Machine Learning Project
Road Segmentation

Olivier Lam, Jalel Zghonda, R. Ekin Kubilay
Section of Communication Systems, *Section of Mechanical Engineering, EPFL, Switzerland*

***Abstract*—The objective of the project is to create a classifier that successfully segments roads by differentiating them from background in satellite images. Convolutional neural network with traditional and U-Net architecture is implemented and compared. Both results is found to be successful and a maximum F-1 score of 0.905 is achieved using the U-Net model.**

I. Introduction

A successful classifier is created in order to label each pixel from satellite images either road or background. Initially 100 RGB images and their corresponding ground truth images in grey scale are provided. After initial data analyses and augmentations are done, traditional convolutional neural network and U-Net architectures are built. The two models are tested and optimized using Google Colab and the results are compared.

Section II first introduces the initial data exploration and augmentation and continues with the technical details of the applied methods. Section III presents the results are provides a comparison. Section IV evaluates the results and discusses ideas for any future work.

II. Methodology

A robust solution to the problem requires the inclusion of a proper data analysis step in our methodology. This crucial step allows us to acquire an intuition about the data set. Based on the data analysis results we define the machine learning techniques that could be used.

A. Data Exploration and Analysis

The provided training data set consists of 100 satellite images with their respective ground truth images. The satellite images are 400x400 pixels RGB images and the ground truth images are of the same size. Since the latter are black and white images, they are not RGB images but gray-scale images. These properties are crucial to allow us to preprocess the images if needed before training a model with them. Fig 1 shows an example of a satellite image and its ground truth.

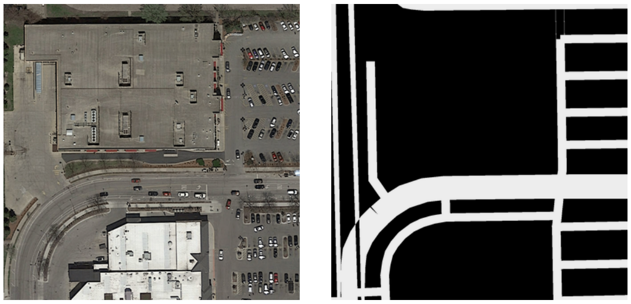

Figure 1. A satellite image on the left and its respective ground truth on the right.

As it can be seen above, predicting the road segmentation does not seem like an easy task. Roads may be hidden by some trees and some vehicles, they are not always straight or vertical/horizontal, roads in a parking area should be recognized but not parking slots and so on.

The provided testing data set differs from the training set by the size of the images. Indeed, they are bigger images of size 608x608 pixels.

B. Data Augmentation

It is evident that 100 satellite images is insufficient to efficiently train the models, so we generate more images by certain rotations and also by adding some noise, in order to have a larger sample size and cover a broader range of possible road-background configurations. Initially, each image and its corresponding ground truth is rotated by 90, 180 and 270 degrees, mirrored with respect to the axes at 90, 180, 45 and 135 degrees. Also, two more sets of images are generated to add some noises to the training data. ”Salt and pepper” noise is added by randomly changing the pixels values of certain number of pixels at each image to either white or black. Lighting condition of each image is also altered randomly at each channel. The ground truth images for the noisy satellite images are kept unchanged. With these generated images, the training set is increased by 10 times in number. Upon realizing that the models are weak in classifying roads which are at an angle, rotations of 15, 30, 45 and 60 degrees are also created. With these rotations, in order to keep the image sizes same, parts of the rotated images that are outside the limits are left out and regions where no image is present are treated as background. With these last additions, the total training data set size is increased to 1400 images. All the data augmentation is done offline.

C. Model composition

According to our research, on the different neural network models used in the domain of image segmentation, it is decided to implement two convolutional neural network models with different architectures. One traditional CNN architecture used as a baseline and one with the U-Net architecture. Having two different models allows some comparisons on their effectiveness and the capability to be made. In the following sub-sections, more details about each architecture are given.

1) Traditional Convolutional Neural Network: For what concerns the classic CNN architecture, and in order to address the per-pixel image classification problem, we use a sliding window approach where the objective is to classify the block at the center of an image, according to its context. It means that the model sees the surroundings of the patch to be able to classify it as belonging to a road or not. The window size is chosen in order take into account a context that is large enough. Since large windows are computationally expensive, we choose a size of 4.5 x patch size which corresponds to a 72x72 window.

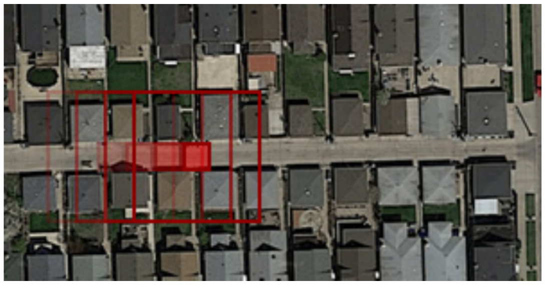

Figure 2. Sliding window approach, the small square is the classified patch and larger one is its context. The windows are spaced apart by a stride of 16 pixels.

The sliding window approach implies that the images need to be padded on the edges since the window are larger than the patch. Reasonably speaking, a padding method is chosen, the image is reflected along the boundary axis. This produces a good estimation of the missing content as shown in Fig 3.

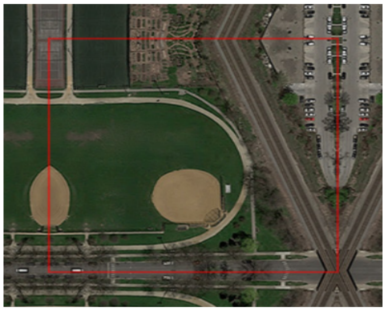

Figure 3. The red line has been added for illustrating the original boundaries of the images

In the table below, we present a summary of the architecture used in this model, taking into consideration the above procedure.

Table I
ARCHITECTURE SUMMARY OF THE TRADITIONAL CONVOLUTIONAL NEURAL NETWORK.

Layer	Characteristics
Input	72x72 RGB
Convolution + Leaky Relu	64 5x5 filters
Max Pooling	2x2
Dropout	0.25
Convolution + Leaky Relu	128 3x3 filters
Max Pooling	2x2
Dropout	0.25
Convolution + Leaky Relu	256 3x3 filters
Max Pooling	2x2
Dropout	0.25
Fully connected + Leaky Relu	128 nodes
Dropout	0.5
Output	2 labels

Finally, in the training process, we use a random batch generator that runs in parallel on a different thread and that extracts randomly a set of windows from the training images and their corresponding patches from the ground truth set. The black and white patches are then mapped to labels with thresholding on the average number of white pixels. We used a threshold value of 0.25 which is tested for both models and turns out to be optimal.

2) U-Net: Convolutional neural networks are often used to classify an image as a whole. Hence, for our task, classifying image patches can be a good idea but we might loose some informations. The U-Net architecture allows the classification for each pixel of an image, which corresponds strongly to the road segmentation problem.

The main idea of the U-Net architecture is to do a traditional convolutional neural network with a high number of channels and from the deepest layer, retrieve the information gained for each pixel. Since the architecture is composed of two phases, first down-sample the network and then up-sample it, it has a shape of an U as shown in Fig 4.

Fig 4 is the U-Net proposed by Ronneberger et al. [1] and our model is heavily based on that. To simplify it, a simple padding is added after each convolution step so the resulting image size remains the same and we don't have to crop any image either to be able to concatenate with other images. The architecture is the following: from an input image of size 400x400 pixels, we down-sample it four times where

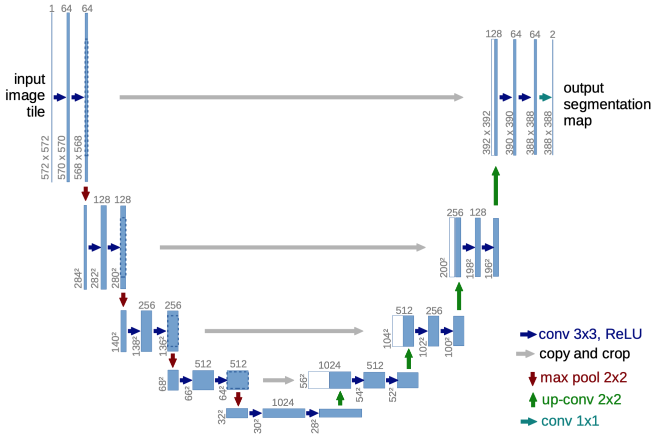

Figure 4. The architecture of the U-Net neural network.

at each down-sampling step, see Table II, we double the number of feature channels. Then, we up-sample four times where at each up-sampling step, see Table II, we divide by two the number of feature channels. A final layer with a 1x1 convolution is added to the network to give a 400x400 pixels image with 1 feature channel, followed by a sigmoid activation function which will produce the final output.

Table II
DETAILED DOWN-SAMPLING AND UP-SAMPLING STEPS.

Down-Sampling Step	Up-Sampling Step
• Convolution block • 2x2 max pooling with stride 2 • Dropout	• Convolution block • 3x3 up-convolution that halves the feature channels • Dropout
Convolution Block	
• 3x3 convolution that doubles the feature channels • Activation function • 3x3 convolution that keeps the same number of feature channels • Activation function	

The parameters that will improve the model are the following:

- *Activation function*: It defines the output of a node given an input set. The most popular ones are ReLu and LeakyReLu, which are defined by $(x)_+ = \max\{0, x\}$ and $f(x) = \max\{\alpha x, x\}$, respectively.
- *Batch size and initial number of filter*: We believe that having an initial number of 64 filters and a batch size of 64 will lead to the best result ([1]) but having a limit on the computational power, tuning these parameters is necessary.
- *Dropout rate*: One of the most popular way to regularize a convolutional neural network and to avoid our model to overfit.

Table III
RESULTS OF OUR MODELS

Model	Activation Function	Dropout Rate	Training F1-Score	Testing F1-Score
U-Net	ReLu	0.5	0.947	0.882
U-Net	LeakyRelu ($\alpha = 0.3$)	0.5	0.943	0.888
U-Net (more data)	LeakyRelu ($\alpha = 0.3$)	0.25	0.931	0.897
U-Net (more data)	LeakyRelu ($\alpha = 0.3$)	0.5	0.941	0.905
Traditional	LeakyRelu ($\alpha = 0.1$)	0.25 & 0.5 (dense)	0.954	0.887
Traditional	LeakyRelu ($\alpha = 0.001$)	0.25 & 0.5 (dense)	0.958	0.880

D. Predictions

1) Predicting with Convolutional Neural Network: The prediction process follows the same procedure than in the training, we split a test image into windows and predict the labels of each window with the model. Then we map the labels to their corresponding patches to obtain a black and white image.

2) Predicting with U-Net: Since the test images are of size 608x608 pixels, it is not possible to feed our U-Net model with them since it takes 400x400 pixels images as input. Therefore, we take four 400x400 pixels images for each test image, predict them and merge them together to have the final 608x608 ground truth prediction.

III. RESULTS

The final results of our models are presented in Table III.

With the traditional convolutional neural network, due to computational complexity of the model, we only test the following LeakyRelu hyperparameter $\alpha \in \{0.1, 0.001\}$

With the U-Net architecture, the LeakyRelu activation performs better than the Relu and the optimal α is 0.3. The alphas tested are $\alpha \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ and the values tested for the dropout rate are $\{0.25, 0.5, 0.75\}$. A batch size of 16 with initially 32 filters are used and for the model with the best hyper-parameters, we try to train with a batch size of

32 with initially 16 filters but the testing F1-Score is lower. It was not possible to train with bigger batch size and higher number of initial filters, due to the memory limit of the GPU used with Google Colab.

After a first work on the data augmentation, the maximum value of the testing F1-Score is 0.888. To have an idea on what can be done to improve this result, we look into our prediction images and we can observe that all the predictions for diagonal roads are very poor. The reason of that is because we rotate the images by $k * 90°$ and a majority of the training images are composed of vertical and horizontal roads. Adding training images with a rotation of 15°, 30°, 45° and 60° yields much better predictions, as shown in Fig 5.

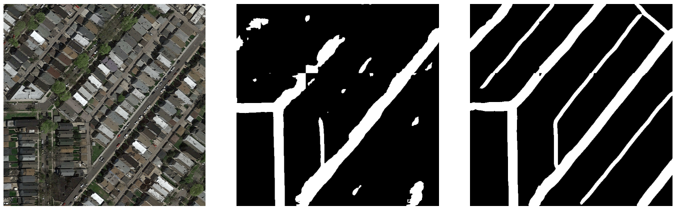

Figure 5. Improving predictions with U-Net. From left to right: a satellite image, the first prediction with data augmentation, the final prediction with better data augmentation.

In Fig 6, we compare our predictions, generated by the traditional and the U-Net neural network. Since the difference in the testing F1-Score is 0.018, we can see that the predictions are quite similar, at least for the main roads and indeed, the U-Net performs a little better.

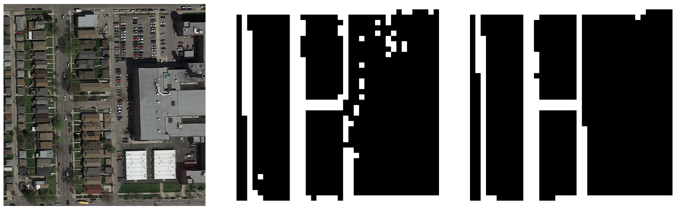

Figure 6. Prediction comparison of our two models. From left to right: a satellite image, the prediction of the traditional neural network and the prediction of U-Net with 16x16 pixels labelled patches

IV. Discussion

The results obtained are quite good since the final F1-Score on the testing images is higher than 0.9. There are still some weak areas in our both models. Predicting the roads of a parking slot and the roads where the are some shadows seems to be an issue for both of our models. The next step would be to improve the data augmentation by focusing on the lighting changes of the images and generate new images from the images that contain some parking slots. Having that amount of images would force us to manage the memory allocation in a better way in order to be able to train the models. Another improvement could be to process the prediction images of the U-Net by applying some filters in order to have ”smoother” predictions. Indeed, we are predicting four images for 1 testing image and merge them together. Even if we pay attention to the overlapping pixels, we can clearly observe some small shifts in the final prediction. Since we need to label 16x16 pixels patches in order to submit, correcting some pixels can have an impact on the final result.

V. Summary

In order to achieve the aim of creating a model that successfully classifies roads from satellite images, careful data analyses and augmentation is done on the provided 100 images and their corresponding ground truths. To ensure a robust method, two methods are implemented using a traditional convolutional neural network and using a U-Net architecture. Several parameters for each method are tested in order to observe the response of the models and achieve the best prediction possible. F1-score above 88% is obtained frequently on testing data and a maximum of 90.5% is reached for U-net with improved data augmentation, 0.5 dropout rate and LeakyReLu activation function with $\alpha = 0.3$. Through this process it was observed that better data augmentation significantly improved the model even though it increased memory requirements, so better performance is possible by further improving data augmentation if better memory allocation is done. Further improvements are thought to be possible with a better post-processing scheme.

References

[1] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. Germany, 2015. `https://arxiv.org/pdf/1505.04597.pdf`