Road Segmentation from Aerial Images

Francesco Saverio Varini
Department of Computer Science,
ETH Zurich

Robin Bader
Department of Computer Science,
ETH Zurich

Jakob Beckmann
Department of Computer Science,
ETH Zurich

***Abstract*—This project deals with semantic segmentation of aerial images where a semantic labeling of either road or non-road is assigned to the patches on the images. Recently, *deep convolutional neural networks* (CNNs) have shown impressive performance for this task. Our method uses an adapted version of the U-Net [1] architecture which enables to predict on the full pixel images patch-wise and, additionally, it is proposed to use augmentation and dropout layers to overcome the sparsity of the available training data. The experiments conducted show that this model outperforms some other tested techniques.**

I. Introduction

Understanding an image and extracting its information is an important area of application in computer vision. In particular, image segmentation is the process of labeling parts of images according to the given criteria. This paper focuses on the automatic segmentation of aerial images into *road* or *non-road* which has several application due to numerous real-world related issues ranging from civil infrastructure [2] to updating geographical information systems [3].

The most popular tool used for this task is based on supervised machine learning. However, the recent increase in computing resources terms has led to the development of new techniques such as *deep convolutional neural networks* (CNN) which are currently top-performers for image segmentation. This particular architecture has found a very broad attention in the previous researches of road segmentation [4], [5]. Nevertheless, this application focused on a prediction of per-pixel segmentations or multiple-class segmentation, whereas in the following novel proposed architecture, it is examined the prediction on image patches with respect to exactly two classes.

It has been adopted the *U-Net* [1] architecture which is a variant of the *Fully Convolutional Network* (FCN) [6] which predicts pixel-to-pixel segmentations. The main additional contribution introduced in this work is an adapted architecture of the U-Net to allow a pixel-to-patch segmentation. Further, the new proposed architecture makes use of different activation functions as well as a dropout layer to prevent overfitting, while overcoming the sparsity of the data induced by image augmentation.

Different baseline algorithms were used for the sake of comparison with the main approach followed. These baseline architectures specifically comprehend the standard U-Net implementation and the patch-to-patch implementation approach presented in [7].

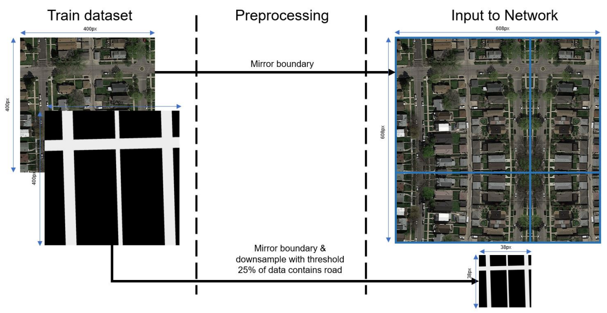

Figure 1. Preprocessing necessary for the training data. The given dataset (left) with dimensions 400x400 along with its pixelwise labeling are preprocessed (middle) using a mirror effect which enlarges the images to a resolution of 608x608 (right). Furthermore, the labels are downsampled to a 38x38 for each 16x16 image patch.

II. Models and Methods

The contribution of this paper is a CNN architecture that takes as input an image and outputs the prediction on whether the image-patches contains roads or not, given a certain threshold of 25%. This section attempts to describe the given datasets, the pre-processing techniques used and the different model's architectures chosen.

A. Dataset & Preprocessing

The dataset has been provided by the Kaggle competition held by the ETH Computational Intelligence Lab course 2018 [8]. This dataset contains 100 training samples which consist of an input image and its pixel-wise labeling both with a resolution of 400x400 pixels. The competition's goal is to predict on 16x16 sized patches within input images of resolution 608x608, where every patch is labeled as 1 if more than 25% of the patch's area is recognized as road and labeled 0 otherwise.

The training dataset is preprocessed due to the fact that the images contained are of different dimensions than the one in the test set. Figure 1 shows the implemented preprocessing which extends the images using a mirror boundary condition, i.e. the image is reflected along the boundary axis (See Figure 1 in its right part).

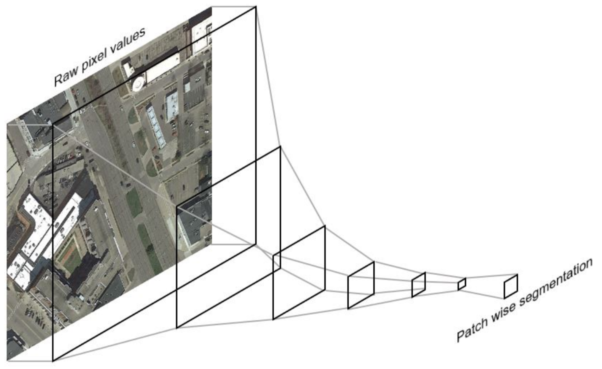

Figure 2. Architecture of the proposed network. Visualizing the 7 layers used where the first 6 downsample the image and the 7th layer upsamples it again using a residual connection to layer 5 and outputs a patch-wise segmentation.

B. Image Augmentation

To overcome the limitation of the small number of training samples, some data augmentation technicals have been used. The effectiveness of data augmentation was previously demonstrated [9] for image classification. Because the roads of many of the provided training images are axis aligned it has been decided to implement data augmentation that compared to other previous approaches, arbitrarily rotates the images according to multiples of 90 degree [7]. The disadvantage of our approach is that the generation of these arbitrary rotations takes considerably more time. To maintain the advantage of the U-Net architecture about its fast training time [1], the augmentation is split into two phases which are executed on different moments during the training. The first part of augmentation is only executed once every epoch, while the second part of it is applied to each training sample on each batch.

The first phase of augmentation includes steps as randomly zooming (range 0.8-1.2), shearing (range 0 - 0.1), rotation (0 – 360) and height as well as width shifting (range 0 - 0.1) on the images. Finally, before generating each batch as input to the model, it is additionally applied a multiple of a 90-degree rotation as well as a random mirroring. This second augmentation phase can be executed very fast during training time, while the first one needs more time. This technique allowed to overcome the disadvantages of the slow arbitrary rotation while still generating a higher number of different training samples.

C. CNN Architecture

In order to achieve the desired pixel-to-patch prediction of the model, a truncated version of the U-Net [1] as presented in Table I has been adopted. This architecture can be further visualized in Figure 2. The original implementation of the U-Net uses the up-sampling layer (e.g. layer 7) as many times as the down-sampling layers (e.g. layer 1 – 4) to output the same dimensions as the input. In this project, this isn't desired, and the model has been truncated to fit the newer needs where the dimension 38x38 corresponds to the desired patch size $16 \cdot 38 = 608$.

To further limit the arising of overfitting on the training data, a supplementary adapted version of regularization is proposed as shown in Table II. The adaptations were inspired by [7]. In particular, instead of using *rectified linear units* (ReLU) as activation function, like it is proposed by the U-Net implementation [1], it is used the *Leaky ReLU* to avoid the *dead filter effect* which some units can experience. *Leaky ReLUs* are defined as $f(x) = \max(\alpha x, x)$. After different trials, it has been found empirically that the value $\alpha = 0.1$ yields good results. This alpha setting is confirmed to be a good one also in other methods [7]. Furthermore, in order to regularize the model, after each level it has been added a dropout layer with a dropout rate of 0.25. The output layer is penalized using a L2-regularizer.

Level	Layer	Dimension
0	input	608x608x3
1	2 x conv(3x3, ReLU) maxpool(2x2)	304x304x32
2	2 x conv(3x3, ReLU) maxpool(2x2)	152x152x64
3	2 x conv(3x3, ReLU) maxpool(2x2)	76x76x128
4	2 x conv(3x3, ReLU) maxpool(2x2)	38x38x256
5	2 x conv(3x3, ReLU)	19x19x512
6	up-conv(2, 2) 2 x conv(3x3, ReLU)	38x38x256
7	conv(1x1, Sigmoid)	38x38x1

Table I
U-NET, TRUNCATED FOR PIXEL TO PATCH PREDICTION.

Level	Layer	Dimension
0	input	608x608x3

1	2 x conv(3x3, Leaky ReLU)
	maxpool(2x2)
	dropout(0.25)
		304x304x32

2	2 x conv(3x3, Leaky ReLU)
	maxpool(2x2)
	dropout(0.25)
		152x152x64

3	2 x conv(3x3, Leaky ReLU)
	maxpool(2x2)
	dropout(0.25)
		76x76x128

4	2 x conv(3x3, Leaky ReLU)
	maxpool(2x2)
	dropout(0.25)
		38x38x256

5	2 x conv(3x3, Leaky ReLU) dropout(0.25)	19x19x512
6	up-conv(2, 2) 2 x conv(3x3, Leaky ReLU)	38x38x256

7	conv(1x1, Sigmoid)
	L2(10E-6)
		38x38x1

Table II
U-NET, TRUNCATED, DROPOUT & LEAKYRELU FOR PIXEL TO PATCH PREDICTION.

Model	Train	Val	Test
All background	-	-	0.85
CNN [7]	-	-	0.87
U-Net, truncated	0.96	0.85	0.90
U-Net, truncated, dropout	0.94	0.86	0.89

Table III
COMPARISON OF DIFFERENT METHODS WHERE EACH COLUMN SHOWS THE SCORES OBTAINED DURING TRAINING, VALIDATION AND TEST.

D. *Loss function*

The final score on the Kaggle competition [8] is calculated using the F1-Score which was used for the training the model as well.

III. RESULTS

Here it is compared the two proposed methods from section II-C with two baselines which have been chosen to be the all-zero prediction and the CNN model advanced in [7] which does patch-to-patch prediction. According to our knowledge, the latter was the highest performing architecture implementation for this problem.
The neural networks are all implemented using Keras with Tensorflow as backend. All the models are evaluated using a train/validation split of 2/3 of the training dataset.

The results are presented in Table III. The train and validation scores displayed are the one obtained at the end of the training. The test-score was found based on the results of the Kaggle competition [8]. For the CNN model proposed in [7] and all zero prediction, there are no train and validation scores available. Based on the test set it is possible to conclude that the proposed methods based on the re-adapted U-Net architecture outperform the patch-wise trained and predicting CNN model. Comparing the models presented in section II-C, it turns out that the expected result for the additional regularization with dropout lead to a closer training/validation difference and finally to a better score on the test set.

Interesting is the fact that the all background baseline where each patch is predicted to be non-road obtains an already high test result.

An example of prediction using U-Net truncated with dropout & LeakyRelu, which is the one performing best, is shown in Figure 3. The last submission to the Kaggle competition was trained with all the provided training data and no validation-set. The obtained score for the Kaggle competition is 0.90242.

IV. DISCUSSION

Even for a human, it is not always possible to correctly classify a patch into road or non-road. This shows the difficulty of obtaining a perfect predictor for this task. Visually analyzing the prediction in Figure 3, the predictions of the network closely resemble the expected result by a human. Examining the top horizontal street some road prediction

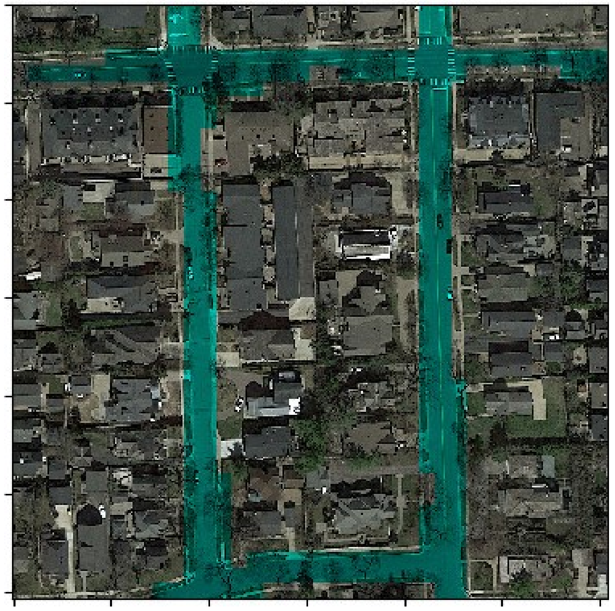

Figure 3. Example prediction using U-Net, truncated, dropout & LeakyRelu where the predicted roads are marked as blue.

discontinuities are evident. A proposal for additional work could be to either find a way to penalize discontinuities or, alternatively, correct them in a post-processing step.

To further improve the result we propose to train the model with more data. In [4] it was shown that training on datasets from other data sources of aerial images can improve the prediction. Additional training data for this task is indeed available (e.g. Google Maps, OpenStreetMap). Therefore, additional work could be done in obtaining and preprocessing the new data in order to be advantageous for better predictions.

V. SUMMARY

We proposed an image-to-patch prediction using the U-Net architecture which is based on a CNN model. This enabled us to predict if a given patch of an aerial image is occupied by a certain threshold area of road or not. To permit the U-Net architecture to predict on images-to-patches it was necessary to truncate the end of the original network architecture and to directly predict to the 38x38 output labels. The final proposed model adds dropout layers to counteract overfitting. Some attempts to overcome the problem of a small training dataset by using image augmentation were performed. A comparison of our method to the two baselines suggests that the proposed model surpass them.

REFERENCES

[1] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," may 2015. [Online]. Available: http://arxiv.org/abs/1505.04597

[2] S. C. Radopoulou and I. Brilakis, "Improving Road Asset Condition Monitoring," *Transportation Research Procedia*, vol. 14, no. 0, pp. 3004–3012, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.trpro.2016.05.436

[3] J. F. Girres and G. Touya, "Quality Assessment of the French OpenStreetMap Dataset," *Transactions in GIS*, vol. 14, no. 4, pp. 435–459, 2010.

[4] P. Kaiser, J. D. Wegner, A. Lucchi, M. Jaggi, T. Hofmann, and K. Schindler, "Learning Aerial Image Segmentation from Online Maps," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 11, pp. 6054–6068, 2017.

[5] S. Saito and Y. Aoki, "Building and road detection from large aerial imagery," E. Y. Lam and K. S. Niel, Eds., vol. 9405. International Society for Optics and Photonics, feb 2015, p. 94050K. [Online]. Available: http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.2083273

[6] J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," nov 2014. [Online]. Available: http://arxiv.org/abs/1411.4038

[7] D. Pavllo, M. Martinelli, and C. Hwang, "EPFL Machine Learning Project 2 Report Road Segmentation Team : The Overfitters," 2017.

[8] "Kaggle - CIL Road Segmentation 2018," 2018. [Online]. Available: https://www.kaggle.com/c/cil-road-segmentation-2018

[9] J. Wang and L. Perez, "The Effectiveness of Data Augmentation in Image Classification using Deep Learning." [Online]. Available: http://cs231n.stanford.edu/reports/2017/pdfs/300.pdf