in normal weather. However, ICNet and SegNet make serious segmentation errors in the Foggycityscapes dataset. In summary, our network is relatively robust.

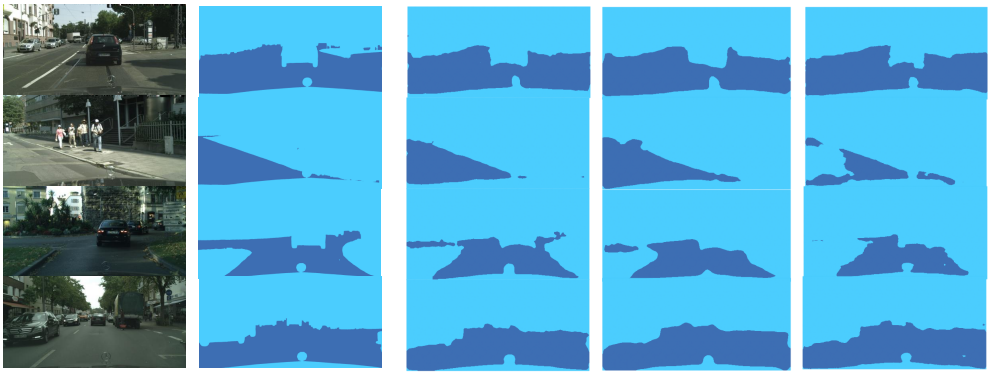
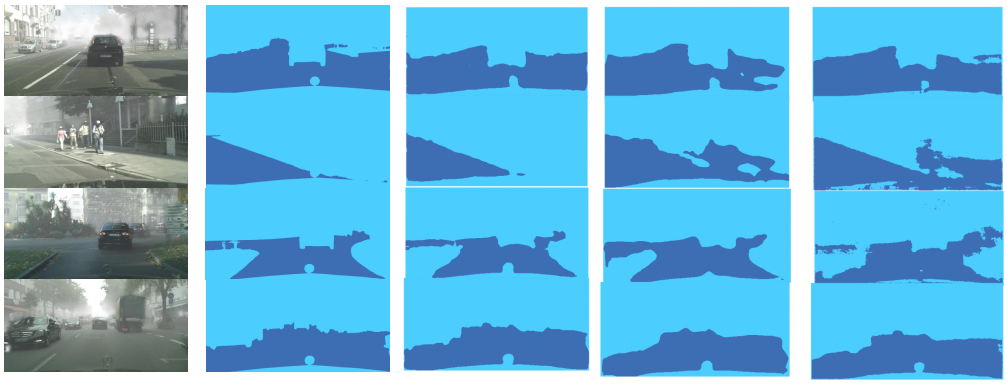


Figure 7: Cityscapes



(a) Image (b) BiSeNet (c)FCN8 (d) ICNet (f)SegNet

Figure 8: Foggycityscapes

5.3 Ablation study

In this subsecton, we detailedly investigate the effect of each component in our BiSeNet.

Ablation for Spatial Path: Existing modern approaches of real-time semantic segmentation task face the challenge of lost of spatial information. Therefore, we propose a Spatial Path to preserve the spatial size and capture rich spatial information. This improves the performance from 97.2% to 97.4%, as is shown in Table 3. The Spatial Path encodes abundant details of spatial information.

Ablation for FPN:FPN is the short for 'Feature Pyramid Network'. As you can see in Figure 3, FPN is a significant part of our Context Path. With FPN, we can capture more refined semantics, which improves our network's performance.

Table 3: Detailed performance comparison

Method	Mean IOU (%)
CP	96.5
CP + FPN	97.2
CP + FPN +SP	97.4

This shows that every part of our network helps to improve performance. Due to the limited time, we don't do a more in-depth exploration.

6 Conclusion

Bilateral Segmentation Network (BiSeNet) is able to improve the speed and accuracy of real-time semantic segmentation simultaneously. The Spatial Path is designed to preserve the spatial information from original images. And the Context Path utilizes the lightweight model and global average pooling to obtain sizeable receptive field rapidly. With the affluent spatial details and large receptive field, we achieve the result of 76.2% Mean IOU in the original Cityscapes dataset with speed of 128 FPS. Furthermore, we find it robust to harsh environments, which demonstrates the potential of the network in areas such as autonomous driving.

References

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[2] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[4] 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.

[5] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, 2016.

[6] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[7] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision*, 126(9):973–992, 2018.

[8] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 761–769, 2016.

[9] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 325–341, 2018.

[10] Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia. Icnet for real-time semantic segmentation on high-resolution images. In *Proceedings of the European conference on computer vision (ECCV)*, pages 405–420, 2018.

[11] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.