# Computational Intelligence Laboratory 2019

Project report: Road Segmentation on GoogleMap Images

Jingyuan Ma*, Yongqi Wang†, Zhi Ye‡,
Group: +Vx_wangyq977,
ETH Zürich
Zürich, Switzerland
*majing@student.ethz.ch, †wangyong@student.ethz.ch, ‡yezh@student.ethz.ch

***Abstract*—Image Segmentation has been an challenging and yet inviting topic in computer vision. The application of such techniques pave the way to various service, such as the conversion of the satellite images. In this semester project task, road segmentation maps are asked to generate from the aerial images obtained from GoogleMap. In order to achieve concrete segmentation result, a good convolutional neural network (CNN) should be designed to include local information variance in a single image as well as global variance variance across the images. In this project, evaluation on FCN [1] and Bilateral Segmentation Network (BiSeNet) [2], the state-of-the-art method on real-time semantic segmentation on Cityscapes Dataset are compared on the provided image dataset to generate two baselines. From these models, we incorporates ideas from both structures in our proposed network. The performance of our network beats both baselines by successfully incorporation of spatial information with good amount of context information. This project provides inspiration for further investigation on how to include sufficient context information and spatial information for image segmentation tasks.**

## I. Introduction

Image segmentation has a wide range of applications in various fields, such as road autonomous driving and scene understanding. With the assistance of GPU, CNN has been proven as one of the most effective methods to achieve good image segmentation result. The challenge of this semester project is road segmentation from aerial images obtained from Google Map. To segment road area properly, the network should tolerate variance on light conditions, road orientations and similarity between road and non-road areas like parking space. In this semester projects, study on past publications like Residual Network (ResNet) [3], Fully Convolutional Networks (FCN) [1], U-Net [4] and the state-of-the-art method BiSeNet was done firstly. The performance of FCN with U-shape based on ResNet and BiSeNet [2] are set as baselines. Furthermore, our proposed network combining idea from BiSeNet and U-Net are evaluated and compared with two baselines as well. Evaluation metric is pixel-wise root mean square error. Scores for comparison between networks are obtained through submission on Kaggle.

## II. Models and Methods

### A. Base Models

*1) ResNet:* Residual Neural Network is a derivation of CNN, the core idea of ResNet is so called Identity shortcut connection, which means that some layers are skipped [5]. It is constructed with basic block called residual block, shown in Figure 1. With the residual block stacked together, it forms a residual neural network with different layers. Current construction includes 18-layers, 34-layers, 50-layers etc.

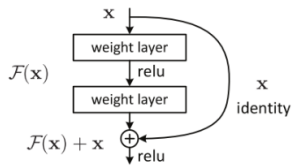



Fig. 1: Residual Block

### B. Network Architecture

*1) FCN with U-Shape based on ResNet:* Departed from original U-Net and FCN, the contraction path (also called as the encoder), which is used to capture the context in the image, is a ResNet. Then, the symmetric expanding path (also called as the decoder),which is used to enable precise localization using transposed convolutions, uses the outputs of different residual blocks. Instead of concatenation of residual block output and transpose convolution, a summation of two is used like that in FCN. Detailed Architecture is shown in Figure 3b.

*2) BiSeNet:* Then, BiSeNet is constructed with two paths: Spatial Path and Context Path. In spatial path, a shallow CNN is implemented. In context path, a deep neural network is implemented to extract rich context information. And an attention refinement module is added to fuse all context information. In the end, context information and spatial information are combined in feature fusion module. Detailed structure is shown in Figure 2.

*3) Proposed network:* Similar to BiSeNet and U-Net, our network ResNet as contraction path, and adopted ARM for more context information and FFM module for fusing spatial and context information in the expansion path information. Detailed architecture are shown in Figure 3a.

### C. Data Prepossessing

In this semester project, there are 100 aerial images and corresponding ground truth with size $400 \times 400$ pixels in the training dataset. In test set, there are 94 aerial images in test

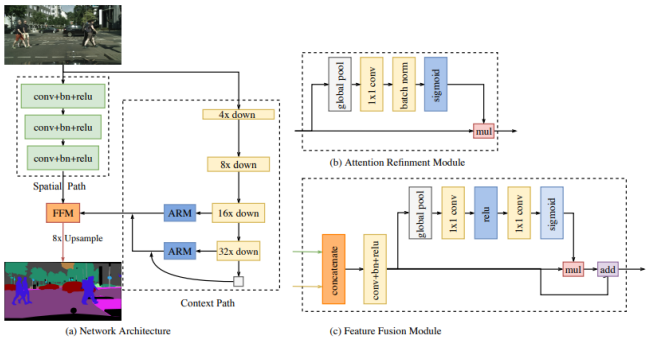
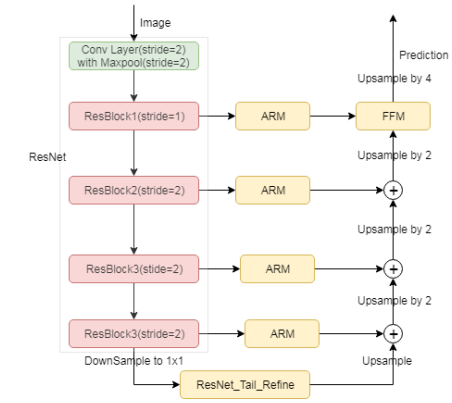
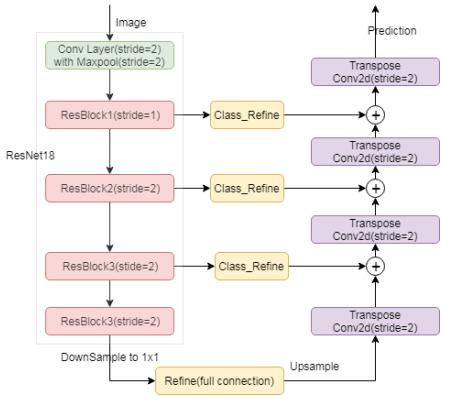



Fig. 2: BiSeNet Architecture




(a) Proposed.




(b) U-Net

Fig. 3: Model Architecture

data set with size $608 \times 608$ pixels to evaluate our model in Kaggle system. Training dataset is split into train set and evaluation set at random with a ratio of $90 : 10$ prior to training.

Before training, visual inspection of the training dataset reveals some potentials factors that might hinders or affect the performance. Assuming the labels were accurate in the training set, separation from "road" and "non-road" are overall dominated by mostly following aspects in a satellite image.

- **Parking lot (-)**: Paths in the parking lot would generally be identified as "road" if connected to the main road and not presented in the rooftop. However, the model has to distinguish the rather small variations, especially around the edges.
- **Rooftop (-)**: Difficult for detection especially because the colors and the patterns can be indistinguishable from road even by human inspection. Successful labelling requires identification of minor details on the rooftop to be incorporated and the ability to well separate the edges in the borders of the large building from models.

In training process, original image are normalized. The grey-scale images of ground truth are read, with value in $[0, 255]$. Binary label of $0, 1$ are assigned according to a binary filter. A threshold of 50 is applied during transformation: if below, the pixel is assigned to $0$; otherwise, $1$. The images is scaled at random from a array of $[0.75, 1, 1.25, 1.5, 1.75, 2.0]$. The scaled images and transformed ground truth are later cropped randomly for the purpose of image augmentation. This step will add more variance to the original dataset.

### D. Data post-processing

To obtain the submission table, the predicted grey scale images are partitioned into patches with size $16 \times 16$ pixels to form labels $0, 1$ to each patch following the guideline provided. The patching is done by a binary filter of $50\%$. In the evaluation set, both prediction and ground truth are patched in this way, while in the test set, only prediction is patched for submission on Kaggle.

### E. Setup and hyper-parameters configuration

PyTorch 1.0 [6], CUDA 9.0 [7], Apex [8], easydict are necessary packages and software for the project. [1].

- Hardware Device: NVidia Xp collection with 12 GB memory.
- Training setting: Adam Optimizer with initial learning rate of $1e^{-3}$.
- Batch size: 16
- Loss function: cross entropy loss with two classes [2]

During training, models usually converges around epoch $40$ to $60$ [3]. Without special specification, the prediction in the figures below were obtained using the same configuration at epoch $40$.

Parameters were saved every five epochs for evaluation purpose.

## III. Results

### A. Architecture result inference

From Table Ia, we can clearly see that the performance of BiSeNet is the worst followed by u-shape network. The proposed network achieved best result with public score of $0.89165$ and private score of $0.87208$. By looking at patch test

[1]Environment can be easily replicated and set up using docker container, see https://github.com/wyq977/cil-road-segmentation-2019 for more details

[2]Stochastic gradient descent (SGD) with poly learning rate policy in BiSeNet

[3]run times depends on the networks structure spanning from 12 min to 22 min per epoch

| Architecture | Public | Private | Speed(FPS) | Parameters[a] |
|---|---|---|---|---|
| BiSeNet | 0.87798 | 0.85545 | 134.84 | **12.89** |
| FCN with U-Shape | 0.88194 | 0.86472 | **200.76** | 36.87 |
| Proposed Network | **0.89165** | **0.87208** | 123.87 | 12.94 |

(a) Different Architecture with Same ResNet as Base Model

[a]size in MB

| Model | Public | Private | Speed(FPS) | Parameters |
|---|---|---|---|---|
| ResNet18 | 0.89165 | 0.87208 | **123.87** | **12.94** |
| ResNet34 | 0.89407 | 0.87517 | 89.55 | 23.05 |
| ResNet50 | **0.90328** | **0.88567** | 95.00 | 28.83 |
| ResNet101 | 0.90169 | 0.88407 | 57.81 | 47.82 |

(b) Proposed Network with Different ResNet Depth

TABLE I: Comparison of Network Architecture on Performance and Accuracy

images of different architectures in Figure 4, we can see that u-shape network successfully extracted sufficient details while failed on generating large road patches. For BiSeNet, on the other hand, it is an opposite of FCN with U-shaped Network, given that large road patches are generated well but details are lost. For our proposed network, the large road patches is enlarged at a minor loss of details on small road patches. In addition, we run speed and parameters evaluation. U-shape network achieved 200 FPS with the largest parameter size, while BiSeNet achieved 134 FPS with least parameter size of 12.89 MB, which is almost identical as our proposed model in size. Speed of proposed network is 123.87 FPS

### B. Model result inference

From Table Ia, as base model goes deeper, both public and private scores are higher until resnet50. For network with resnet101, the score is roughly about the same as that of base model of resnet50. Public score of resnet 50 is 0.90328, and private score of resnet 50 is 0.88567. By looking at patched test images of different base model in Figure 5, we can see that context information are gradually enlarged at a cost of spatial information until base model of resnet50. For network with resnet101, indeed both spatial information and context information is a bit more compared with that of network with resnet50. However, more wrong prediction has been made. For different base model, we run speed and parameters evaluation as well. As parameter size increases, network speed decreases. As a result, base model of resnet101 is the slowest one.

## IV. Discussion

### A. Architecture

In this road segmentation task, the road class prediction includes not just large context patches of wide roads like highway or main street, but also details of small driveways in neighborhood and drive-throughs of parking lot. Three architectures are tested. For BiSeNet, due to the existence of attention refinement module, large context patches are predicted correctly. However, spatial path of BiSeNet is so shallow that prediction on details like drive-throughs fails. For U-shaped Network, more details are included compared with BiSeNet, but large road patches are lost due to lack of context information. For our proposed network, details and large patches are both preserved the best. That is because feature fusion module takes input from output of the first residual block instead of the output of spatial path in BiSeNet. That ensures receptive field is large enough to include details and inference between large context and details. In addition, by adopting U-shape in our proposed network, it ensures that the rich context information by fusing receptive fields of different sizes.

### B. Model

From Table Ib, we can see that as the base model goes deeper, the performance of network gets better until base model of resnet 50. Theoretically, the deeper the network, the larger the receptive field is. However, in our task, the training image is of size 400x400. That leads to a problem that is the network goes too deep, the receptive fields will be so large that overlap of receptive field on central area. That causes severe disproportion of information between border area and central area. By looking at the prediction images of resnet50 and resnet101 in Figure 5, one can tell that resnet101 has more error prediction at the border and missing prediction of large road patches than that of resnet50.

### C. Speed and Parameter size

By analyzing speed and parameter size reported in Table I, we can see that with the same architecture, more parameters means slower speed. However, for different architecture, parameter size does not have direct influence on network speed. In general, network architecture of parallel computation will accelerate the network.

### D. Potential Improvement

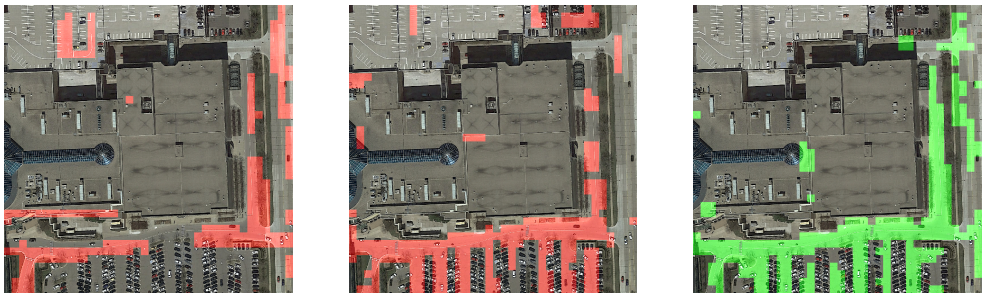
Due to the resource constraint, only single Titan Xp GPU is used in this project, except the version of resnet101. So, training can only be done in size of $400 \times 400$, although prediction is made on size of $608 \times 608$. That actually causes errors. If given enough resources, image can be re-sized to $608 \times 608$. Then network takes re-sized images as input during training.

By comparing high confidence image with low confidence image shown in Figure 6, we can tell our proposed network failed in complex scenes like the low confidence image with rooftop, roads with different luminosity. Solution for this will be more data augmentation on images with complex scenes in train set.

## V. Summary

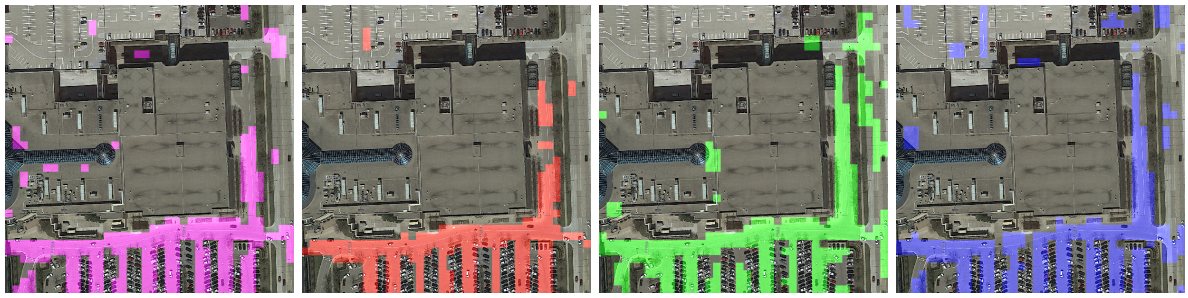
Given the discussion presented above, some conclusions can be made as followed.

- In order to generate good segmentation result, a successful fusion context information and spatial information is critical. Through this project, we can see that by extracting a weight through convolutional layers and



(a) BiSeNet (b) U-Net (c) Proposed Network

Fig. 4: Prediction on Test Image for Different Network Architecture with Same Base Model



(a) ResNet18 (b) ResNet34 (c) ResNet50 (d) ResNet101

Fig. 5: Prediction on Test Image for Proposed Network with Different Base Model.

(a) High confidence. (b) Low confidence.

Fig. 6: Variance in Prediction on Different Images from Test Dataset

applying before output is a successful way of fusing context information with spatial information

- Unlike semantic segmentation, road segmentation requires both rich context information and spatial information. Therefore, the trade-off between these two has to be done.
- Potential improvement for further work includes transformation of training image size and adding more data augmentation on specific training images.
- Speed of the network is influenced heavily by network architecture; parallel computation might increase speed. Also, parameter size has an impact on speed as well, if given a specific model.


## ACKNOWLEDGEMENTS

The authors thank BiSeNet authors for releasing their code on github [4]. It serves as a reference for implementation of this project.


## REFERENCES


[1] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *CoRR*, vol. abs/1411.4038, 2014. [Online]. Available: http://arxiv.org/abs/1411.4038

[2] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Bisenet: Bilateral segmentation network for real-time semantic segmentation," *CoRR*, vol. abs/1808.00897, 2018. [Online]. Available: http://arxiv.org/abs/1808.00897

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[4] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *CoRR*, vol. abs/1505.04597, 2015. [Online]. Available: http://arxiv.org/abs/1505.04597

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," *CoRR*, vol. abs/1603.05027, 2016. [Online]. Available: http://arxiv.org/abs/1603.05027

[6] "Pytorch," 2019. [Online]. Available: https://pytorch.org/

[7] "Cuda," 2019. [Online]. Available: https://developer.nvidia.com/cuda-zone

[8] "Apex," 2019. [Online]. Available: https://nvidia.github.io/apex/


[4] https://github.com/ooooverflow/BiSeNet, https://github.com/ycszen/TorchSeg/