Machine Learning - Project
Road Segmentation

Marion Chabrier, Valentin Margraf, Octavianus Sinaga
Department of Computer Science, EPFL Lausanne, Switzerland

***Abstract*—The goal of this project is to implement and train a Neural Network that is able to segment satellite images into roads and background. We implement a Convolutional Neural Network and use the sliding window aAproach in order to tackle this classification task. We got a F1-score of 0.881 as the best result.TO**

I. Introduction

Image segmentation is a well known task in the field of Deep Learning which can be handled by Neural Networks. There exist different approaches such as Convolutional Neural Networks ([1]) or newer ones such as the U-Net ([2]) which both yield good results according to our research. We decided to choose the Convolutional Neural Network, since it seemed very natural to us to classify patces of the image by taken into account their context inside the image.

We start explaining our methods, i.e. description of the classification task and doing data augmentation. Then we present the network structure, the training process as well as tuning of the hyperparameters. With this trained network we predict on the test data and then present our results obtained with the CNN. We conclude with a discussion and summarize our results.

II. Methods

A. Classification Task

Our Neural Net takes as input parts of the image of size 72x72, so called windows, and classifies if the 16x16 patch in the center of this window belongs to "road" or not. If the net classifies the patch as road, the output should be [0,1].

This corresponds to the label 1 and hence will be white in the predicted label. Otherwise it should be [1,0] and the patch in the label will be black. This is illustrated in Figure 1.

The idea behind this approach is, that the patch in the center is classified by taken into consideration the surrounding of this patch. Intuitively speaken, if we have a lot of vehicles near or in the patch, it is more likely that the patch belongs to road. Conversely, if there are buildings everywhere around the patch, the patch also belongs to background with high probability. Each patch hence is classified by its context in the image.

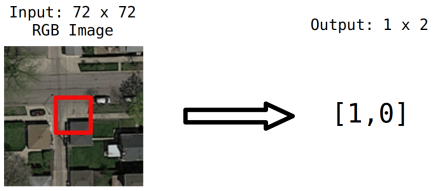
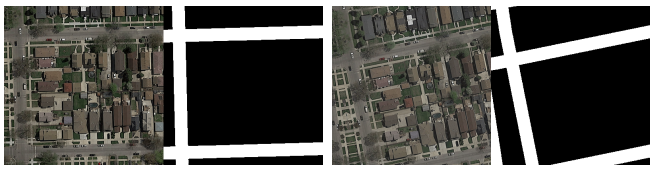


Figure 1: Our CNN classifies if the patch in the center of the window belongs to road or to background.

B. Data Augmentation

The training data consists of 100 RGB satellite images of size 400x400 and their groundtruths, 400x400 binary images respectively. Obviously this classification task described above is not easy to handle. Sometimes roads are hidden by a tree, or s may drive on them and many other difficulties could arise. We therefore augment our training data in order to achieve better results. We rotate each image and its groundtruth by 15, 30, 45, 60, 90, 100, 180 and 270 degrees. We reflect the images along their boundary axes. Then we crop out images of size 400x400 to get training images and groundtruths in the same size as the 'original' training data. This is illustrated in Figure 2. + SALT PEPPER NOISE?



(a) 'Original' training image and label.

(b) Rotated training image and label.

Figure 2: Example of augmenting data: Rotated by 15 degrees, mirrored along the boundary axes and cropped out 400 x 400 pixels.

By doing so we augment the amount of training data by a factor of 9.

C. Convolutional Neural Net

With aim of classifying the whole image and not just one patch, we make use of thege. This so called 'sliding window'h that is illustrated in Figure 3.

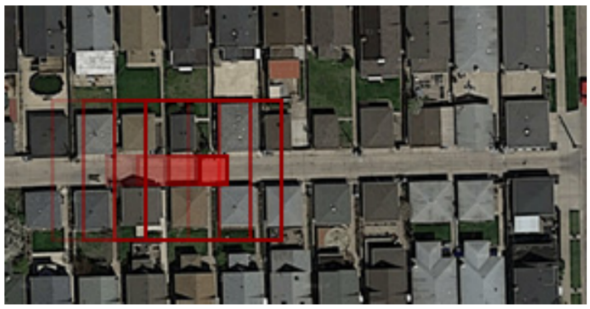


Figure 3: For each window the patch in the center will be classified. We then move the window through the image by a certain stride.

In order to apply this technique to patches near the border, we have to enlarge the images. We therefore choose a padding method: At the border we reflect each image along its boundary axes. Then we move the window through the whole image. For each window we classify the label of the patch in the center.

The architecture of our first Convolutional Neural Net is displayed in Table 1. We implemented three convolutional layers with increasing depth. After each of them we apply the Leaky Relu as activation function and use Max Pooling and Dropout layers in order to fight overfitting and make our model more robust.

Layer	**Characteristics**
Input	72x72x3 RGB
Conv + LReLU	64 filter each 5x5
Max Pooling	2x2
Dropout	0.25
Conv + LReLU	128 filter each 3x3
Max Pool	2x2
Dropout	0.25
Conv + LReLU	256 filter each 3x3
Max Pool	2x2
Dropout	0.25
Dense + LReLU	128 nodes
Dropout	0.5
Softmax + Output	2 nodes

Table I: Architecture of the Convolutional Neural Network.

We also implemented another quite similar CNN, it consists of the same structure of layers. We feed in bigger windows of size 128x128. This approach allows to consider even a bigger context of the patch in the center of the window and hopefully makes it easier for the CNN to classify the patch. Due to the computational increase that will arise, we decrease the depth of the convolutional layers. We also use bigger filters of size 10x10.

Layer	**Characteristics**
Input	128x128x3 RGB
Conv + LReLU	10 filter each 10x10
Max Pooling	2x2
Dropout	0.25
Conv + LReLU	10 filter each 10x10
Max Pool	2x2
Dropout	0.25
Conv + LReLU	10 filter each 10x10
Max Pool	2x2
Dropout	0.25
Dense + LReLU	128 nodes
Dropout	0.5
Softmax + Output	2 nodes

Table II: Architecture of the Convolutional Neural Network 2.

D. Training

In the training process we randomly choose a window of an image of the training data. We then compute the corresponding patch of the groundtruth image. The groundtruth label is then given by the mean of all pixels in this patch: If it is higher than a certain threshold, 0.25 in our case, then the label will be 1, otherwise 0.
By doing so we get even much more training data since each window of each image actually corresponds to one training image.
We tuned the following hyperparameters: the parameter α for the Leaky Relu Function $f(x) = max(x, \alpha x)$, the dropout probability, window size, patch size.

E. Prediction

The test image data consists of 50 RGB satellite images of size 608x608. We first have to do some data preparation: We enlarge the images by a padding method similar to the one explained in section II-C. After cropping the image into windows of size 72x72, we feed each window into our neural network and predict a label for the patch in the center of this window. We get our final predictions by mapping each label back to its corresponding patch and setting these patches together to an image. We use a stride that has the same size as our patch.

III. Results

For every test image we obtain a prediction, which segments roads from background. Two test images and their predictions are displayed below.

Figure 4: Test image 2 and its prediction.

Figure 5: Test image 9 and its prediction.

In general the roads are well segmented from the background. In Figure 4 the network was able to recognize the roads, although on the right they were hidden by some trees. It also did not misclassify the parking lot as road. The small part of road on the bottom left was also correctly classified, unfortunately not that smoothly due to the patch size. In Figure 5 one can see some misclassified patches on the highway, although the rails and the parking lot were correcty labelled as background. Also in this case all roads were well detected in general. The CNN performs quite good
.
As one can observe, a window size of 72x72 and patch size of 16x16 doesn't give very smooth predictions. This is obviously caused by the fact, that we predict one label for a whole patch. If one chooses smaller patches, one will get smoother results. This can be seen in Figure 6. In the smoother case we chose a window size of 16x16 and patch size 4x4.

Figure 6: Test image and its predictions, patch size 16 and 4 respectively.

AICrowd expects labels for patches of size 16x16. Therefore in the latter case we computed the mean over 16 smaller patches of size 4x4 to compute the label for the 16x16 patch in which the smaller patches are lying inside. Unfortunately this did not give us better results. The final results for both Convolutional Neural Networks are displayed in Table 3. The CNN with 72x72 window size performs best achieving F1-Score of 0.882. The CNN 2 also performs quite good, the F1-Score is 0.862. q

Model	**F1-Score**
CNN, 16x16 window, 4x4 patch	0.741
CNN, 72x72 window, 16x16 patch	0.882
CNN 2, 128x128 window, 16x16 patch	0.862
U-Net	0.894

Table III: Results.

IV. Discussion

V. Summary

Acknowledgements

References

[1] S. Bittel, V. Kaiser, M. Teichmann, and M. Thoma, "Pixel-wise segmentation of street with neural networks," 2015, https://arxiv.org/pdf/1511.00513.

[2] O. Ronneborger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015, https://arxiv.org/abs/1505.04597.