Road Detection Based on U-net Architecture

Wanli Wang, Junkai Wang
Baylor College of Medicine

Abstract

Automatic road extraction from remote sensing imagery plays an important role in many applications. However, extracting road from very high-resolution images accurately and efficiently still remains challenging due to some limitations, including increased data size and superfluous details, the spatial and spectral diversity for road targets, disturbances (for example trees and buildings), the necessity of riding weak road edges while avoiding noise, and the fast-acquisition requirement of road information [1,2]. Here, we take the advantage of the U-net architecture, a strong network and training strategy that relies on the strong use of data augmentation to use the annotated samples more efficiently. we attempt to solve the task of detecting roads from ten thousands satellite images by implement U-net.

Introduction

Road detection and classification from satellite imagery has been an active area of research for the past decade. We consider road detection as a semantic segmentation task. The goal of semantic segmentation is to label each pixel of an image with a corresponding class of what is being represented. So for road detection task, the label for each pixel only has two classes: road and non-road.

The most popular architecture for segmentation is the Fully-Convolutional Network (FCN) [3]. The FCN consist of only convolutional and pooling layers, without fully connected layers. FCN shows a quite good performance on segmentation tasks. However, one issue is that the direct predictions of FCN are typically in low resolution, resulting in relatively fuzzy object boundaries.

A group from University of Freiburg modified and extended FCN architecture such that it works with very few training images and yields more precise segmentations. This more elegant network is called U-Net [4]. The main idea is to supplement a usual contracting network by successive layers, where pooling operations are replaced by upsampling operators. Hence these layers increase the resolution of the output. What's more, a successive convolutional layer can then learn to assemble a precise output based on this information. U-nets have originally developed for biomedical image segmentation, but they are also used for many different applications. We decided to use U-Net to solve our road detection task here.

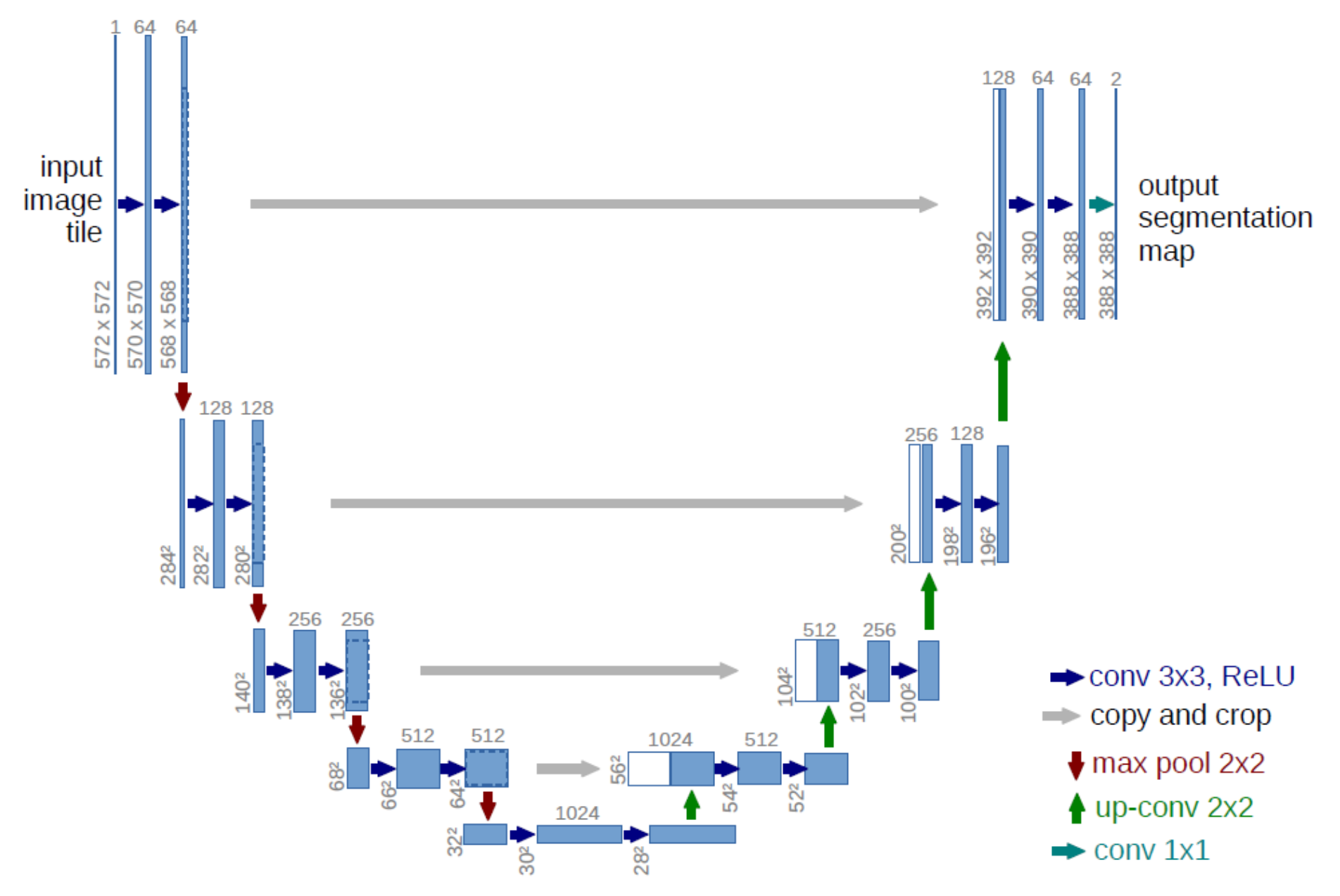


Figure 1. U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

Methodology

(1) Data processing
Kaggle provides a training set consisting of more than ten thousand satellite images and corresponding Boolean masks, for the purpose of building up a neural network. Additionally, we will use our well-trained model to generate predictions on an unlabeled (without the masks) set of images. All the images and masks in the project come from DeepGlobe (http://deepglobe.org). The satellite images cover a wide range of landscapes and each image covers roughly 16 acres of land in a 256 m by 256 m cube. Each image is of size 512 by 512 resulting in a pixel resolution of 50 cm. For the Boolean masks, the 'road' pixels are labeled as 1, so the other not "road" pixels are then labeled as 0.

(2) Data augmentation
Even though our training data size is already quite large, we believe it is necessary to conduct data augmentation before training U-Net model. We used the package called "imgaug" (https://github.com/aleju/imgaug), which is designed for data augmentation purpose. We used some methods for training dataset, including horizontal flip, vertical flip and ninety-degree rotation. One representative example is shown in Figure 2.

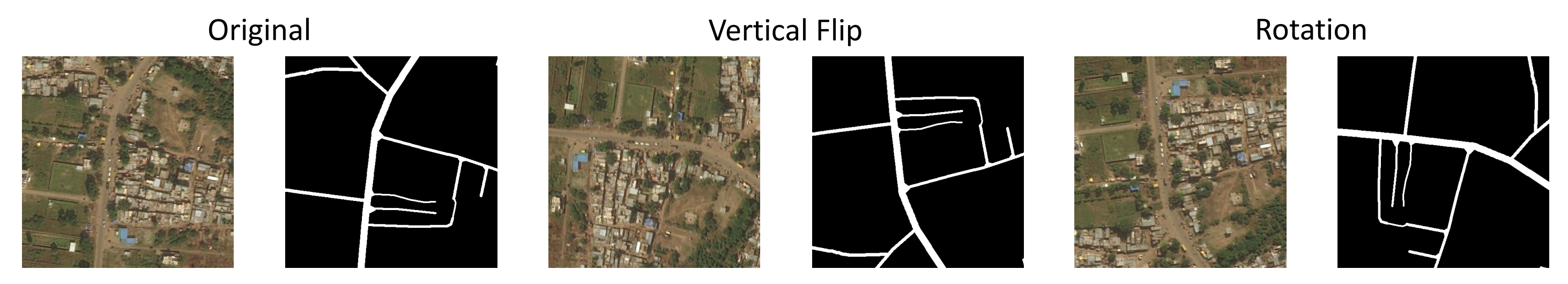


Figure 2. The representative data augmentation example for one satellite image and corresponding mask from training set.

Methodology

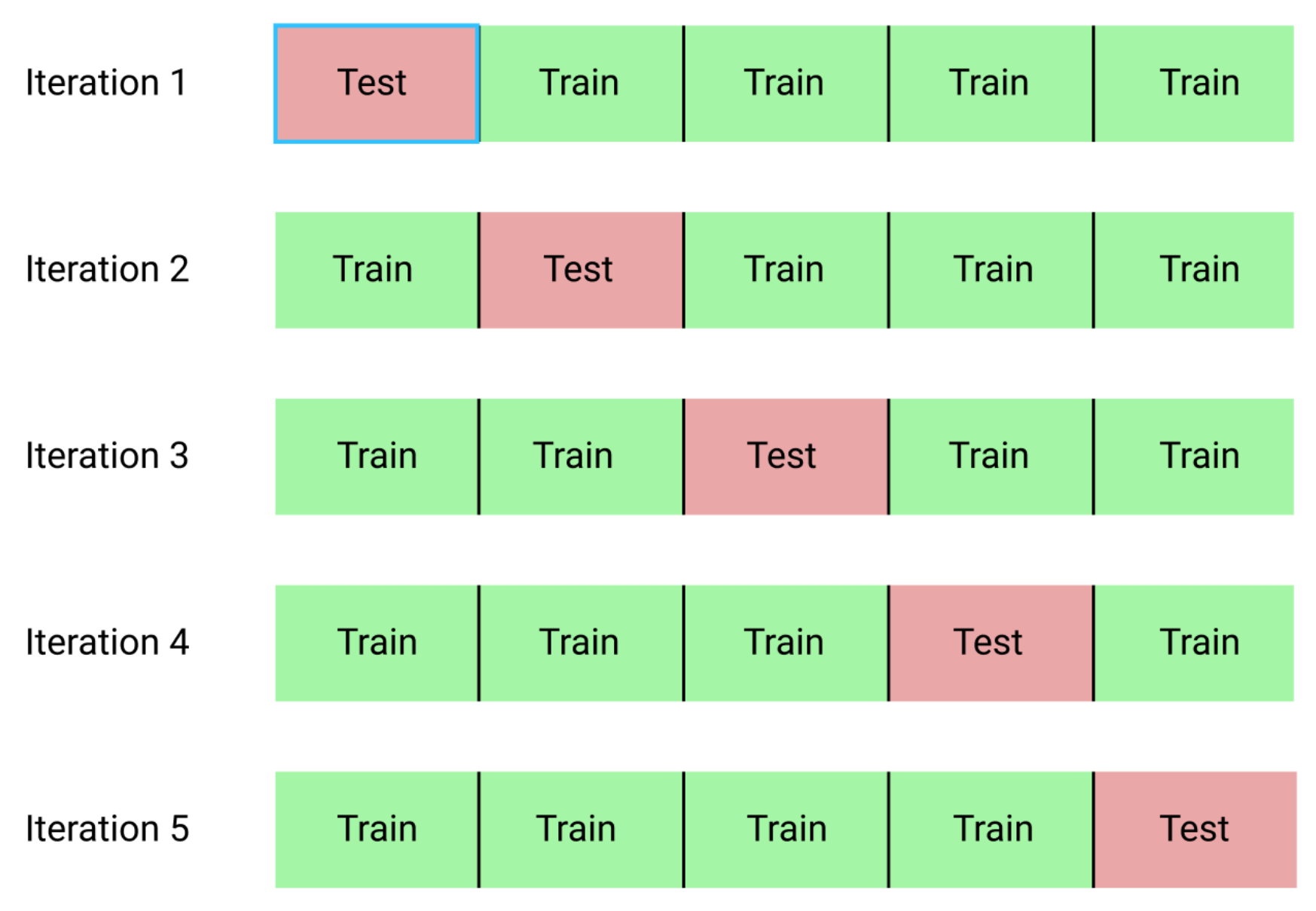
(3) U-Net
The U-Net architecture is illustrated in Figure 1. It consists of a contracting path (left side) and an expansive path (right side). In our training procedure, we use "binary cross-entropy loss" as the loss function, defined as equation (1).

$$CE = -\sum_{i}^{C} t_i log(s_i) \qquad (1)$$

(4) K-fold Cross-Validation
Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample. k-fold cross-validation split the given data sample into k groups. Compared to other methods, like simple train/test split, K-fold Cross-Validation always performs better and has less biased results.
The general procedure is as follows:

1. Shuffle the dataset randomly
2. Split the dataset into k groups
3. For each unique group:
   a. Take the groups as a validation set
   b. Take the remaining as train set
   c. Fit a model on train set and validate it by validation set.
   d. Apply the model to the next training set, and obtain a final model based on all the training processes. Or, retain the evaluation score for every model after one training process and then discard the model, average the scores.

The pipeline for K-fold cross validation is shown in Figure 3.



Iteration 1	Test	Train	Train	Train	Train
Iteration 2	Train	Test	Train	Train	Train
Iteration 3	Train	Train	Test	Train	Train
Iteration 4	Train	Train	Train	Test	Train
Iteration 5	Train	Train	Train	Train	Test

Figure 3. Pipeline of K-fold cross-validation

Result

Table 1 shows the comparisons among different parameter settings of U-Net architecture. Shallow U-Net displays relatively simple architecture, having 32 filters for the first layer. In contrast, deep U-Net has 64 filters and shows higher complexity. Batch size changes due to the limitation of the memory. Model2 uses K-Fold cross validation, comparing to model1. As expected, complex U-Net achieves higher score, and shows better predictions for training data sets and test data (Figure4).

Table 1

	model 1	model 2	model 3
U-Net	Shallow	Shallow	Deep
First Channel number	32	32	64
Batch Size	8	8	4
Train/Val split method	Simple	K-Fold CV (k = 8)	Simple
Epoch	30	10 * 8	20
Score	0.6562	0.6223	0.6978

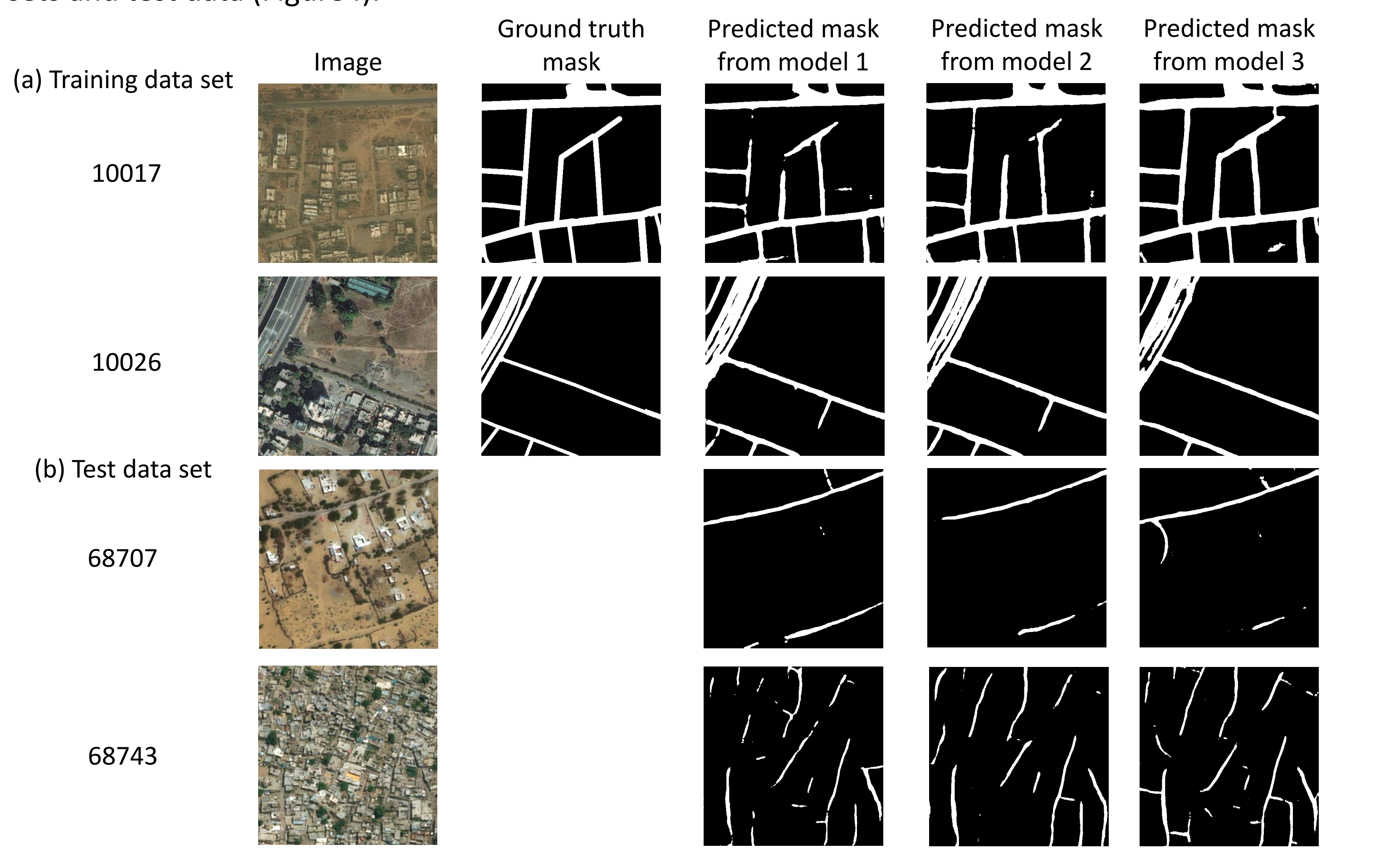


Image
Ground truth mask
Predicted mask from model 1
Predicted mask from model 2
Predicted mask from model 3
(a) Training data set
10017
10026
(b) Test data set
68707
68743

Figure 4. Comparisons among different settings of U-Net Architecture on training data and test data. Representative predictions are shown in training data (a) and test data (b).

Result

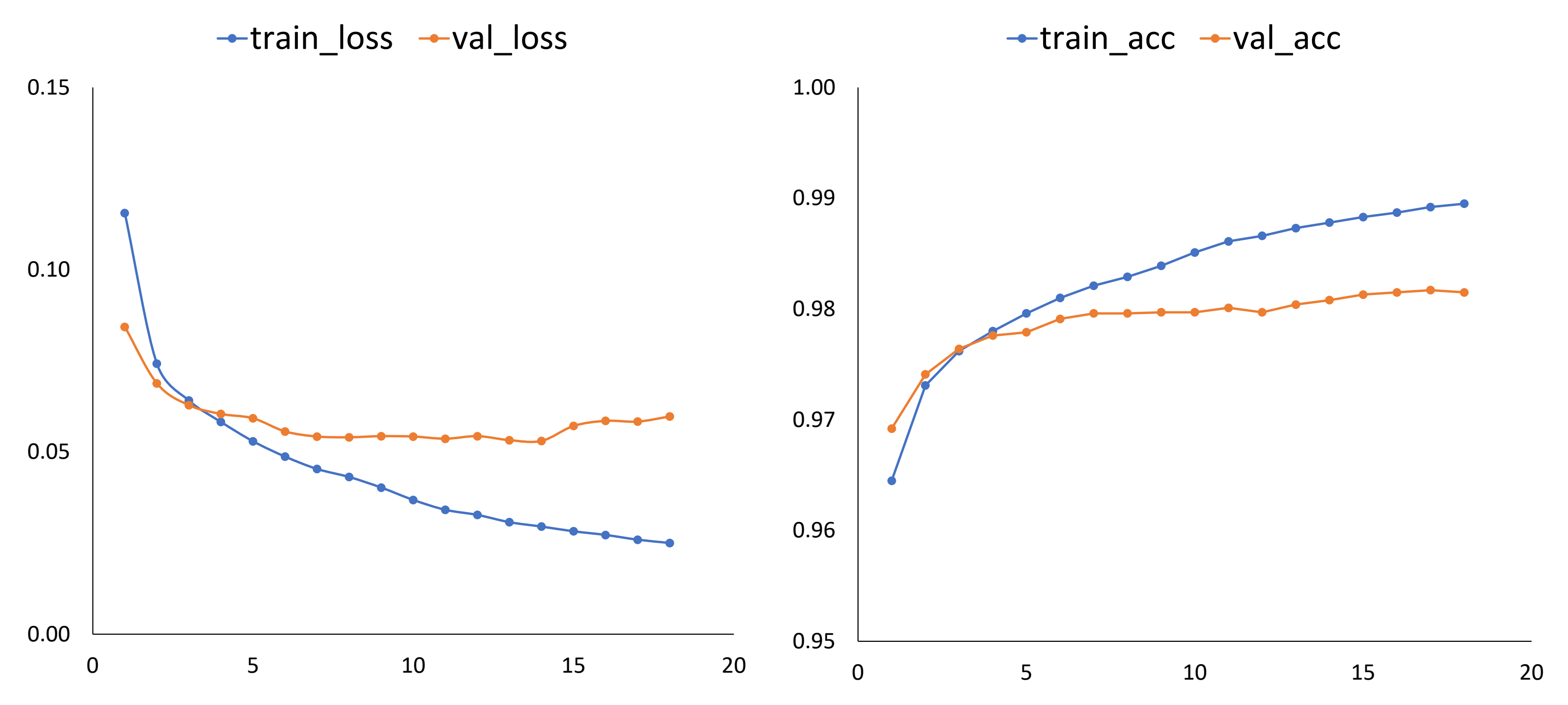


Figure 5. The loss of training data and validation data is shown on the left panel for Deep U-Net architecture with 20 epochs. The training loss keeps dropping along with epochs, but the validation loss reaches the plateau stage after 10 epochs, indicating the convergence of the model. Similarly, the accuracy of training/validation data shown on the right panel demonstrates that Deep U-Net is pretty well-trained after 10 epochs. Overall, the train accuracy and loss accuracy are pretty optimal, greater than 97% after few epochs.

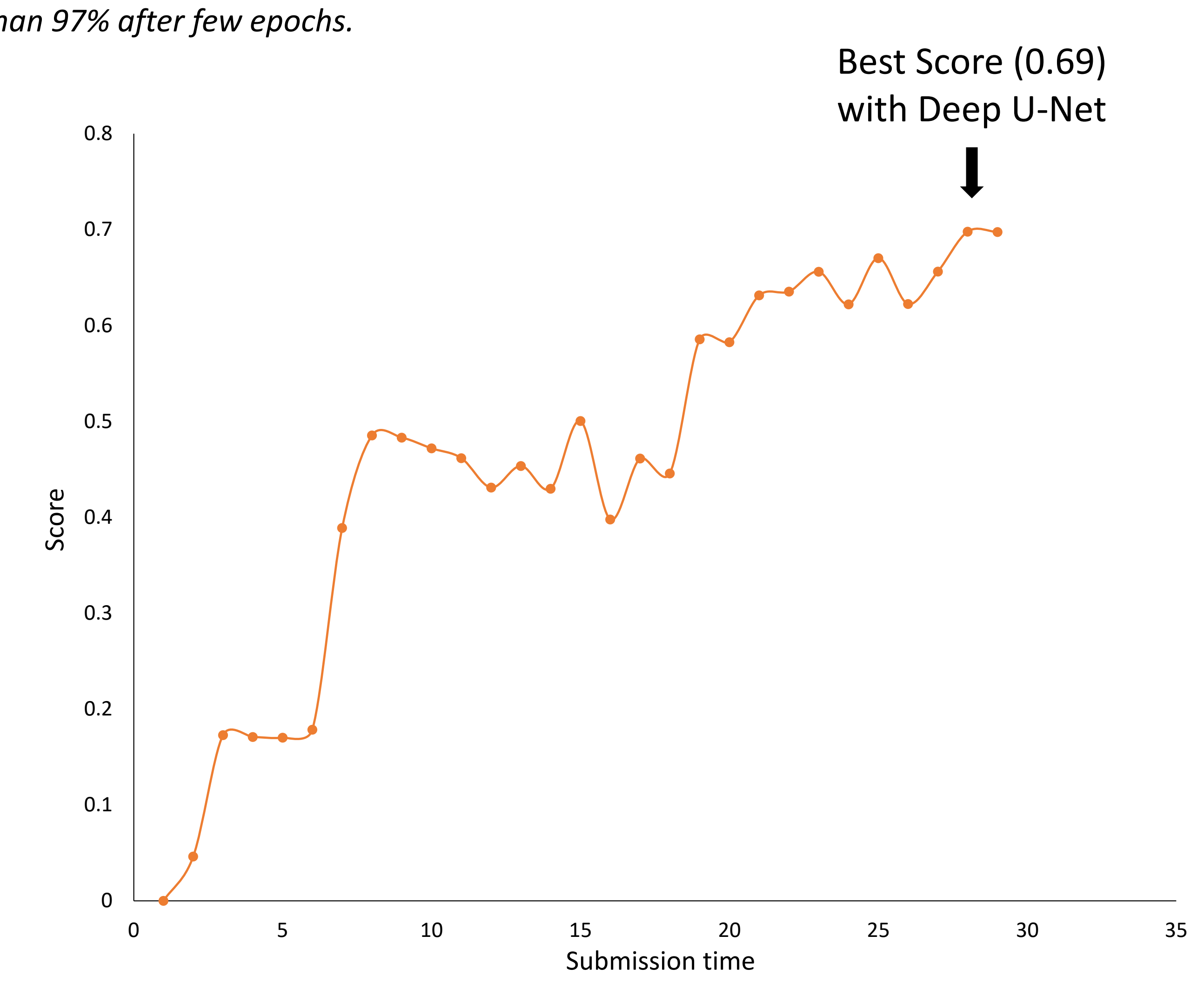


Figure 6. Score changes along with submission time. We have highest score with the Deep U-Net (0.69).

Reference

[1] J. Cordonnier, A. Schneuwly, and J. Zenh, "Machine Learning — Project 2 — Road Segmentation," vol. 25, no. 224633, pp. 0–3, 2014.

[2] L. Chen, Q. Zhu, X. Xie, H. Hu, and H. Zeng, "Road Extraction from VHR Remote-Sensing Imagery via Object Segmentation Constrained by Gabor Features," ISPRS Int. J. Geo-Information, vol. 7, no. 9, p. 362, 2018.

[3] E. Shelhamer, J. Long, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 4, pp. 640–651, 2017.

[4] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 9351, pp. 234–241, 2015.

Acknowledgement

We want to express our sincere gratitude to Professor Devika Subramanian and all TAs of course COMP 540. This project wouldn't be finished without their generous guidance and help. We also thank DeepGlobe (http://deepglobe.org) for making this dataset available for academic use.