Landscape Classification (CS 216 Project)

Chenhao Li

June 13, 2020

Abstract

This is a challenge dataset of UCI Machine Learning Hackathon[1]. A map of Chbar Mon, Kampong Speu Province of Cambodia. The task is to classify six primary interest areas in order to take research on the spread of Dengue fever. A classified subarea of Chbar Mon is provided. We use the idea of semantic segmentation to classify the landscape. The ResNet50 [1] is used as feature extraction encoder, and PSPNet [2] as decoder.

1 Introduction

A research group from UCI is interested in the relationship between Dengue virus and environment. They want to research on whether different land influence the spread of the virus in the Southeast Asia. They provided this data as challenge dataset in UCI Machine Learning Hackathon. The main task is to classify size different land type they are interested in: water, rice field, paved road, vegetation, trees, and buildings. They provided a single classified area so I create my own training data by cropping and rescaling to size between 100 and 150 pixels. I also controlled stride so that most of the training data will be distinct. I use ResNet50 as the feature extraction model, and apply PSPNet to gather the local and global information of feature to do the classification. The model is trained on the data I prepared, and the performance is evaluated based on the classification on the large images that have size larger than the overall size of training data and small images which have size smaller. The result of classification is desired. However, it needs further improvement in details. Furthermore, since the data provided is limited, either training or evaluation is biased. It is not so credible to evaluate the result from the same source. The evaluation of this project can only be viewed as a reference on how semantic segmentation model works on classifying land type on satellite image.

2 Data

The original data is the map of Chbar Mon, Kampong Speu Province of Cambodia. The task is to classify the six land types over the area. However, the data can be used to train

[1] https://uci-ml-repo.github.io/events/hackathon20/

the neural network is few. The only classified data is provided from a part of Chbar Mon area as shown in Figure 1. Therefore, it is necessary to make more training data.

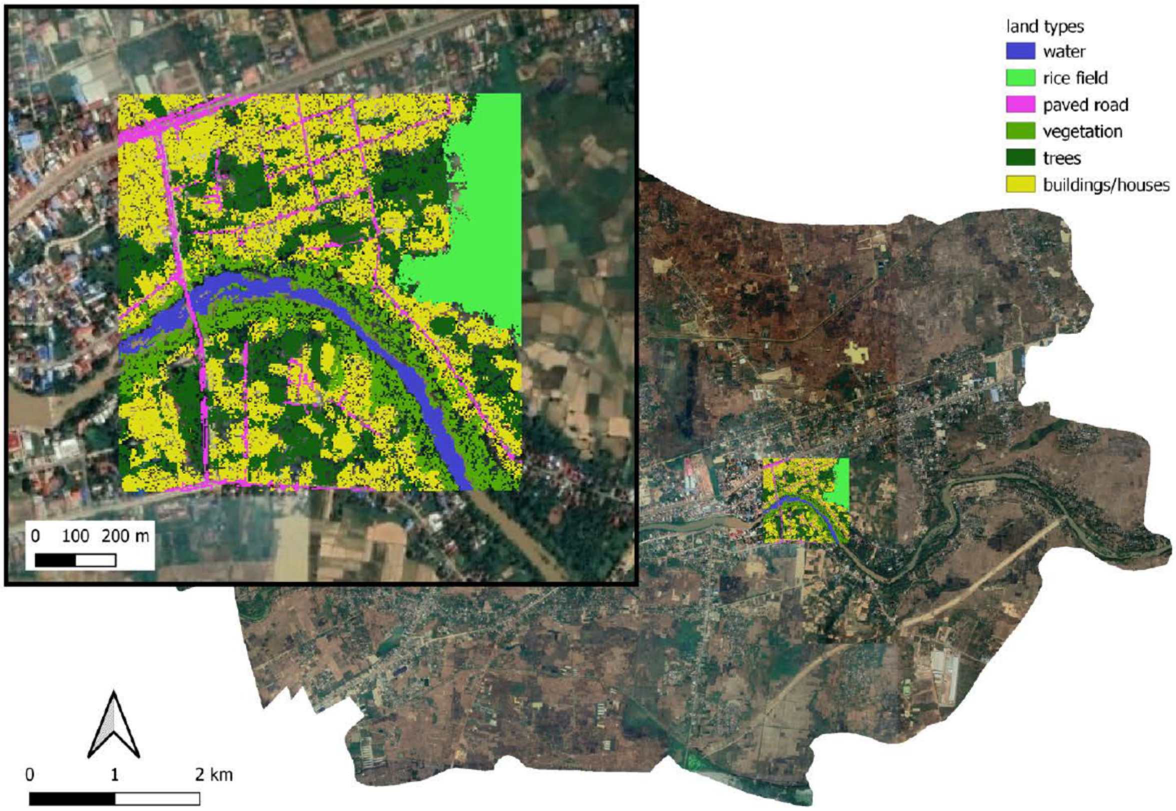

Figure 1: Classified area of Chbar Mon Provided by Daniel M. Parker, UCI

The method I use to produce more training data is to crop and rescale the original classified data. However, it is essential to notice that we do not want too small image. Otherwise, there is limited information we can use to classify since we may crop an area of one class and we cannot use it to distinguish it from other types. Moreover, we also do not want large area. If the image is close to the original classified map, and when we want to evaluate our model, we would have high accuracy and this may cause overfitting which we do not want to see.

Therefore, I choose the pixel range from 100 to 150 pixels as potential side length of cropped and rescaled image. Considering similarity of cropping images, I control the stride that the images with same side length are at least different in 10 pixels. Moreover, the start point of the first image is randomly picked from the 50 x 50 pixels wide area at the top left.

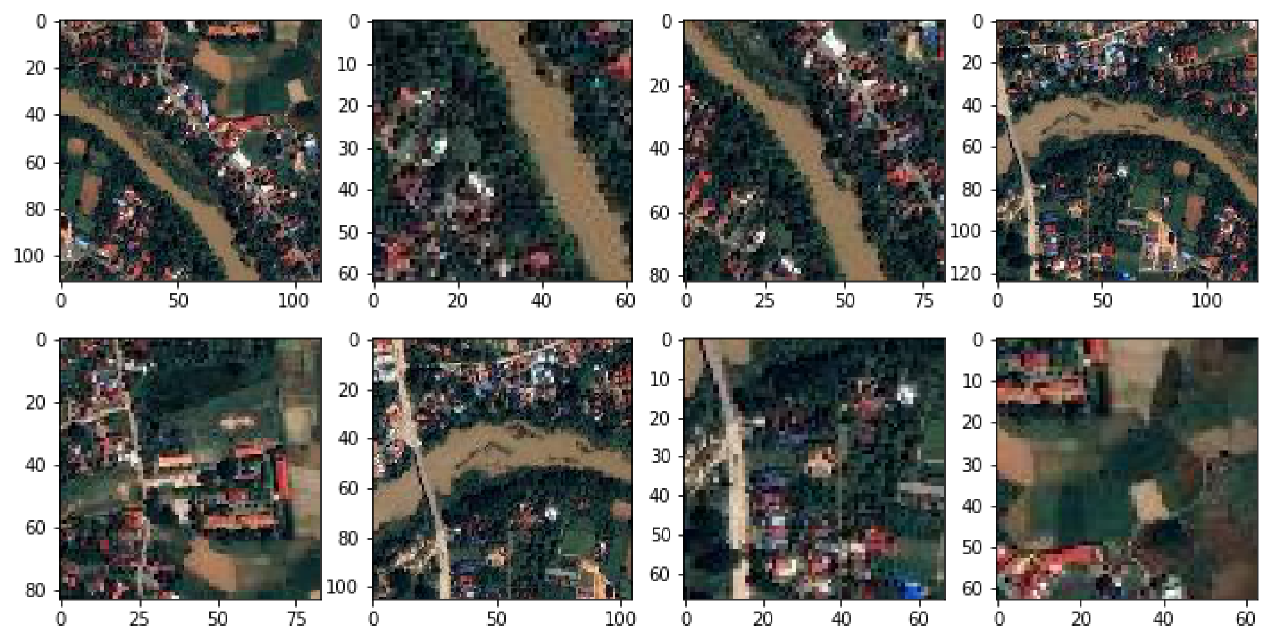

Figure 2: Examples of training data

In addition, the number of training data prepared should in a reasonable size. Since the provided classified data is small (250 x 250 pixels). It is hard to construct large data set if considering the similarity and size of training data. Also considering the possible overfitting phenomenon and training time, I control the number of cropped and rescaled training data below 10000 but more than 5000 images. Some example training data are shown as Figure 2.

3 Methods

The basic idea of the model I used is for semantic segmentation. The overall structure of neural networks for semantic segmentation consists an encoder, which performs the feature extraction, along with a decoder, which perform the main task of segmentation based on the extracted feature. The encoder I used for this project is ResNet50 [1], and the decoder I used is PSPNet [2].

3.1 ResNet

Residual neural network (ResNet) is widely used in the image classification task. It also has good performance in doing feature extraction. The basic architecture of ResNet is the residual block, which contains a double- or triple- skips that contain nonlinearities (ReLU) and batch normalization in between. We use typical ResNet50 (50 residual blocks) pretrained on ImageNet [3] dataset.

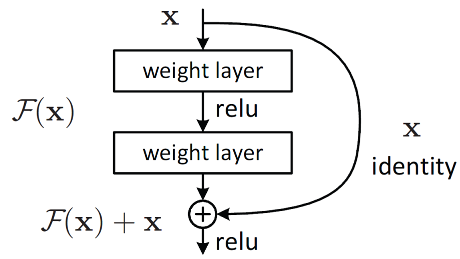

Figure 3: A residual block

3.2 PSPNet

PSPNet is known as Pyramid Scene Parsing Network. It is developed by Hengshuang et al. The network aggregate the global context information by different region through proposed pyramid pooling module. The module applies upsampling to the feature expression gotten from the feature extraction. Then concatenate sub-regions with the global context information to the last convolution layer and get the final pixel-level prediction.

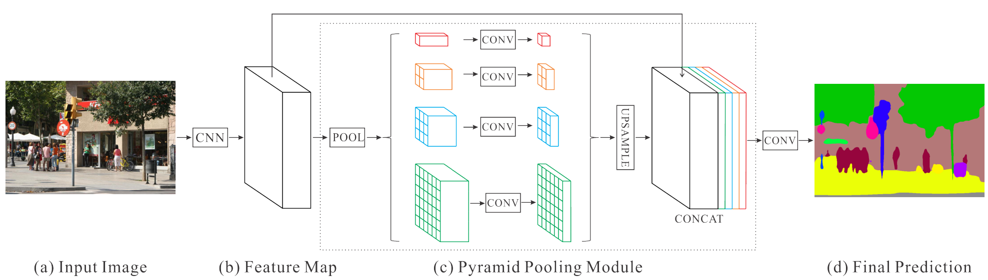

Figure 4: PSPNet architecture by Hengshung Zhao et al.

4 Results and Evaluation

Since we are only given one image with classification result, the result evaluation will only based on the image provided. Moreover, the training data are built based on the provided image, so there will be bias in evaluating the goodness of the result.

In order to increase the credibility of evaluation. The test image used will contain two parts. One is original image. Since this is the one I did not use as training data, it could be good to know the prediction results on large scale image while the model is trained based on small images. Another part of testing images are images that have side length smaller than 100 pixels. It is necessary to check whether higher resolution images can yield better prediction result.

4.1 Large Scale

The prediction of original image is shown as Figure 5. The numerical values are shown in Table 4.1. The overall prediction is desired, but for some specific land types, we did poorly in prediction. From the classification map and IoU we calculate, we found that the accuracy of paved road is low. There are lots of areas in the original map that are classified as paved road while the prediction we generated has few area indicating the paved road. It could be possible that the width of paved road in the original data is small (compared with the large size of other land type). Since paved road consists small number of pixels it is hard to predict them as nearby pixels could influence the prediction of those areas. Though the overall shape of each area in classification matches. In details, the prediction mix up trees and buildings. That is areas of trees are predicted as buildings and vise versa. This may because in the most part of original data, trees and buildings come together and consist the most part of image. Moreover, the original image is not high resolution. Therefore, it is possible that the color of trees and buildings does not distinguish from each other.

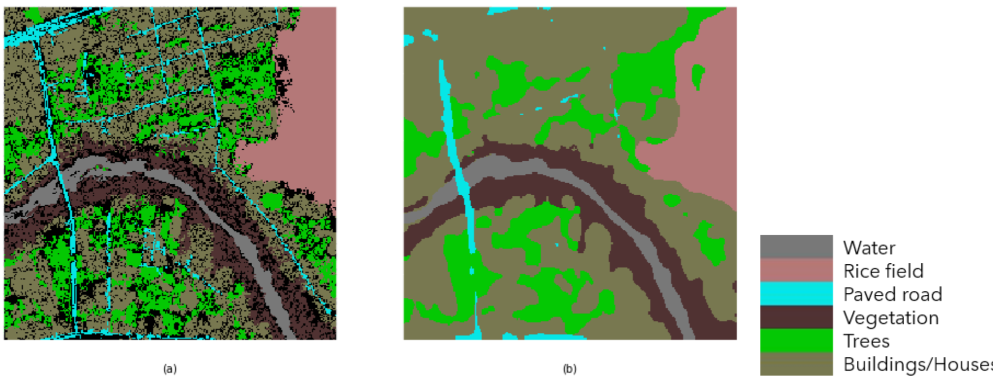

Figure 5: The original image classification: (a) ground true classification (b) predicted classification

	mean IoU	water	Rice field	Paved road	vegetation	Trees	Buildings
IoU	0.5675	0.7669	0.8634	0.2066	0.5572	0.4606	0.5503
Pixels accuracy	0.6879						

Table 1: Numerical result of prediction on the original image

4.2 Small Scale

I randomly cropped images that are smaller than 100 pixels. The sizes of images and positions are chosen randomly. The classification map is shown as Figure 6. For the most images, the classification result is desired, and most results yield better prediction than the large scale image (original image). As we can see from the prediction (a) through (c), the area of trees and buildings are distinguished. Though the area of paved road is small, we still correctly predict where and how large it is.

However, it is essential to notice that as the image is too small, the prediction result can vary a lot. For those too small images (graph (e), (g) and (j)), there are areas that are incorrectly predicted. Especially the prediction on the graph (j), the predicted classification is totally different from the ground true classification. The area of water is predicted as rice field, while vegetation is predicted as trees. It is possible to incorrectly predict between vegetation and trees since the color of these two land types are close if looking back to Figure 1 and 2. Moreover, to classify these two area is more based on the nearby pixels: vegetation area is close to water while trees are surrounded by buildings and close to rice field. Therefore, in the prediction we have, we classify the rice field instead of water, then we get trees instead of vegetation. In other words, the problem we have is to classify the water in small images. However, if we take a look at graph (b), (c) and (h), the area of water is correctly predicted. Moreover, in graph (e), some area of buildings is also predicted as rice field. It could be possible that in the original data given, the color of water, buildings, and rice field are very close in small scale images. It is also possible that in predicting, the algorithm use the pattern information along with the color information. Thus, when we deal with small scale images, the pattern information loses, and the model cannot distinguish the land type only based on the color information while the colors are close to each other in the satellite image.

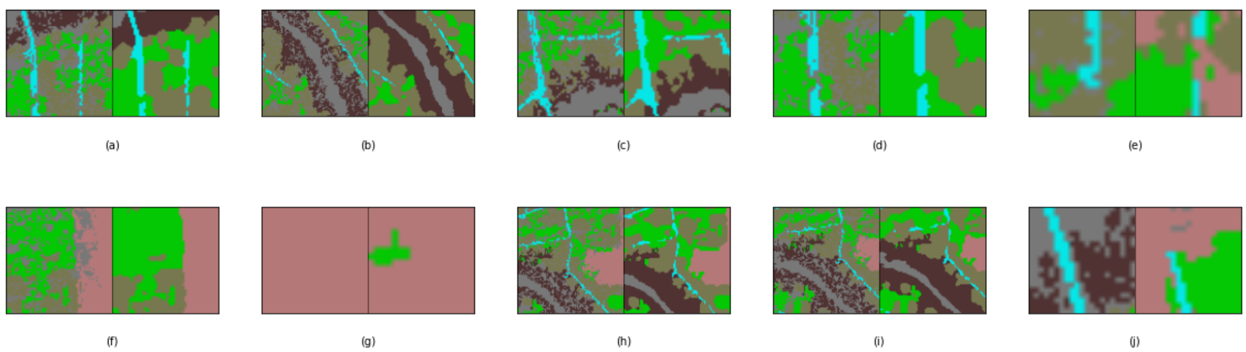

Figure 6: Small scale image comparison. Left image is the ground true classification. Right image is the predicted classification. (a) 61 pixels (b) 96 pixels (c) 52 pixels (d) 40 pixels (e) 15 pixels (f) 58 pixels (g) 18 pixels(h) 93 pixels (i) 93 pixels (j) 21 pixels

5 Conclusions

We achieve a desired result based on what we are given. Because we were not given sufficient data. It is hard to create a machine learning model of semantic segmentation to classify. And we are testing on the same original image to the training data, the evaluation result could be biased. However, the model does classify the different land types from the images. We can see the potential ability of it to separate areas if having sufficient amount of training data. Moreover, the size of training data also influence the accuracy of model's classification. The accuracy becomes lower for test images larger or smaller than the overall size of training data. However, the overall classification result is desired. In details, it needs further improvement. This is also can be solved if having training data with different size beforehand.

6 Future Work

It is better to have more training data, and those data should have various size. It is also good to try a different model for feature extraction and decoder to find one best fit to the land type segmentation. To improve the credibility of evaluation, we also need to have test images that are not from the same source as training data.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[2] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2881–2890, 2017.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A Large-Scale Hierarchical Image Database,” in *CVPR09*, 2009.

A Code

A.1 mitcv

The PyTorch implementation package[2] of semantic segmentation created by MIT CSAIL Computer Vision group.

A.2 train_test_builder.py

Written by my own to build the training data set. And in the format that can be used for the semantic segmentation package model.

A.3 main.py

The modified version of training and testing function from the semantic segmentation package.

[2] https://github.com/CSAILVision/semantic-segmentation-pytorch