CS-433 Machine Learning Project 2
Road Segmentation

Vahid Shahverdi , Vladislav Levitin , Seyedehfatemeh Arfaeezarandi
Ecole Polytechnique Federale de Lausanne, Switzerland

***Abstract*—In this report, we are presenting our approaches to classify satellite images into two principal segments, which are road and background.**

There are some difficulties in this approach as it can often be difficult for the trained model to distinguishing between road and background. In one of our approaches, we built a model and extracting more than 140 features from the RGB colors of the images. The result was relatively satisfying, but it cost enormous effort to generate appropriate features. This fact leads us to use Convolutional Neural Networks, which is suitable for the aim of this project and gives a precise result. We got our best result using U-net model with 90.6% F1-score and 95% percent accuracy on the test set.

I. Introduction

High processing power, uanbles us to perform extraordinary tasks such as image segmentation with the help of machine learning. The purpose of Image segmentation is dividing an image into sets of pixels, in which these sets carry some standard features. This action brings us many applications, such as the road segmentation, developing autonomous vehicles, identifying meteors and Chariots, and recognizing cancer cells from a blood test. The aim of this project is the segmenting of satellite images into two classes, which are road and background. Our data set includes images that are available from Google satellites, which mostly represent urban areas. First, we should be aware that these images are from high altitude, so distinguishing roads from background can be quite difficult when cars, trees, shadows might cover roads. Additionally, some structures such as parking lots, railroads, rooftops, or more generally, structures with boundaries look very similar to road, but should not be classified as road. There are some useful models such as U-net to help us overcome this issues. The U-net architecture originates from the fully convolutional neural network,first proposed by Long and Shelhamer[1]. The main idea behind the U-net is to use successive down-sampling layers, where pooling operations are replaced by up sampling operators, and the down sampled output is then upsampled to the original input size.

II. EXPLORATORY DATA ANALYSIS

Our training data set includes 100 satellite images, and each image has 400x400 pixels, with each pixel consisting of the RGB colors. Every image in the training data set has a ground truth image is associated with it. The ground-truth is a black and white mask, in which white pixels belong to roads, and black pixels belong to the background(figure 1). Our task is to perform the binary classification of patches with the size of 16x16, where a patch is considered as a road if more than 25 percent of its pixels are labeled as such.

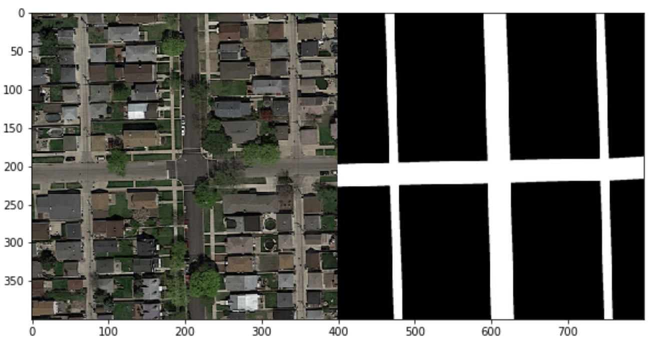

Fig. 1. Example of a picture and the corresponding ground truth

Besides the difficulties mentioned in this project, we faced other challenges such as when the color of a road had significant variation in color due to shadows and trees that were covering it. Besides, many other ambiguities such as parking lots, railroads, etc. are present in the satellite images. They should not be considered as road, but they are carrying some standard features as the roads. At last, our data set mostly includes images where roads are either vertical or horizontal, which gives bias. To handle this problem, we rotated the training set by 45 degrees.

III. Scoring

Besides the ordinary accuracy, we need a better measurement for our prediction; This is lying on the fact that our images approximately consist of 75 percent as background. So high accuracy, such as 75 percent, does not necessarily means a well perform. Therefore, we consider the F1 score in this report, in which the competition framework evaluates submissions with this criteria. The F1 score is the harmonic mean of precision and recall. It ranges from 0 (worst) to 1 (perfect precision and recall), where the precision is the ratio between true positives and false positives, and the recall is the ratio between true positives and false negatives. In our case, a true positive happens when the model correctly classifies a patch of the image as a road. Via the F1 score criteria, the trivial constant model would have zero precision and zero recall(see figure 3), which is defined as a zero F1 Score. In most cases, the F1-score and accuracy highly depend to each other; it means an increase in the value of one leads to increase the value another one, hence besides maximizing the accuracy we should check the F1-score when we tune a model.[2]

IV. MODELS AND METHODS

In this section, we will introduce our models and methods used for this segmentation task. As described in section III, we will compare our models based on the two criteria: accuracy and f1-score. For each method, we generate more training data by rotating images or removing the isolated patch for getting a better result. In the following, we compare our models Moreover, check whether those actions cause a better result.

A. Basic Convolutional Model

For first run, we test the given model as a reference for starter. This model use a two layer convolutional neural network with soft max loss and 32 filters on first layer. This method gives us almost 61 percent accuracy and 55 percent f1 score on training test. This accuracy is not delightful, so for the next method we create our approach by generating features.

B. Special Logistic Regression

Every image contains lots of features, so the most important fact is that how we want to use these features for our purpose. For this model, we generate more than 140 features from each patch. Most of them are color-based and matrix-based, this means that we consider each patch, containing three square matrices(matrix M_r for red, M_g for green and M_b for blue), and each cell of a matrix takes a number between zero to one. In this report, we are not able to express the reason behind each picked feature, so we just explain some important ones. First, each matrix with dimension n^2 has n eigenvalues(In our case n is equal to 16), which are the same as another matrix if it is similar to it(A is similar to B if there exist an invertivble matrix P such that $B = P^{-1}AP$). By this fact, for each patch with size 16 we have 48 eigenvalues which can describe M_r, M_g and M_b. The determinant of each three matrices can vary from $1 - n$ to $n - 1$(It happens when all entries are one except for diagonal, which is zero). As a patch gets saturated with color, the entries of matrices get a value greater than zero. These criteria help us to understand how much a patch saturated with three based colors, even if some few saturated patches gives the determinant equal to zero, but the probability is near to zero. So we get determinant of each matrix for representing a characteristic of that patch.

As a road is not purely observable because of the existence of trees, cars, etc., we change the picture by decreasing the effect of specific colors. For instance, we generated a new image when the effect of green color(Most critical color for trees) decrease(see figure 2).

By generating many different patches, these able us to make new properties. We took the mean and variance of each matrix(M_r or M_g or M_r) and powers of them, so we can calculate how much a particular color approximately get involved. So if a patch belongs to a road, whether there is a car in it or covered by a tree, we built features(by minimizing a specific color) that respond to it and decrease the effect of that object, especially when it corresponds to an irregular color. Finally, we used Logistic Regression to obtain weight vector, which gives us 80 percent accuracy and 65 percent F1-score on the training set(figure 3), and also we check the method for test data and get 60 percent F1-score and 72 percent accuracy. In every method, we generated more training data by rotating both ground truth and satellite images. In this method, cropping and stretching an image cause to build some saturated patches in which the logistic regression failed.

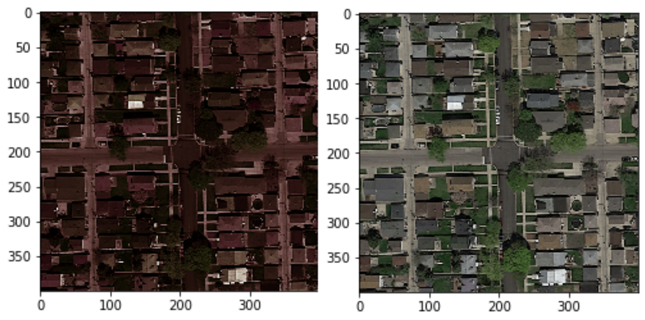

Fig. 2. Decreasing effect of green color by generating new picture

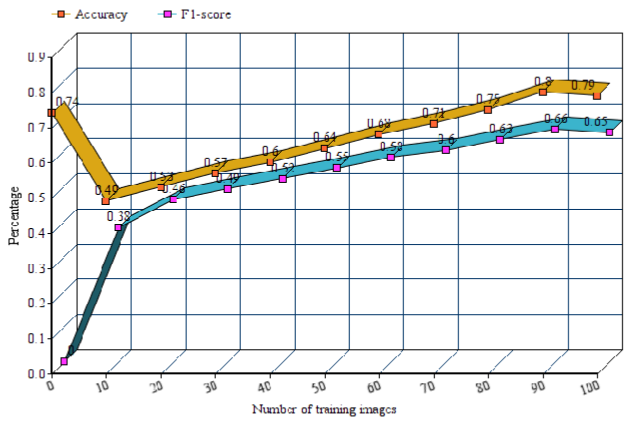

Fig. 3. F1-score and accuracy for Special Logistic Regression on different size of training set. The chart shows improvement when we test on 100 train images.

C. Combination of Logistic Regression and Basic Convolutional method

In those two methods, we observe that the main weakness relies on identifying the background(see figure 4). Since, both of them are doing well on finding the roads and both using completely different approaches, we combine both methods. We label a patch as a road if it both methods consider it as road. As a generalized way, we modified a result by omitting the isolated patches. So an isolated patch will consider background if all eight surrounding patches labeled as background. Besides this, we generate 800 training images by rotating them by a multiple of 45 degrees. As a result we got 68 percent F1-score and 83 percent accuracy on test set, which showed

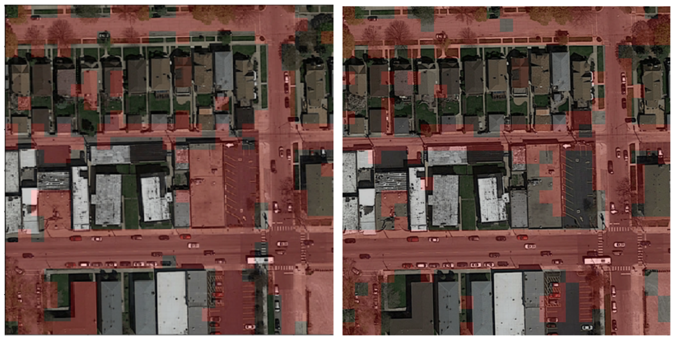

Fig. 4. Comparing result of two different methods on training set. Left image belongs to Logistic Regression and right image belongs to Basic Convolutional Model.

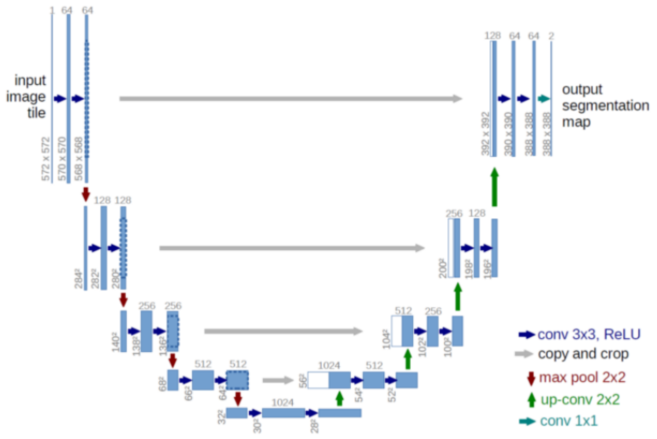

Fig. 5. The U-net architecture

combining two methods can also be helpful. We also get 70 percent F1-score and 85 percent accuracy on training set.

D. U-net Model

Since the result of previous methods were not fully satisfying, we decided to search for better methods for semantic segmentation using convolutions neural networks. One of the most useful models for image segmentation is U-Net architecture. The model consists contracting path and expansive path (figure 5). The objective of the contracting path is to capture the context of the input image, however when the image is down sampled the information about the location of the objects is slowly lost. In semantic segmentation we need to know both "what" and "where" information of the objects, hence the image has to be up sampled from low to high resolution. The U-net uses transpose convolutions (the reverse of regular convolution) to up sample the image. Figure 5 displays the u-net architecture in more detail:

The table below shows the type of different layers of the U-net that was used to train the model. The dropout value is the percentage of randomly selected neurons that are ignored during the training of the network, it is a common regularization technique used when training convolutions neural networks.[4]

The U-net takes in a 400x400x3 RGB image and outputs 400x400x1 predicted groundtruth.The output is then converted to patches for the submission file.

Layer type	Parameters
Input	Image size 400 x 400 x 3
Convolutional Block	64 (5 x 5) filters
Dropout	p = 0.25
Convolutional Block	128 (3 x 3) filters
Max Pooling	2 x 2
Dropout	p = 0.5
Convolutional Block	256 (3 x 3) filters
Max Pooling	2 x 2
Dropout	p = 0.5
Convolutional Block	512 (3 x 3) filters
Max Pooling	2 x 2
Dropout	p = 0.5
Convolutional Block	1024 (3 x 3) filters
Transpose Convolution	512 (3 x 3) filters
Concatenate	
Convolutional Block	512 (5 x 5) filters
Transpose Convolution	256 (3 x 3) filters
Concatenate	
Convolutional Block	256 (5 x 5) filters
Transpose Convolution	128 (3 x 3) filters
Concatenate	
Convolutional Block	128 (5 x 5) filters
Transpose Convolution	64 (3 x 3) filters
Concatenate	
Convolutional Block	64 (5 x 5) filters
Convolutional layer	1 (1 x 1) filter
Output	400x400x1 Grayscale image

TABLE I
FINAL CNN ARCHITECTURE WITH ALL LAYERS

1) Contraction path: The contraction path also known as the encoder consists of convolutional and max-pooling layers. The convolutional layer apply different sized filters to extract features from the image and the max-pooling selects the maximum pixel value within a 2x2 matrix to obtain the most important features and discard the less significant once. [5]

2) Expansion path: The expansion path also know as the decoder allows the network to determine the precise localization of the pixels by using transposed convolution. During the decoding the output of the down sampling layer gets concatenated with the corresponding up sampling layer. A convolutional block is then applied to this result, followed by transpose convolution to up sample the image. [5]

3) Training the model: To train the model we used the jaccard[6] distance loss as the loss function and trained it on kaggle - a platform used to train neural networks.

V. Discussion

We started with a simple tensor-flow model(Basic convolutional model) that given to us, and we got an f1 score of 0.51. We thought about the main concept of a neural network and decided to generate features ourselves by observing the matrix of each patch, and we came up with 140 features.The apply logistic regretion to these features we were able to differentiate the patches between road and not road. Although this action was not fully successful, it taught us the importance of the neural network. In the next step, we combined the results

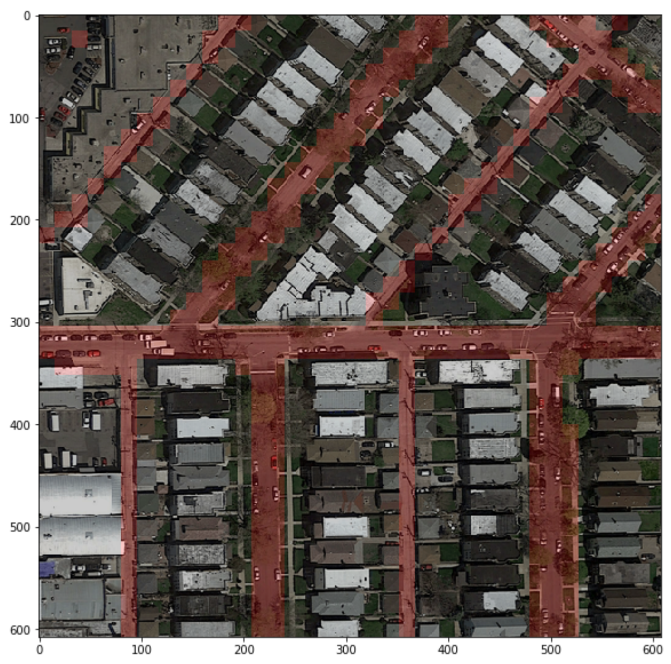

Fig. 6. Overlay of the prediction created by the U-net

of basic convolution model and the logistic regression with feature augmentation, and got a better result(see TABLE II). At last, after finding some articles and models related to image segmentation, we found out that the U-net model is perfect for our goal. We attempted to implement this model and train it on our data, but due to the computational difficulties of running this model we failed in this part. We found another code(see acknowledgment), which was similar to our U-net model, but we managed to run it and train our model on Kaggle.

Model	F1-score	Accuracy
Basic convolutional model	0.51	0.53
Special logistic regression	0.60	0.72
combine model	0.68	0.83
U-net model	0.906	0.95

TABLE II
RESULT ON TEST DATA FROM AICROWD, WITH DIFFERENT MODELS

The table [II] shows the f1-scores of the prediction on the test data from the models that we used. It can be seen that the most significant improvement in f1 score for the test data was when we used the U-net. The U-net had far more parameters than the previous model and predicted whether a pixel was road or not, rather than a patch. This allowed the predictions of the U-net to be more precise than of the other models.

VI. Conclusion

We understand that creating a model for the aim of this project is not possible without considering the neural network. So after testing U-net, we got 90.6 percent F1-score and 95 percent accuracy. The hardest part of the project lies in recognizing parking lots and trees on the roads. These were only solved by using U-net.

Acknowledgement

We want to thank you from last year team who public their work on this project for every one on Github[3]. Their names are Maxime Schoemans, Pedro de Tavora Santos, Yannick Paul Klose(Department of Computer Science, EPFL , Switzerland). We got help just for U-net model to came up our problem with it.

References

[1] https://en.wikipedia.org/wiki/U-Net
[2] K. P. Shung, "Accuracy, precision, recall or f1?" 2018, *https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9*.
[3] https://github.com/pedrotbsantos/epfl$_{road_segmentations}$
[4] https://machinelearningmastery.com/dropout-regularization-deep-learning-models-keras
[5] https://towardsdatascience.com/understanding-semantic-segmentation-with-unet-6be4f42d4b47//
[6] https://en.wikipedia.org/wiki/Jaccard$_{index}$