Image Processing and Computer Vision (ENGG5104)
Final Project – Spring 2022

ZHANG Xiaojun
1155181655
1155181655@link.cuhk.edu.hk

Abstract

In this project, we have implemented a deep neural network for semantic segmentation. At first, the PSPnet [5], which is one of the most popular models for semantic segmentation task, is implemented using ResNets as backbone. Then, the ASPP module [1] and the OCR module [4] are applied separately to improve the performance of the model. Next, an efficient CNN model-mobilenetv2 [3] is used as backbone to improve the inference speed. Finally, in the open challenge part, we have tried the PSPnet with ResNet of 101 layers and successfully increased the mIoU value to a higher level.

1. Task 1

Implementation Details As described in [5], the basic principle of PSPnet is extracting feature maps from the dilated ResNet and then enriching these features by PPM module. In this part, resnet18 and resnet50 are performed as backbone to extract the features at first. Next, the four-level Pyramid Pooling Module is implemented to extract the regional contextual information. Since the structure of resnet18 and resnet50 is different, some implementation details will vary slightly. The way we use resnet18 is as follows: the extracted feature dimension is 512; for the conv1 in layer3, the parameters are modified with dilation (2, 2) , padding (2, 2) and stride (1, 1); for the conv1 in layer4, the parameters are modified with dilation (4, 4) , padding (4, 4) and stride (1, 1); the downsample0 is replace with the stride (1, 1). As for resnet50, the implementation details are as follows: the extracted feature dimension is 2048; for the conv2 in layer3, the parameters are modified with dilation (2, 2) , padding (2, 2) and stride (1, 1), and the downsample0 is replaced with the stride (1, 1); for the conv2 in layer4, the parameters are modified with dilation (4, 4) , padding (4, 4) and stride (1, 1), and the downsample0 is replaced with the stride (1, 1). After finishing the part used to extracting features map, we begin to build the PPM module[1]. The input feature dimension is set to 512 for resnet18 backbone and 2048 for resnet50 backbone separately (which are the same as the ResNet outputs). The bins are set as [1, 2 , 3, 6]. For each level feature extraction, the sequence of modules is as follows: DaptiveAvgPool2d, Conv2D, BatchNorm2d, ReLU. The size of scaled pyramid features will be different from that of original input. Therefore, we use the F.interpolate function to resize it. Then, they will be concatenated together. Finally, a classification layer will be use to predict the classes for each pixel. To speed up the training, an auxiliary layer is added to calculate the auxiliary loss. And the input dimension is 256 for resnet18 and 1024 for resnet50.

Results & Analysis The validation result is shown in Table 1 and the representative segmentation results are shown in Fig. 1. From these results, we can see that compared to resnet18, the mIoU valued of resnet50 is much higher, which can increase to about 68.97. This is resulted from the resnet50's deeper structure, which provides a higher ability for feature extraction and therefore makes it perform better than resnet18 that has a shallower network.

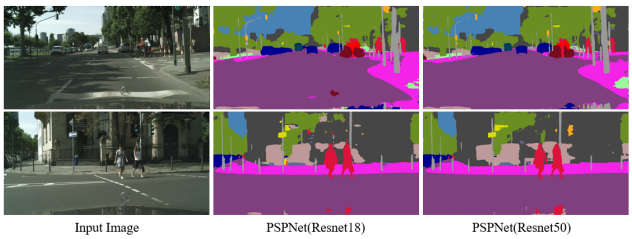

Figure 1. Visual comparison between PSPNet (Resnet18) and PSPNet (Resnet50).

2. Task 2

Implementation Details of ASPP module The working principle of ASPP module is using multiple dilated convolutions with different dilation rates to capture contextual cues in different regions. Implementation details are as follows:

Backbone	mIoU	mAcc	allAcc
Resnet18	62.89	71.05	93.36
Resnet50	68.97	76.67	94.52

Table 1. Validation results of resnet18 and resnet50 backbones for semantic segmentation in Cityscapes dataset. Unit: Percentage.

different levels of dilation (1, 2 , 3 , 6) are applied; for each level of feature extraction, a sequence of modules are used: Conv2d with dilation, BatchNorm2d, ReLU. In this part, the output size will not be changes, so interpolation process is not required here. Then, all these are concatenated together.

Implementation Details of OCR module As illustrated in [4], the working principle of OCR module includes using an attention-like method to grasp useful information for each element. The code is implemented with reference to the original open-source code of [4]. The implementation methods are as follows: building the spatial context head and spatial ocr head based on the original SpatialGather_module and SpatialOCR_module; for the forward function, the Conv or a sequence Conv dsn_head is used to calculate the soft object regions and dsn features; based on both features, the context features are calculated and merged with origin input features; the object contextual representations is calculated, which is followed by a Conv2d and an interpolation process.

Results & Analysis The validation results of applying ASPP module and OCR module are shown in Table 2. The results show that the performance of ASPP and OCR module is almost the same, while neither of them performs as well as the PPM module. Nonetheless, these structures are able to extract the contextual information by pyramid pooling and concatenation, so the performance is still better than those not considering the contextual information. The representative segmentation results are shown in Fig. 2. We can see that the OCR module have a more consistent object segmentation than PPM and ASPP, and the noise segmentation using OCR module is much smaller. That's because the OCR module uses the attention mechanism which can focus the segmentation in individual object.

Module	mIoU	mAcc	allAcc
PPM	62.89	71.05	93.36
ASPP	62.09	71.08	93.34
OCR	61.86	70.87	93.31

Table 2. Validation results of PPM, ASPP and OCR module for semantic segmentation in Cityscapes dataset. Unit: Percentage.

3. Task 3

Implementation Details According to [3], the MobileNet model is a lightweight deep neural network proposed by Google for embedded devices such as mobile phones. And the MobileNetV2 is an upgraded version. There are two specific structures used in this model. First is inverted residuals, which is different from the bottleneck design in ResNet. Second is linear bottlenecks, which is designed to avoid the destruction of features by ReLU. The downsample factor is set to 8, and the input features dimension is 320. To make the codes more clearer, a MobileNetV2 class is build. In this part, the convolutional stride and dilation are modified according to the downsampling factor value.

Results & Analysis The comparing results of mIoU-speed between resnet18, resnet50 and mobilenetv2 are shown in Table 3 and Fig. 4. The visual comparison between resnet18, resnet50 and mobilenetv2 is illustrated in Fig. 3. The device I use is NVIDIA 2080Ti. From these results, we can see that the efficiency of mobilenetv2 has improved considerably compared to resnet18 and resnet50. The good performance regarding to efficiency of mobilenetv2 mainly results from its different structure in the inverted residual block. ResNet uses standard convolutional methods to extract features, while MobileNetV2 uses depth-wise convolutional methods to extract features. And ResNet first reduces the dimension, then makes convolution, and finally increases the dimension, while MobileNetV2 chooses to increases the dimension at first, then makes convolution, and finally reduces the dimension. Intuitively, the microstructure of ResNet is hourglass-shaped, while MobileNetV2 is spindle-shaped, just the opposite. And that's why the paper [3] calls the structure inverted residual block. The MobileNetV2 structure is designed for the adaptation of its depth-wise convolutional work, which can reduce the time complexity and space complexity exponentially, and hope that feature extraction can be performed in high dimensions. That's the reason why the efficiency of MobileNetV2 is significantly higher than ResNet50 model.

Model	ResNet18	ResNet50	MobileNetV2
Batch latency (ms)	34	79	22
mIoU	62.89	78.97	59.25

Table 3. mIoU-speed results of ResNet18, ResNet50 and MobileNetV2 for semantic segmentation in Cityscapes dataset. Unit: Percentage.

4. Task 4

Implementation Details In this part, I choose to apply resnet101 model to improve the mIoU value and MobileNetV3 [2]. For the resnet101, the implementation de-

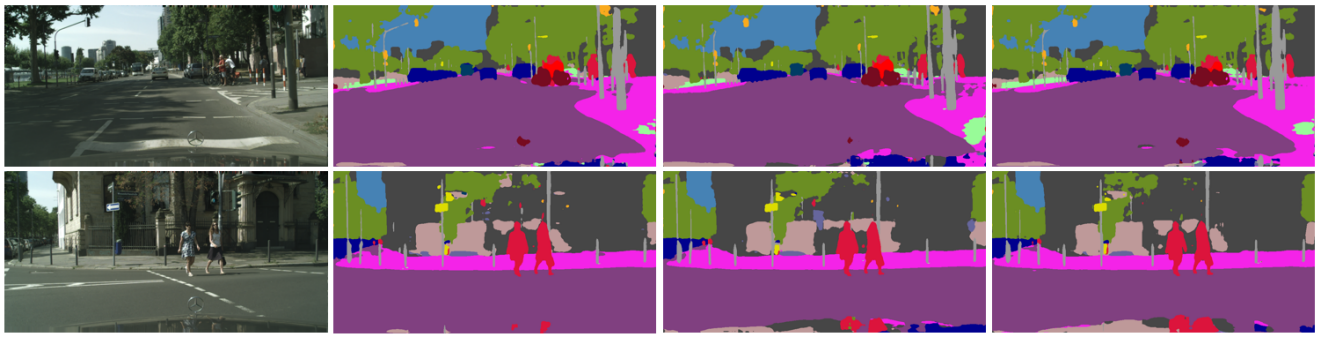

Figure 2. Visual comparison between PPM, ASPP and OCR.

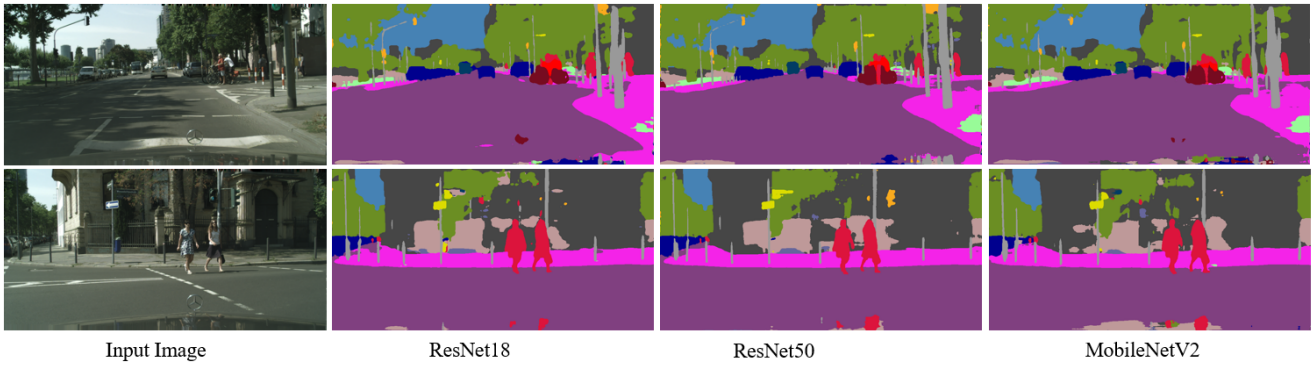

Figure 3. Visual comparison between ResNet18, ResNet50 and MobileNetV2.

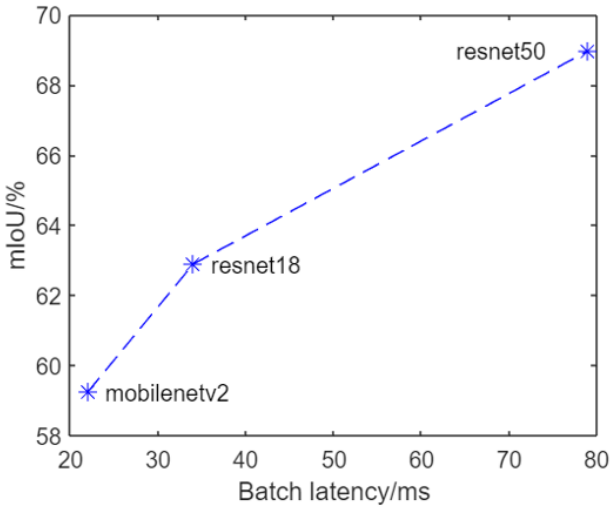

Figure 4. mIoU-speed curve of ResNet18, ResNet50 and MobileNetV2 for semantic segmentation in Cityscapes dataset.

tails are similar to resnet50. For MobileNetV3, the implementation method is similar to MobileNetV3, while some details have been modified according to the new structure.

Results & Analysis The validation results are shown in Table 4. From the results, we can see that using ResNet101 model as the backbone can improve the accuracy significantly. However, the results of MobileNetV3 are different than expected. In theory, the efficiency of MobileNetV3 should be higher than that MobileNetV2. But the results show that the efficiency performance as well as accuracy of MobileNetV3 is worse than MobileNetV2. We have tried lots of experiments, but output similar results. Some details may be incorrect when using MobileNetV3 as backbone. This should be further improved.

Backbone	mIoU	mIoU-speed
MobileNetV2	59.25	25.17
MobileNetV3	56.82	21.42
ResNet101	71.07	5.45

Table 4. Performance comparison between MobileNetV2, ResNet101 and MobileNetV3.

5. Conclusion

In this project, we implement the semantic segmentation considering the contextual information which consists of PPM, ASPP and OCR modules. Several backbones are

used in different parts, including ResNet18, ResNet50, MobileNetV2. Finally, we tried using ResNet101 and MobileNetV3 as backbones aiming to improve the mIoU value and mIoU-speed. But only ResNet101 succeed. The performance of MobileNetV3 is not good, which shows that further debug is required.

References

[1] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 1

[2] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1314–1324, 2019. 2

[3] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 1, 2

[4] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In *European conference on computer vision*, pages 173–190. Springer, 2020. 1, 2

[5] 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. 1