Applying Semantics Segmentation to Road Scene for Autonomous Driving

Yang Lu(u6274652)
, Honggu Lin(u6135394)
, Zhe Zhang(u6128882)
, and Teng Ma(u6123792)

ENGN6528 Group 11

Abstract. Image semantic segmentation is a more and more popular research topic in computer vision. The growing applications of image semantic segmentation such as autonomous driving, indoor navigation, and virtual reality systems,etc have led to its constant improvement in accuracy and efficiency. The deep learning approach which is widely used in most field of computer vision and machine learning nowadays is also used in Image semantic segmentation and it produces a great result. We use Fully Convolution Network(FCN) model and SEG-net model to do a pixel level classification for image semantic segmentation in this project. In this paper, we first introduce semantic segmentation and its related fields. Then we describe two kinds of widely used methods in semantic segmentation. Next, We introduce three evaluation metrics, IoU, Accuracy, and FPS, that we used in this project to measure our result. Finally, quantitative results and the related results discussion are given for the two mentioned methods and the datasets in which we were evaluated. At last, we make conclusions about this project and point out a set of promising future works for this project.

Keywords: Semantic Segmentation, Deep Learning, Scene Labeling

1 Introduction

Image semantic segmentation is a kind of pixel-level classification which classify each pixel of the input image to a color which represents to a class [1]. With the rising demanding of more accuracy and more efficient semantic segmentation mechanisms in applications such as autonomous driving, indoor navigation, robot vision and understanding, etc, it has become a very important research topic in computer vision and machine learning nowadays [2]. Semantic segmentation first became a part of the computer vision community at 2007, similar to other areas in computer vision, there is no major breakthrough in this area until 2014 when fully convolution neural networks were first used by Long et. al. to perform end-to-end image segmentation[1]. There are lots of approaches to resolve these problems, we can roughly divide them into traditional approaches and

deep learning approaches. Traditional approaches for segmentation include unsupervised methods, Decision Forests and SVMs [3]. Deep learning approaches in solving this problem mostly are based on Convolution Neural Network(CNN) [3].

Semantic segmentation is not an isolated field, to understand it fully, we need to know how this field evolves from coarse to fine inference. Figure 1 shows the evolution progress of object recognition or scene understanding from image classification to instance segmentation. The origin task is image classification, which means making a prediction on the classification of the objects in the input image. If there is only one kind of object in the input image, then just output one predicted label. If there are several kinds of objects in the input image, then output a ranked list of predicted labels about the objects. The next step is to detect and localize the objects and their spatial location in a rectangle box to make the segmentation more accurate and specific. After that, semantic segmentation becomes the natural step to achieve a higher fine-grained inference. Different from object location, semantic segmentation makes the segmentation prediction on pixel level, which means it makes dense prediction inferring labels for every pixel. Although semantic segmentation is very fine-grained comparing to the first two tasks, further improvements still can be made, for example, instance segmentation. Instance segmentation does not assign pixel in the same class with the same label, instead, it only assign pixel in the same object with the same label. So, given two cube objects in the image, these two objects are not the same object even though they are both belong to cube class, so these two objects will be labeled with different labels. In this paper, we only focus on semantic segmentation task.

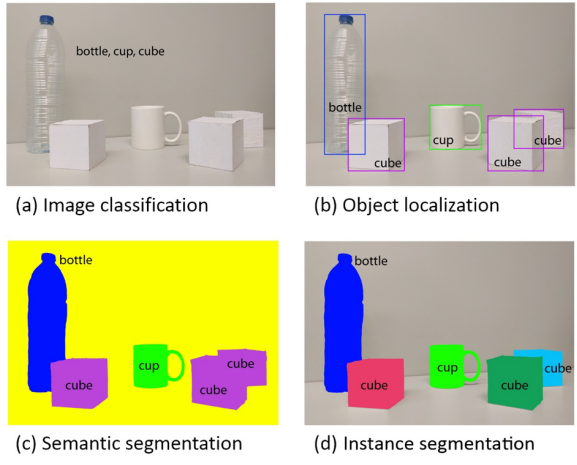

Fig. 1. Evolution of object recognition or scene understanding from coarse-grained to fine-grained inference: classification, detection or localization, semantic segmentation, and instance segmentation[2]

2 Related Works

Since 2012, Convolutional Neural Networks (CNN) has tremendous achievements and extensive applications in semantic segmentation. There are many models based on CNN architecture. For further understanding the common network architecture, we did some research on the current important network, such as AlexNet and VGG[4].

AlexNet was proposed by the SuperVision group and won the championship of ILSVRC-2012 by achieving the test accuracy of 84.6%, more than 10.8% ahead of the traditional architecture in the same challenge [2]. The AlexNets architecture was quite simple and easy to achieve [5].

VGGnet was proposed by Visual Geometry Group(VGG) from Oxford University, and it is a deep convolutional neural network. It became well-known because VGG-16 performed well in ILSVRC-2013 and achieved the test accuracy of 92.7%. VGGnet explores the relationship between networks depth and performance, and it successfully constructs the deep convolutional network in 16-19 layers[6].

However there are still some disadvantages in CNNs, in response to these issues, Jonathan Long et al. from UC Berkeley proposed Fully Convolutional Networks (FCN) for image segmentation[7]. The network attempts to recover the category of each pixel from the abstract features. That is, the classification from the image level further extends to the pixel level classification.

SEG-Net is based on FCN and revised the VGGnet, and could achieve 88.1% accuracy in semantic segmentation dataset. Its encoder network is topologically identical to the convolutional layers VGG-16 and remove the fully-connected layers of VGG-16, so that the SegNet encoder network significantly smaller and easier to train[8].

In this paper, we used FCN and SEG-Net to conquer the disadvantages caused by CNNs and improve the performance. We also used AlexNet and VGG as the benchmark and contrast to show the performance of our network in experiments. They are important components of our experiments.

3 Methods

In this section, we introduce two methods we used for semantic segmentation in the project.

3.1 Fully Convolution Network

Generally, convolution neuron network is (CNN) widely used in image segmentation. The advantage of CNN lies in multi-layered structure that automatically learns features [5]. It is basically composed of convolution layer, pooling layer and fully connected layer. Actually, different depth convolution layers allow CNN to perceive features in a large range of perspectives. For example, shallow layers can sense details with complicated information, such as position and direction.

In contrast, the deep layers can distinguish more abstract features in a large perceptual domain. Based on this, the method to classify every pixel according to convolution with surrounding pixels. It really works to segment the images approximately. However, traditional CNN can lead to the high cost of storage since every pixel needs a storage for the area size times of the pixels block. Actually, the efficiency is relatively low because the surrounding pixels are calculated many times for different pixels. Furthermore, the performance can be suppressed by the size of the kernel.

To solve these problems, fully convolutional network (FCN) strategy is raised in UC Berkeley to segment images more explicitly [7]. As the precedent for image segmentation in semantic scope, FCN achieves pixel level end-to-end classification. There are two main issues to be solved in FCN, one is to settle strict image size of the input, and another is developing the efficiency for storage by avoiding duplicate calculation. In detail, the first five layers in CNN are convolution layers. After that, there are two one dimension vectors whose length is 4096. Moreover, a similar vector sized 1000 accounts for probability. The only difference between CNN and FCN is that FCN replaces all one dimension vectors for fully-connected layer with convolution layer. For example, a kernel, whose channel, width and height are 4096, 1 and 1 respectively, substitutes for the one dimension layer whose length is 4096. Thus, all layers are convolution layers as shown in figure 2, and this is why FCN is named fully convolution network.

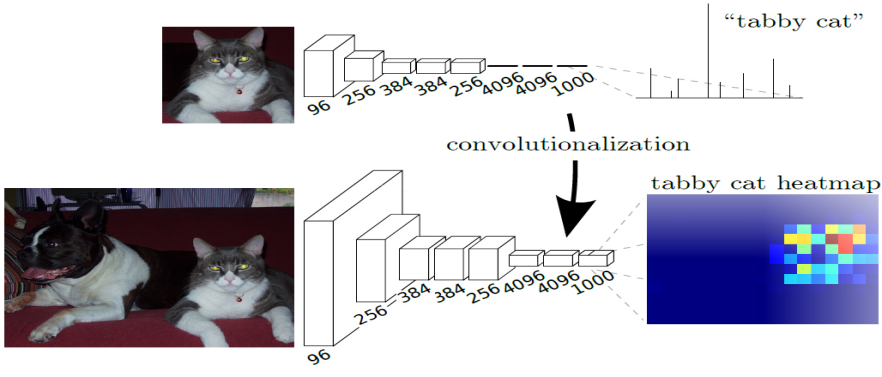

Fig. 2. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition(Long, Shelhamer and Darrell, 2015).

In convolution layers, there are procedures including pooling and up-sampling. After n times convolution and pooling, the image gets smaller and small whose size actually becomes $\frac{1}{2^n}$ times. For example, after the max 5 times convolution and pooling operations, the resolution of the image turns into 1/32 times from original image. For the output of the image at the end layer, the up-sampling

should be 32 times to get the same size comparing to the original image. This process is implemented by transposed convolution, in other words, backward convolution and deconvolution. To get more explicit details, the up-sampling can be done on the output of third layer and the fourth layer, for 16 times and 8 times respectively. In conclusion, FCN experiences a resizing process from getting small to retrieving to the original size to predict every pixel classification explicitly. In up-sampling, increasing steps perform better than one step with coordinate assistance from traits in different steps. However, FCN has a low distinguishing ability from different depth layer traits. This may lead to the waste of high dimension traits. Nevertheless, it provides a good thought on image semantic segmentation.

3.2 SEG-net

Base on FCN, there is a developed encoder-decoder convolution network, named SEG-net, put forward by Cambridge in 2016 [8]. Comparing to FCN, there is a significant improvement in pooling layer. In FCN, much information is abandoned in pooling layer. And in up-sampling, there is no operation to recover the information. However, in Segnet, sparse up-sampled maps are transferred to dense feature maps. Similarly, Segnet, based on first thirteen layers in VGG-16 network, replaces the fully connected layers with convolution layers as FCN [9]. Furthermore, it develops the detail segmentation and improves the memory utilization.

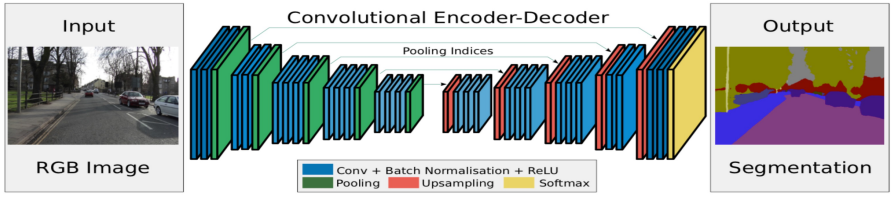

Fig. 3. Segnet: A deep convolutional encoder-decoder architecture for image segmentation.(Badrinarayanan, Kendall & Cipolla, 2017)

As illustrated in figure 3, the input is an image, and output is segmentation result. The output classifies different categories with distinct colors. There are two symmetrical parts, one is encoder part, the other is decoder part. Every encoder layer has a corresponding decoder one. In detail, encoder part includes a number of blue layers, which is composed with convolution layer, batch normalization layer and ReLU layer, and a max pooling layer with a non-overlapping 2*2 window, whose stride is 2. However, the max pooling and subsampling can lead to the loss of boundary details. Thus, max pooling indices should be saved, and it enhances memory utilization comparing to FCN. Moreover, transfer of

maximum pooling index towards decoder layer can develop segmentation resolution. For decoder part, it is made up with up-sampling layer and the blue layers which is similar to FCN. In the end, the result is delivered to a soft-max classifier to calculate the max probability for every pixel.

The SEG-net convolution is same as in FCN, using the 'same' convolution to keep the same size. One thing should be concerned is that process of up-sampling in SEG-net, concentrates on mapping the values of feature map to the new feature map according to the pooling indices and coordinates saved in max pooling. And other positions are all replaced with zeros. Instead, FCN does the adding operation with results after transpose convolution and corresponding feature map after encoding. Before the activation function, batch normalization (BN) is a way to accelerate the learning rate, particularly in avoiding gradient disappearance and explosion with ReLU. BN layer only preserves the average and variance without changing in forward propagation. It helps with ReLU to calculate the learning rate. As for output, the softmax layer makes the end-to-end sense. It labels the pixels with the highest probability classification to perform the pixel-wise.

4 Evaluation Metrics

In this section, we introduce three evaluation metrics used in this project.

4.1 IoU

The purpose of a suitable evaluation matrix for a computer vision problem is to identify the (dis)similarity between the possible solution and the ground-truth presented in a perceptual way. For solving the problem of semantic segmentation, we decide to choose the popular matrix standard Jaccard Index or known as Intersection-Over-Union (IoU) measures[10], which computed the discrepancy between the ground-truth area and the predicted area, and calculate the average value among all categories [11].

The IoU score is a common method to evaluate the performance in the semantic segmentation problems. If the images dataset is given, the IoU score will reflect the similarity between the ground-truth area and the predicted area for a specific object presented in such images [12]. Figure 4 is an example to calculate the IoU score.

By examining the equation, we can find the IoU is simply a ratio, which is the correct prediction region divided by the union area including the predicted and ground-truth region. Thus, the IoU score could be defined by the following equation:

$$IoU = TP/(FP + TP + FN) \tag{1}$$

where TP, FP and FN correspondingly stand for the true positive, false positive and false negative [13].

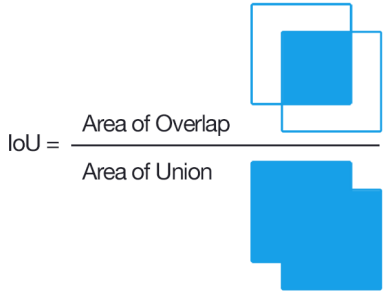

Fig. 4. Computing the Intersection of Union.[12].

From equation (1), We can find that the IoU score is a measurement of pixels in the practical semantic segmentation problems. But FCNs outputs are probability values of each pixel for being a segment in the target object. Therefore, we are not able to get the accurate score from the FCNs outputs. We tried to get the approximate score by computing the probability values.

According to Rahman & Wang (2016)[12], we can get the IoU count by the following equations in the practical situations:

$$IoU = (I(X))/(U(X)) \tag{2}$$

where, I(X) and U(X) can be approximated as follows:

$$I(X) = \sum_{v \in \mathbf{V}} X_v \times Y_v \tag{3}$$

$$U(X) = \sum_{v \in \mathbf{V}} (X_v + Y_v - X_v \times Y_v) \tag{4}$$

Where, V ={1, 2, . . . , N} is the all pixels of the images in the training set. X is the output of the FCN. And X represents the probability values over the set V. And Y∈{0, 1} is the ground-truth assignment for the set V, where 1 represents object pixel and 0 represents background pixels [12, 14]

In the experiments, we will report IoUclass as mean performance score, and the invalid labelled pixels will not influence the score.

4.2 Accuracy

Accuracy indicates the percentage of correctly identified pixels for each class. We use accuracy metric to evaluate the network performance as another standard. For each case, we calculate the accuracy as the ratio of correctly classified pixels to the total number of pixels in that class, according to the ground truth.

$$Accuracy = TP/(TP + FN) \tag{5}$$

where, TP is the number of true positives pixel and FN is the number of false negatives pixel. Though the class accuracy is a simple metric analogous to global accuracy, it can be misleading.

Thus, we used global accuracy as the 'Accuracy Score' to evaluate the network performance. Global accuracy is the ratio of correctly classified pixels, regardless of class, to the total number of pixels. And it can provide more comprehensive results.

4.3 Inference Runtime Performance

Autonomous driving system needs to have immediate reactions to the new events so that it can ensure the safety of passengers and other associated personnel. In order to satisfy this, it is often acceptable that the objects borders are not perfectly recognized. Real-time image segmentation is another strong requirement in autonomous driving. Therefore, it is significant that the system with deployed neural network could meet the requirements of execution speed[15].

There have been many methods for reducing the execution time for neural network. One of these is improving the network architecture, such as SqueezeNet adopted a more efficient architecture to reproduce the image classification accuracy of AlexNet [16, 17], and ENet used the same method and demonstrated that semantic segmentation could be used in real-time mobile and embedded equipment[18]. The others increase the efficiency of the current network. The method is deriving simpler network from the large entire network by pruning or executing the network on the specific devices. These methods can be used to speed up execution in autonomous driving system [19].

In this task, we are asked to run on a regular laptop at the speed of processing one image in less than two seconds, or faster than 0.5 FPS. Our implementation is benchmarked and tested on the datasets of KITTI and Cityscapes for semantic segmentation in 'Scene Labeling for Autonomous Driving'.

5 Experiment

5.1 Dataset

The dataset we used in this project is KITTI road scene semantic segmentation dataset. This datasets contains 445 origin RGB images and their corresponding ground true semantic segmentations. We hand picked 245 images as training dataset, and 100 for validation, and the rest 100 for test purpose.

In this pixel level semantic segmentation task, each pixel in the scene can be classified into 11 semantic categories: building, vegetation, sky, road, fence, pole, sidewalk, sign, car, pedestrian, bicyclist.

5.2 FCN Model

We present and analyze the evaluation results using FCN model in this section. The accuracy of the results is evaluated using IoU metric and accuracy metric,

the evaluation results are presented in Table 1. We also measure the speed of the model using Frame Per Second(FPS) evaluation metric, the result is evaluated on NVIDIA K40 with 18.3FPS, which much more quicker than 0.5FPS as require.

FCN	IoU	Accuracy
train	0.72	0.79
validation	0.61	0.78
test	0.62	0.78

Table 1. The evaluation results of FCN model in IoU and Accuracy evaluation metric

The training loss and the training accuracy and IoU over each epoch is shown in Figure 5. As we can see from Figure 5, the training loss descend over the epoch while the training accuracy and IoU increase with the similar trend over epoch.

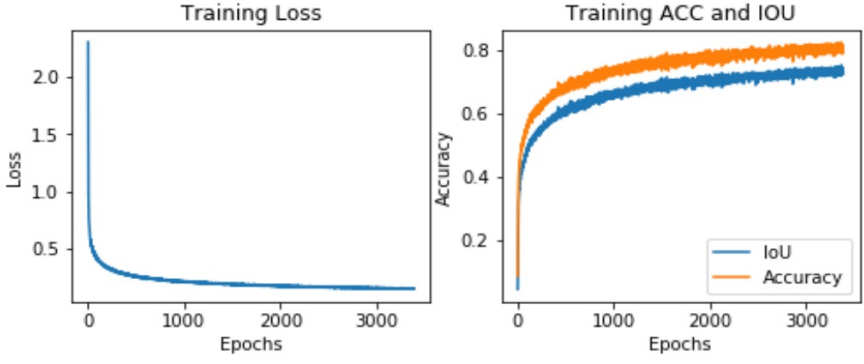

Fig. 5. Training Loss and Training Accuracy and IoU of our FCN Model

The comparison between the original image, the ground true image and the prediction image using FCN model is shown in Figure 6. The ground true semantic segmentation is almost the same as the prediction semantic segmentation.

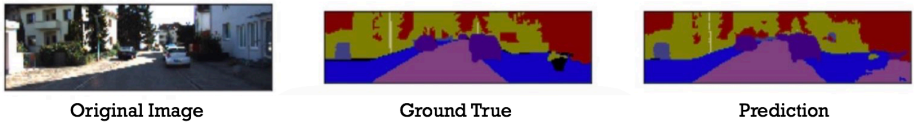

Fig. 6. Comparison between the original Image, Ground True Image and Prediction Image in FCN Model

5.3 SEG-net Model

The evaluation results using SEG-net model is presented in this section. The evaluation result is shown in Table 2. The process speed of our SEG-net model is 15.26FPS evaluated in NVIDIA K40.

SEGNET	IoU	Accuracy
train	0.74	0.85
validation	0.44	0.58
test	0.44	0.70

Table 2. The Evaluation Results of SEG-net in IoU and Accuracy Evaluation Metric

The training loss and training IoU and Accuracy over each epoch is shown in Figure 7. The loss is decrease over each epoch while the accuracy and IoU is increase.

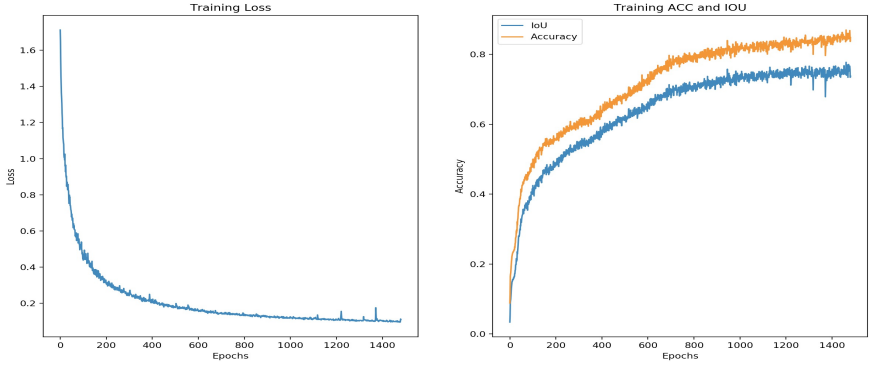

Fig. 7. Comparison between the original Image, Ground True Image and Prediction Image in SEG-net Model

Fig. 8. Comparison between the original Image, Ground True Image and Prediction Image in SEGNET Model

A sample of the semantic segmentation prediction result of SEG-net Model is shown in Figure 8.

6 Conclusions and Future Work

6.1 Conclusions

Comparing to traditional convolution neuron network, fully convolution network performs well on pixel-wise and end-to-end image semantic segmentation. FCN develops the efficiency dramatically by avoiding the duplication of storage and convolution calculation. It means that FCN is competent to autonomous driving by road scene segmentation. However, it is only regarded as the basis of the image semantic segmentation field. Because there are many fatal flaw in FCN. Apparently, the training process of FCN is complicated. For example, FCN-8s requires at least three times of training. Besides, the result shows that the segmentation is not really delicate. From the comparison of the result, it can be reduced that the rough segmentation comes from the decoding process. The up-sampling process in recovery demonstrates sparse label maps. Actually, it is a simple process of transposed convolution, in other word, just a deconvolution. Thus, the insensitivity of the boundary details may lead to a terrible experiments. Moreover, the classification process for pixels ignores the spatial relationship. It can result in the failure of spatial consistent. Furthermore, a potential risk comes from experiments on different dataset, particularly when the sizes between the training set and training set are significant different, the segmentation performance reduces dramatically. It results from the ignoring for high dimension traits instead of reducing the weight for shallow layers. Nevertheless, FCN illustrates a new efficient way for image semantic segmentation, and it provides enough basis for further research. In contrast, SEG-net shows a developed design with high efficiency. It only store the mapping index for the max pooling. It enhances the performance of the decoding process. It demonstrates a efficient network for road scene segmentation understanding both on memory and computational time.

6.2 Future Work

Even though the SEG-net performs a high score in road scene semantic segmentation, end-to-end learning of deep segmentation architecture is still a challenge. Actually, the more challenging part concentrates on the small object distinguishability. However, the tiny detail mistake can bring misleading to some extent. If it is possible, the weakly-supervised training can be used to localize the difference for local traits, and it can perform to a high level accuracy.

References

1. Meet Shah: Semantic Segmentation using Fully Convolutional Networks over the years (2018)
2. Alberto, G.G., Sergio, O.E., Sergiu, O., Victor, V.M., Jose, G.R.: A Review on Deep Learning Techniques Applied to Semantic Segmentation. (2017)
3. Thoma, M.: A survey of semantic segmentation. (2016)
4. Wu, Y., Zhang, Y., Zhang, C., He, Z., Zhang, Y.: Semantic segmentation of mechanical parts based on fully convolutional network. Modelling, Identification and Control (ICMIC), 2017 9th International Conference (2017) 612–617
5. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. (2012) 1097–1105
6. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. (2014)
7. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2015) 3431–3440
8. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE transactions on pattern analysis and machine intelligence **39**(12) (2017) 2481–2495
9. Badrinarayanan, V., Handa, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. arXiv preprint arXiv:1505.07293 (2015)
10. Barth, R., IJsselmuiden, J., Hemming, J., Henten, E.: Data synthesis methods for semantic segmentation in agriculture: A capsicum annuum dataset. Computers and Electronics in Agriculture **144**(284) (2018)
11. Ahmed, F., Tarlow, D., Batra, D.: Optimizing expected intersection-over-union with candidate-constrained crfs. (2015)
12. Rahman, M.A., Wang, Y.: Optimizing intersection-over-union in deep neural networks for image segmentation. International Symposium on Visual Computing (2016)
13. Everingham, M., Eslami, S.A., Van Gool, L., Williams, C., Winn, J., Zisserman, A.: The pascal visual object classes challenge: A retrospective. International journal of computer vision **111**(1) (2015)
14. Hariharan, B., Arbelez, P., Girshick, R., Malik, J.: Simultaneous detection and segmentation. European Conference on Computer Vision (2014)
15. Treml, M., Arjona-Medina, J., Unterthiner, T., Durgesh, R., Friedmann, F., Schuberth, P., Nessler, B.: Speeding up semantic segmentation for autonomous driving. MLITS, NIPS Workshop (2016)
16. Iandola, F., Han, S., Moskewicz, M., Ashraf, K., Dally, W., Keutzer, K.: Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. (2016)
17. Krizhevsky, A., Sutskever, I., Hinton, G.: Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems (2012)
18. Paszke, A., Chaurasia, A., Kim, S., Culurciello, E.: Enet: A deep neural network architecture for real-time semantic segmentation. (2016)
19. Han, S., Liu, X., Mao, H., Pu, J., Pedram, A., Horowitz, M., Dally, W.: Eie: efficient inference engine on compressed deep neural network. Computer Architecture (ISCA), 2016 ACM/IEEE 43rd Annual International Symposium on (2016)

Peer Review

At the very first beginning, I would like to express my sincere gratitude to Prof Li and our tutors for their great contribution and time spend on this course. All members in our group has spent lots of time in this project and we all learn a lot from it.

We have equal contribution to this project. The detail contribution to Presentation, Code and Report are list below:

	Part	Contribution
Presentation	PPT	Equal
	Presentation	Equal
Report	Abstract & Introduction	Honggu Lin
	Related Work	Teng Ma
	Methods	Zhe Zhang
	Evaluation Metrics	Teng Ma
	Experiment	Honggu Lin & Yang Lu
	Conclusion & Future Work	Zhe Zhang & Yang Lu
Code	SEG-net	Honggu Lin & Yang Lu
	FCN	Zhe Zhang & Teng Ma
	Data Pre-processing	Yang Lu