CIL: Road Segmentation

Igor Pesic
Department of Computer Science,
ETH Zurich

Felipe Sulser
Department of Computer Science,
ETH Zurich

Minesh Patel
Department of Computer Science,
ETH Zurich

***Abstract*—Image segmentation of the aerial road images has become a significant part of the research in the recent years. The use cases are numerous and we are going to focus on the simpler, but still very useful part, namely just the road segmentation. Our task is to classify the pixels as either road or background. To achieve our goal we have combined techniques proposed in several other papers and have adjusted these to suite our requirements and resources best as possible. The main part of our solution is a convolution neural network (CNN). Beside it, we apply the techniques for data augmentation, feature selection and post-processing. The results we obtained with our solution are close to the state-of-the-art solutions proposed in the other papers, but very often, our model is much simpler.**

I. Introduction

The goal of this work is to segment out the road on the aerial images. The problem is to decide what pixels represent the road and what pixels are not the road. Even though we would ideally like to have a pixel-wise granularity, we are proposing the approach which achieves the patch-wise granularity which is many cases good enough, especially in this project. This also simplifies the problem substantially since each patch has only value 0 or 1.

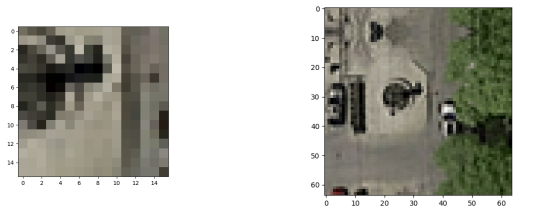
For the CNN, we have implemented the solution which is combination of multiple similar other works on this topic and we have managed to achieve very similar results. The focus of our work was the design of the CNN but we also invested a substantial amount of effort into denoising in order to improve the results obtained by CNN. We would also like to mention that because of constrained resources we had to simplify some parts of our model.

This report has the following structure: in Section II we explain all the steps we do before training and evaluating the model, in Section III we describe our model, both CNN and denoising that comes after that. Finally in IV we discuss the results achieved with our model.

II. Data

A. Data Augmentation

The original data set consisted of 100 labeled aerial images of road maps. As our approach consists of a deep neural network, we consider that the given data is not enough and thus we try to augment it to expand it further. For data augmentation, we first rotate all the images by 90 degrees and to mirror them. Beside the obvious increase in the training data size, this approach also makes our model more robust. Furthermore we have noticed that the predictions on the diagonal roads were much worse than on the horizontal and vertical ones. This appears to be due to the lack of diagonal roads in our original data set. In order to fix this, we pick 9 images with diagonal roads and highways which we rotate again by 180 and 270 degrees. Finally, we have noticed some inconsistencies in labeling of couple of images and we have decided to discard those from our training data. At the end, our data set has 309 images in total.



(a) Original patch

(b) Context added patch

Figure 1: Different approaches

B. Feature extraction

In the baseline model we split the image in patches of size 16×16 pixels. This provided granularity that was good enough, but it lacked the information about the surroundings. In order to address that we have come up with solution that we called *added context*. Later, we have found the same approach in [2], [3] and [**?**]. The approach is as following: split the image in 16×16 patches, then add the surroundings area to it so that patches have the total size of 64×64 pixels. The difference between patches is shown in 1. Labels are based solely on the original 16×16 patch.

We have decide for this patch size empirically by trying out different context-added sizes and 64 turned out to bring the best score on the Kaggle competition[1].

C. Class balancing

As our data was very unbalanced i.e. the number of background patches was $\times 3$ bigger than the number of

[1] inclass.kaggle.com/c/cil-road-segmentation-2017

road patches, we had to balance out our data set so that both classes have similar number of representatives. We have done that by equalizing the number of patches in both classes, i.e. by randomly picking a limited number of background patches.

III. Models and Methodsg

A. Baseline Model

The baseline model works on batches of 16×16 patches. Its configuration is as follows: $IN(3, 16 \times 16) - C(32, 5 \times 5/1) - MP(2/2) - C(64, 5 \times 5/1) - MP(2/2) - FC(512) - FC(2)$ Where:

- $IN(a, b \times c)$ – input image of a channels and size $b \times c$
- $CONV(a, b \times c/d)$ – Convolution layer with depth a, window size $b \times c$ and stride d
- $FC(a)$ – Fully Connected layer of size a
- $MP(a/b)$ – Max Pooling layer of size a with stride b

B. Improved CNN Architecture

The core of our work is the CNN design. We have got the main idea for the architecture of the network from Alina Elena [2]. In this design, they connect both a VGG network [**?**] and an AlexNet network [**?**] to create a dual stream network that takes as input the local patch to be predicted and the whole image as context. Our solution takes this network as inspiration, however it only takes the local information as input. This descision was taken due to limited data and also because empirically, the results obtained with it were good.

Our CNN network can be described as follows: $IN(3, 64 \times 64) - C(64, 3 \times 3/2) - MP(2/2) - C(128, 3 \times 3/2) - MP(2/2) - C(256, 3 \times 3/2) - MP(2/2) - C(512, 3 \times 3/2) - MP(2/2) - FC(2048) - FC(2048) - FC(2)$

Additionally all convolutional layers are followed by rectified linear units ($RELU$) layer. The output of the CNN is the softmax function for the two classes and we use the Adam optimizer [4].

C. Error function

Our CNN model was reducing the log-loss, but for the evaluation we have used two other loss metrics. The first one was the classification error and it was used on the validation set that helped us find the right number of training epochs. Further discussion on that will follow in the next section. The second one was the F-1 score to calculate the test error. It was given by the Kaggle competition.

D. Model hyper-parameters

Since we used the Adam optimizer, there were not many parameters to tune. One parameter was the batch size, which we have not changed from the baseline model since in the one of previous exercises we have learned the it should nether be too big nor too small, so we found 32 to be a good choice. The only other parameter to tune was the number of training epochs. This one we have empirically chosen to be (TODO: state the number) based on training the model on 90 images and validating it in every epoch on the other 10 images. The Figure (TODO: ref the plot of validation error) shows how the validation error changes with the number of epochs. Beside these, there were no further hyper-parameters to tune.

E. Post-processing

TODO

IV. Results

TODO Organize the results section based on the sequence of table and figures you include. Prepare the tables and figures as soon as all the data are analyzed and arrange them in the sequence that best presents your findings in a logical way. A good strategy is to note, on a draft of each table or figure, the one or two key results you want to address in the text portion of the results. The information from the figures is summarized in Table I.

When reporting computational or measurement results, always report the mean (average value) along with a measure of variability (standard deviation(s) or standard error of the mean).

V. Summary

TODO The aim of a scientific paper is to convey the idea or discovery of the researcher to the minds of the readers. The associated software package provides the relevant details, which are often only briefly explained in the paper, such that the research can be reproduced. To write good papers, identify your key idea, make your contributions explicit, and use examples and illustrations to describe the problems and solutions.

References

[1] S. Saito and Y. Aoki, *Building and road detection from large aerial imagery*. SPIE, 2015, vol. 9405.

[2] A. E. Marcu, "A local - global approach to semantic segmentation in aerial images," 2016, master Thesis.

[3] V. Mnih and G. E. Hinton, *Learning to Detect Roads in High-Resolution Aerial Images*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 210–223. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-15567-3_16

[4] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

Basis	Support	Suitable signals	Unsuitable signals
Fourier	global	sine like	localized
wavelet	local	localized	sine like

Table I: Characteristics of Fourier and wavelet basis.