Machine Learning Final Project: MediaTek Low-power Semantic Segmentation

Group 16: 廖家緯, 王綰晴, 林彥廷, 陳品樺
National Yang Ming Chiao Tung University

June 9, 2022

Abstract

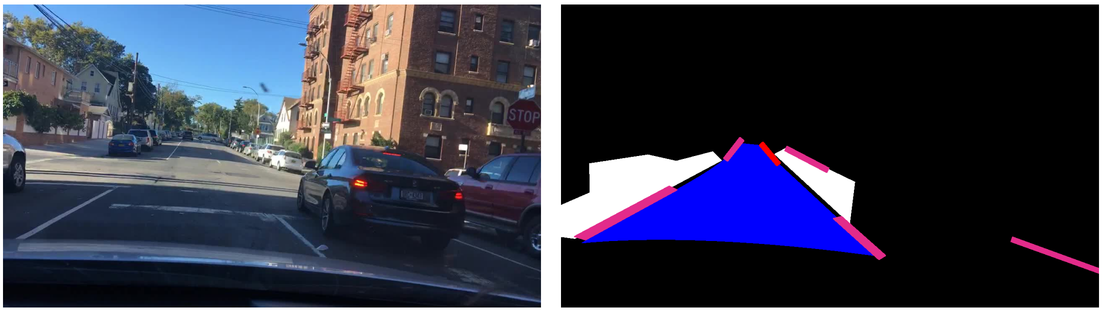
In this project, we design a lightweight deep learning-based semantic segmentation model suitable for constrained embedded system design to deal with traffic scenes in Asian countries like Taiwan. We focus on the improvement of segmentation accuracy, power consumption, real-time performance optimization and the deployment on MediaTek's Dimensity Series platform. The evaluation metrics is mIoU, which is widely use to multi-class semantic segmentation task. After our experiment, we get the test public score of 0.562353. Our code is available at `https://github.com/Jia-Wei-Liao/MediaTek_LowPower_Semantic_Segmentation`.

1 Introduction

In computer vision, the task of Segmentation is to segment a picture into masks that can be tagged and classified, while Semantic Segmentation tries to assign a classification label to each pixel in the image. At first, it was used to segment jointed objects, such as people or animals, and then the technique was gradually used to solve more delicate problems, such as pose estimation. In real applications, this technique is being used to improve resolution in addition to autonomous driving systems. After obtaining the segmented area, the information from the segmented area is used to make a detailed repair, and with style migration and coloring techniques, a high-resolution version is predicted on the low-resolution sample. With this technique, we can obtain better-quality photos without upgrading the camera hardware. If we apply this technique to cell phones, we can get better photos with the only software upgrade.

In this task, the dataset contains 31,725 training images with 720 × 1280 resolution and masks with six labels. Our goal is to segment the image and classify pixel-wisely.

The UNet is a famous network for semantic segmentation. It is an encoder-decoder-based model like U-shape, which is not only accurate but also lightweight. We divide the experiment into the following parts. First, we use the training data set provided by MediaTek and design our own network to train. Second, we do the post training quantization to get smaller model size and faster inference time. We can also get lower power consumption by doing this. Last, we convert the .pb file to .tflite file so that we can execute on the platform.



(a) image

(b) mask

Figure 1: training data

2 Related Work

Semantic segmentation is an important topic in computer vision. In 2014, Fully Convolutional Networks (FCN) [5] has been proposed. FCN substitute the fully conneted layer (last layer) by transposed convolution layer in the classification model. Therefore, the model can accept variant size of input image and can be trained end-to-end. FCN has establish the foundation of semantic segmentation models. In 2015, UNet [6] has been proposed. UNet use a symmetric encoder-decoder structure, which means the decoder part is more robust than FCN. Moreover, skip connection between encoder and decoder that can keeps the finer feature (from the encoder), mitigate the problem of information loss that FCN faced.

UNet has been widely used in medical image segmentation. There are many variant of UNet in this recent years. For example, inspired by ResNet and DenseNet, ResUNet [8] and DenseUNet [2] has been proposeed in 2018. nnUNet (no-new-Net) [4] proposed in 2019, which concentrate on data pre-processing, training scheme and inference-scheme and get a great improvement. This proves the importance of understanding of the data. TransUNet [1], which proposed in 2021, try to enhance the encoder part by combining CNN and transformer.

3 Method

3.1 Data Pre-processing and Data Augmentation

To avoid over-fitting and enhance the robustness of a mode, we use some data augmentation technique as the follow.

- **Resize the image:** Since UNet model's encoder had repeated downsampling 5 times, the image size at least is $2^5 = 32$ or the feature will be vanish. In order to avoid unnecessary photo size issues and distortion, we adjusted the image size from (720, 1280) to (192, 320) so it can be divisible by 32 and maintained the similar photo ratio.
- **Random horizontal flip:** With a probability of 0.5, we flip the image left and right.
- **Random adjust brightness:** With a probability of 0.1, we set the bright value range from 0.9 to 1.1, and modified the grayscale. Then we clipped the value into [0, 1] if out of range.
- **Random add noise:** Add the gaussian noise by $I(i,j) + \sigma \cdot s$ with a probability of 0.1, where $s \sim \mathcal{N}(0,1)$ and I is the grayscale.
- **Normalize image:** The images are represented as tensors with pixel values ranging from 0 to 255. We divide by 255 which rescale the range to [0, 1].

3.2 Model Architecture

In this task, we adopt a UNet based model. We will introduce the model architecture in this section. First, we may want to generalize the architecture of UNet to a abstract topological structure. As shown in the Figure 2, we may divide UNet architecture into three parts: Encoder, Bridge and Decoder and we show the architecture of UNet in Figure 3. In the next subsections, we will introduce them one by one.

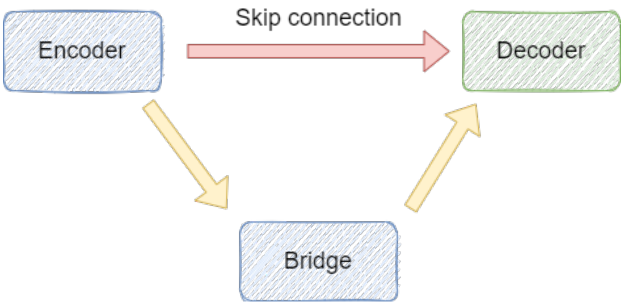


Figure 2: Abstract topological structure of UNet

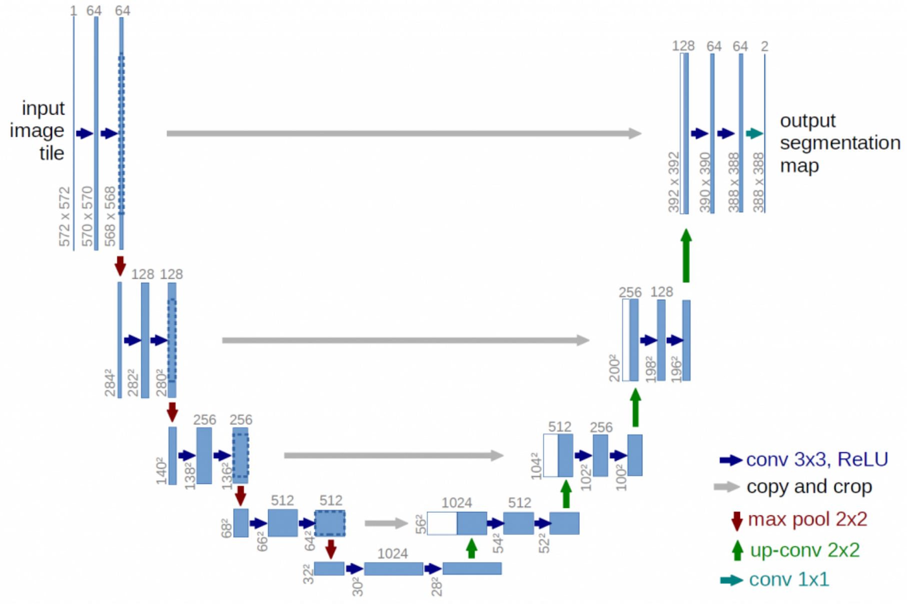


Figure 3: UNet architecture

3.2.1 Encoder and Bridge

The goal of the encoder is to extract useful features. However, it is not necessary to follow the encoder in the original UNet. Any CNN based networks (encoder) with downsample steps can be used as an encoder. In 2015, Kaiming He proposed the ResNet, which use the residual block to get a better performance. So we attempt to add this mechanism to our network. In Figure 4, we display illustration of the residual block.

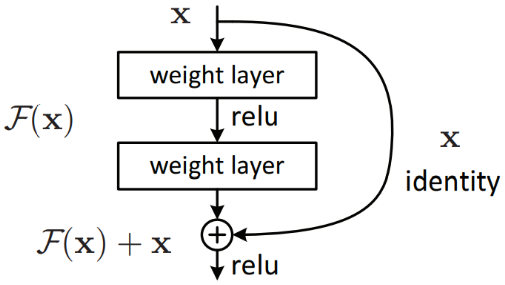

x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 4: Residual block

3.2.2 Decoder

For the decoder part, we follow the original UNet architecture. The only difference is that the transpose convolution is replaced by nearest interpolation.

Combining the Encoder, Bridge and Encoder that we mentioned above, the model architecture looks like the following Figure 5.

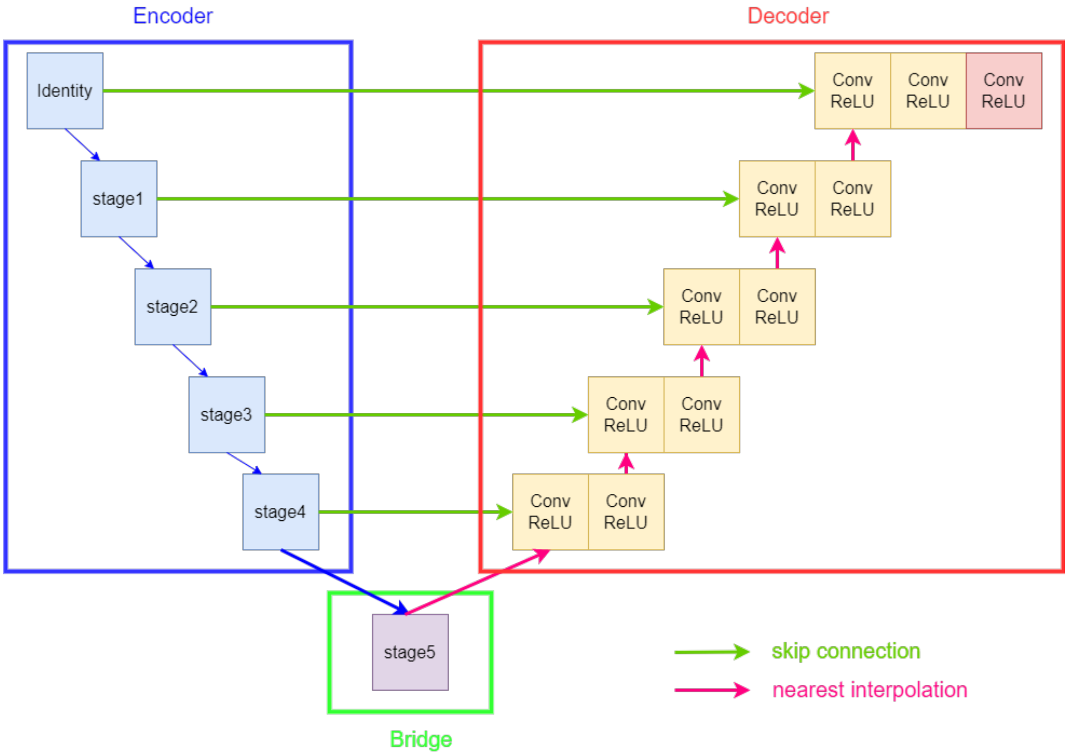


Figure 5: Model architecture

3.3 Loss Function

Semantic segmentation is pixel-wise classification, so we can use cross entropy as the loss function. Let $Y, \hat{Y}$ be the ground truth with one-hot label and prediction. The cross entropy is defined as

$$\mathcal{L}_{CE}(\hat{Y}, Y) = -\sum_{k=1}^{6} \sum_{i,j} Y_{i,j,k} \log \hat{Y}_{i,j,k}$$

Moreover, to avoid over-fitting, we add the L_2 regularization term, limiting the parameter freedom.

$$\mathcal{L}_{reg}(\Theta) = \sum_{\theta \in \Theta} \|\theta\|_2$$

Therefore, the total loss is

$$\mathcal{L}_{total} = \mathcal{L}_{CE} + \lambda \mathcal{L}_{reg}$$

where $\lambda > 0$ is the hyper-parameter.

3.4 Optimization

Adam is one of the most effective optimization algorithms for training. It is a stochastic optimization method that combine advantage of RMSprop and AdaGrad. We give the algorithm as the following:

Algorithm Adam

```
Input: $\gamma, \beta_1, \beta_2, \theta_0, f(\theta), \varepsilon$
Initial: $m_0 \leftarrow 0, v_0 \leftarrow 0,$
for $t = 1$ to ... do
    $g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$
    $m_t \leftarrow \beta_1 m_{t-1} + (1 - \beta_1) g_t$
    $v_t \leftarrow \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$
    $\hat{m}_t \leftarrow \frac{m_t}{1 - \beta_1^t}$
    $\hat{v}_t \leftarrow \frac{v_t}{1 - \beta_2^t}$
    $\theta_t \leftarrow \theta_{t-1} - \gamma \frac{\hat{m}_t}{\sqrt{\hat{v}_t + \varepsilon}}$
end for
return $\theta_t$
```

3.5 Learning Rate Scheduler

To further improve results and make the model converge to the global minimum, we adjust the learning rate by exponent decay. It represent by the following:

$$\text{Learning Rate} = (\text{Initial Learning Rate}) \times (\text{Decay Factor})^{\frac{\text{Epoch}}{\text{Decay Period}}}$$

We set up the Initial Learning Rate $= 5e - 4$, Decay Factor $= 0.8$ and Decay Period $= 3$.

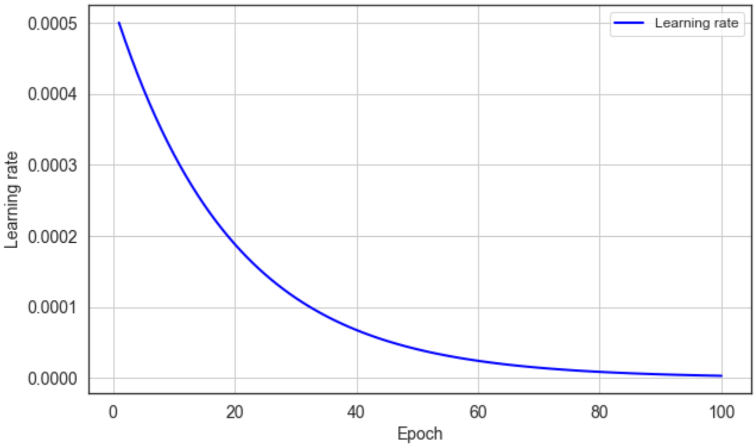


Figure 6: Exponential decay

3.6 Evaluation Metrics

In this task, the measurement indicator is evaluated on the mean intersction over union (mIoU). We introduce the IoU at the first. It can simplify the formulation by confusion matrix as the following Table. The IoU is defined as

$$\text{IoU} = \frac{\text{Intersection}}{\text{Union}} = \frac{TP}{TP + FP + FN}$$

Predicted Actual	Predicted Positive	Predicted Negative
Actual Positive	true positive (TP)	false negative (FN)
Actual Negative	false positive (FP)	true negative (TN)

Table 5. Confusion matrix

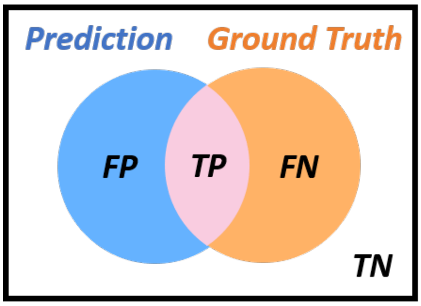


Figure 7: Venn diagram of confusion matrix

To define the IoU on the multiple classes, we can compute the IoU for each classes and then take average of them. We formulate as the follow:

$$\mathrm{mIoU} = \frac{1}{n}\sum_{i=1}^{n} \frac{TP_i}{TP_i + FP_i + FN_i}$$

3.7 Post Processing

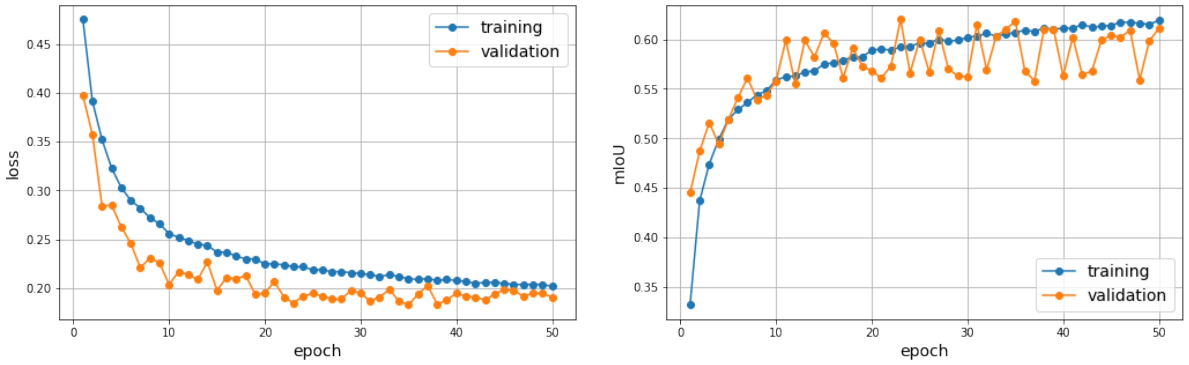
After the model training is completed, you can get a .pb file. In general, there are usually certain requirements for the speed of the model. Therefore, we can optimize our network through MLKits, which can be used to select some tools to use. For example, you can use Neural Architecture Search (NAS) to explore design space or search space, you can also use Quantization techniques to achieve lower bandwidth and so on.

Quantization can be divided into two categories, quantization-aware training and post-training quantization. For quantization-aware training, training is required, there is an input image, the weights need to be updated continuously, there is a loss function, and the learning rate needs to be adjusted. The other is post-training quantization, no need to adjust weights, no loss function, no need to do learning rate decay, etc. However, although quantization-aware training is more troublesome, the accuracy will be maintained through continuous correction, while post-training quantization is relatively simple, but there will be a loss of accuracy.

The most used method in quantization is linear quantization. By adding FakeQuantized models, floating point computations become integer computations so that our models can run faster on the platform.

4 Experiments

The segmentation accuracy of our final project is based on mIoU, which is the so-called Mean Intersection over Union. It is widely used in multi-class semantic segmentation task by calculating the overlap ratio between the actual value and the predicted value. It can be found from Figure 8 (a) that as the number of epochs increases, both the training loss and the validation loss show a downward trend, so it can be predicted that the model will improve in mIoU. So next we see Figure 8 (b), we can find that our model can have a steady upward trend on the training data set. The reason is very simple, because the model is trained with the training set, so the training results are very good. For the validation dataset, our model has a significant improvement in mIoU at the beginning. However, as the number of epochs increases, the performance improvement gradually slows down, and there is a slight oscillation phenomenon, but the overall performance improvement also gradually converges to the effect close to the training data set.



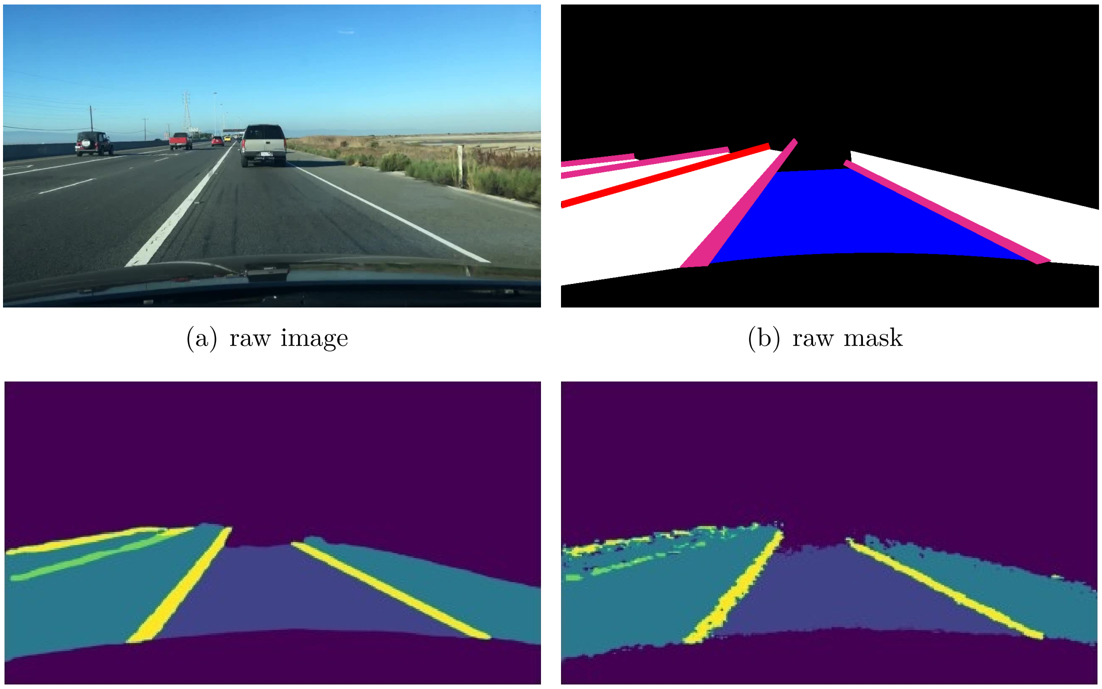
(a) loss

(b) mIoU

Figure 8: Optimization process: The loss of our training process starts from 0.48 and drops steadily to about 0.2 after 50 epochs. Although the validation loss fluctuated a bit, it also dropped from 0.4 at the beginning to about 0.18 after 50 epochs.

Model	Channel	Input size	LR	Down Sampling	Residual	BN	val mIoU
UNet	5	(256, 256)	5e-3	Average Pooling	×	✓	0.557
UNet	5	(192, 320)	1e-3	Average Pooling	×	✓	0.609
UNet	8	(192, 320)	1e-3	Average Pooling	×	✓	0.606
UNet	8	(192, 320)	1e-3	Convolution (stride=2)	×	✓	0.542
UNet	12	(192, 320)	1e-3	Average Pooling	×	×	0.562
UNet	12	(192, 320)	1e-3	Average Pooling	×	✓	0.585
UNet	12	(192, 320)	1e-3	Average Pooling	×	✓	0.601
UNet	12	(192, 320)	1e-3	Average Pooling	✓	✓	0.612

In Figure [9, 10, 11], we show the raw image, raw mask, prediction, and prediction with quantization model. We can observe the difference between picture (c) and picture (d). When we make lightweight improvements to the model, there will be some noise on the line of picture (d), but the outline of picture (d) is all Very close to either figure (a) or figure (b).



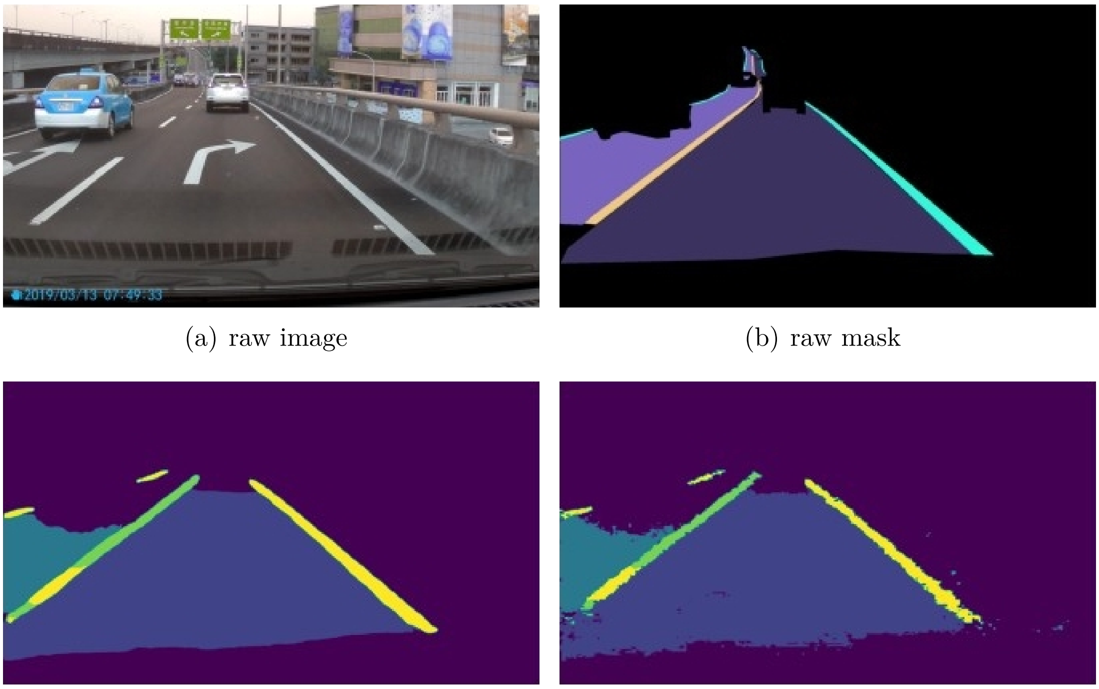
(a) raw image

(b) raw mask

(c) prediction w/o quantization

(d) prediction w/ quantization

Figure 9: training data



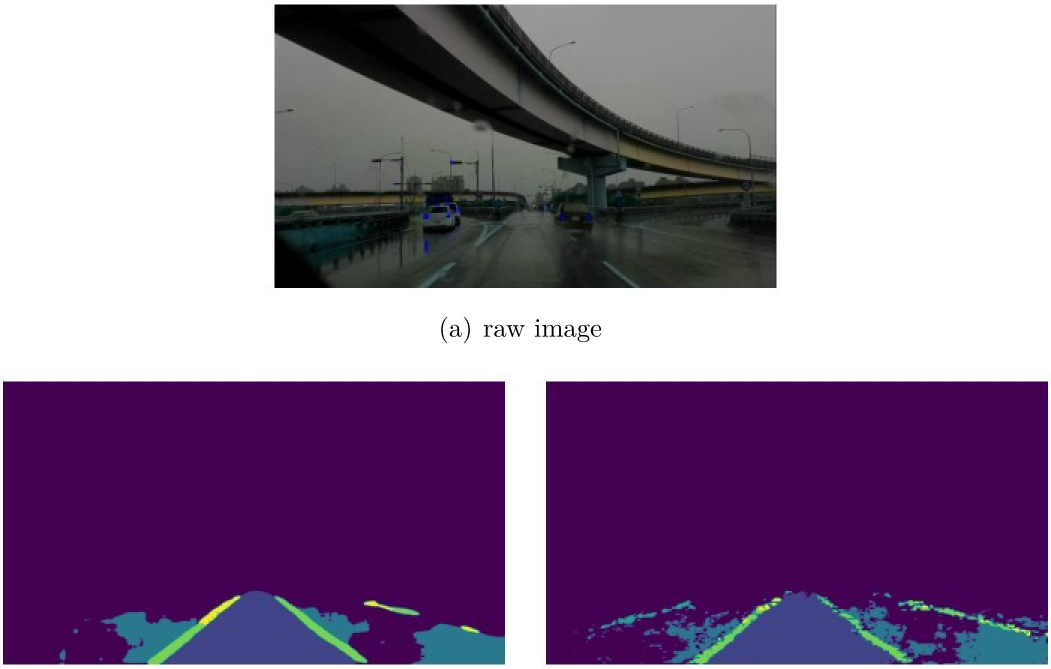
(a) raw image

(b) raw mask

(c) prediction w/o quantization

(d) prediction w/ quantization

Figure 10: validation data



(a) raw image

(b) prediction w/o quantization

(c) prediction w/ quantization

Figure 11: testing data

5 Conclusion

In this project, we implemented a UNet model for semantic segmentation of images. To aviod over-fitting, we use data augumentation and add the regularization term for loss function. It enhance the robustness of our model. Based on the available results, the optimal configuration is Resize shape: height 192 and width 320, Optimizer: Adam with learning rate scheduler and full training implementation. Due to the limit of power in MediaTek Dimensity Serie cellphone, we use the quantization technique after training to get the lightweight model. The network achieves an mIOU score of 0.5624 on the public dataset. Our model also gets the 15th and 14th ranking in terms of latency and power respectively compared to the other group structure. It is clear that our model can be lightweight and applied in practice while maintaining mIOU.

	Value	Rank
Latency	91012.3	15
Power	692.88	14
mIoU	0.5624	25

6 Contribution

309652008 廖家緯	coding (model design, training), and report writing
310510179 王綰晴	data analysis, and report writing
310511061 林彥廷	coding (quantization), and report writing
310511067 陳品樺	coding (quantization), and report writing

References

[1] Jieneng Chen, Yongyi Lu, Qihang Yu, Xiangde Luo, Ehsan Adeli, Yan Wang, Le Lu, Alan L. Yuille, and Yuyin Zhou, TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation *CVPR*, 2021.

[2] Steven Guan, Amir A. Khan, Siddhartha Sikdar, and Parag V. Chitnis, Fully Dense UNet for 2-D Sparse Photoacoustic Tomography Artifact Removal, *IEEE*, 2018.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, Deep Residual Learning for Image Recognition, *CVPR*, 2015.

[4] Fabian Isensee, Paul F. Jäger, Simon A. A. Kohl, Jens Petersen, Klaus H, and Maier-Hein, Automated Design of Deep Learning Methods for Biomedical Image Segmentation, *CVPR*, 2020.

[5] Jonathan Long, Evan Shelhamer and Trevor Darrell, Fully Convolutional Networks for Semantic Segmentation *CVPR*, 2015.

[6] Olaf Ronneberger, Philipp Fischer, and Thomas Brox U-Net: Convolutional Networks for Biomedical Image Segmentation, *CVPR*, 2015.

[7] Mingxing Tan, and Quoc V. Le, EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, *ICML*, 2019.

[8] Xiao Xiao, Shen Lian, Zhiming Luo, and Shaozi Li, Weighted Res-UNet for High-Quality Retina Vessel Segmentation, *IEEE*, 2018.

[9] Sergey Zagoruyko and Nikos Komodakis, Wide Residual Networks, *BMVC*, 2016