Project 2 - Road Segmentation

Raphaël Mirallié, Luca Nyckees and Clarisse Schumer

Abstract—**In this project, we propose methods that train a classifier having for purpose road segmentation, more precisely, our goal is to distinguish what is road and what is not in satellite images. In the past, convolutional neural networks have proven to be very promising in this context. We investigate the case of a convolutional neural network using depthwise separable convolutions and briefly introduce a U-net structure as comparison criteria.**

I. Introduction

The concept of road segmentation embeds in the more general notion of image segmentation, having for general goal to assign a label to every pixel of a given image format. This process have recently proven very useful in many computer vision related areas, such as medical image processing and machine vision. Our project focuses on road segmentation, but similar techniques can be used to identify any type of object or instance in an image.

In this report, we propose a way of building a model appropriate for the task mentioned above. It is based on the notion of *convolutional neural network* (CNN). CNNs constitute a subclass of artificial neural networks, having two main interesting characteristics. (1) CNNs have the ability to capture information in a local way - in the present context, this means taking into account neighbouring pixels when assigning a label to a given pixel. This revolves around the idea of spatial significance within the data : distance matters. (2) CNNs drastically improve on the overall complexity of operations (e.g. training goes faster, the model needs fewer samples to train well).

The specific task we try to accomplish here is to determine wether a given pixel in a satellite image is part of a road or not. In the next sections, we review different approaches, to perform this classification.

II. Related works

Generally, there are different ways to perform image segmentation. Different approaches have been investigated: Recurrent Neural Network [11], Convolutional Neural Network [12].
The publication of Long and all [10] is the first paper that introduces fully convolutional neural network to image segmentation. The main insight is the replacement of fully connected layers by convolutional layers. Fully convolutional network is implemented in VGG-NET. Now, there are many fully convolutional network based on VGG-NET: VGG16, VGG19 [2]. Unlike traditional sequential network architectures, ResNET is a more particular architecture based on micro-achitecture model. In [6], He and all have introduced the first version of ResNet50. Fisher, Ronneberger and Brox [8] have created the U-net. It is a convolutional neural network that was developed for biomedical image segmentation. This network is based on a fully convolutional neural network and its architecture was modified and extended to work with fewer training image and yield more precise segmentation.To compare with our model, we implemented a U-net with the library fast.ai.

III. Data exploration

The dataset contains 100 satellite images and the corresponding ground truth masks. On those, white pixels represent roads and black pixels represent the rest. While exploring the dataset, we noticed that cars and trees may

cover the road, and that some areas that were looking like roads where not labelled as such (for example parking lots or walkaways). To avoid misclassification, it is important to use a classifier that can take into account nearby pixels in order to infer some information about the region that is being classified.

IV. Models and Methods

This section review the model we used to build our classifier

A. Model

The goal is to classify blocks of 16x16 pixels with a label of 1 if more than 25% of the pixels in the block is classified as road and 0 otherwise. As suggested in III, we want our model to take into account the context of a particular pixel in the image, we will classify a block (patch) on the center of a bigger image (window) of size : window_size x window_size. This parameter was chosen to fit the data the best.
To apply this idea, we have chosen to use a CNN. The advantage of using CNN is its ability to develop an internal representation of a two dimensional image. It allows the model to learn position and scale in variant structures in the data.
The main characteristic of this architecture is the convolutional aspect. It performs a matrix multiplication between the input and a filter with a certain size that leads to feature map. Once a feature map is created, we pass each value in it through a non linear function : the ReLU function.
Figure 1 resume a classical CNN architecture.

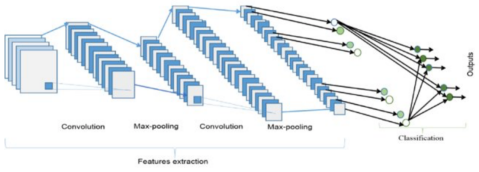

Fig. 1: Classical CNN Architecture

Note that in our case, the output is a real value between 0 and 1. To build our model, we have used the library Keras. It is an interface of the TensorFlow library. Keras library contains common commands for convolutional neural networks as layers, activation functions, dropout, normalization... The specific architecture we used to train is based on the Xception architecture [1]. The original Xception neural network contains 36 convolutional layers forming the feature extraction. In our case, we have used only 11 convolutional layers. Here are the main characteristics of our model :

B. Image augmentation

To have better results, we need to augment its size.
For this purpose, we have performed different transformations for each image. We have used several rotations, vertical flip and horizontal flip. Applying these transformations enables this model to take into account different representations of the image.

C. Padding

The padding defines how the border of a sample is handled.
In our model, we have chosen to use a padding of the input such that the output of each layers has the same dimension as the input. The use of padding avoids the loss of information in the corners of the image and the shrinkage of the outputs when using windows.

D. Separable Convolutional layer

When we use traditional convolution, the filter slides through the image given a certain stride.
The separable convolution layer is a variation of the conventional convolution that was proposed to compute it faster ([1]). It performs a depthwise spatial convolution which acts on each input channel separately followed by a pointwise convolution which mixes together the resulting output channels.

E. Activation function

We have chosen the ReLU function as it is the most convenient function in the case of our model.

F. Pooling layer

After performing the features map, we need to reduce the spatial size of the convolutional features. In order to choose the dominant features, we return the maximum value from the window covered by the filter. We repeat it at the end of each block.

G. Drop out layer

At the end of the model, we have performed a drop out. It is an approach to regularization which reduces dependant learning between all neurons. It allows to learn more robust features and to reduce the training time of one epoch.

H. Residual blocks

A particular feature of our model is the use of residual blocks. Traditionally in a neural network, each layer feeds into the next one. When we use residual blocks, a layer feeds into the next one but also in the layers further away (by summing the result to the output.) This method leads to great improvement of accuracy and reduces the learning time as shown in [9].

I. Visualization of the architecture

Figure 2 and Figure 3 represents the architecture of our neural network.

J. Other models

In this section, we focus on other approaches:

1) Naive model: It is easy to realize that there is more background areas than road areas. A naive approach leads to predict always 0. That is to say to classify all blocks as background. We have presented the accuracy given by this model in the section VI.

Actions	Filter Size	Kernel Size
Data augmentation block		
Entry Block	32	3
	64	3
Block 1	128	3
Residual Block	128	1
Block 2	256	3
Residual Block	256	1
Block 3	512	3
Residual Block	512	1
Block 4	728	3
Residual Block	728	1
Final block	1024	3
Average pooling		
Final Dropout		

Fig. 2: Architecture of the CNN

Entry Block
Convolution + Normalization + ReLU
Convolution + Normalization + ReLU
Save residual
Block x
ReLU + Separable Convolution + Normalization
ReLU + Separable Convolution + Normalization
Max Pooling
Residual Block
Convolution + Sum of layers output and residual
Save residual

Fig. 3: Architecture of the blocks

2) U-net: The second comparative model that we have used is the U-net. Its architecture is composed of two paths: the contracting path and the expansive one.
The model wasn't implemented manually. Using fast.ai library, we have access to a ResNet34 model following the U-net architecture. The main difference is that this model does per-pixel classification. We then do the average of the predictions in the patch to label it as road or background. Visual results of this method are shown in Figure 4.

V. Post-processing

We chose to implement an intuitive correctional method for our predictions. More precisely, given a ground truth prediction of a test image, we delete obvious visual inaccuracies such as an isolated patch of road or a

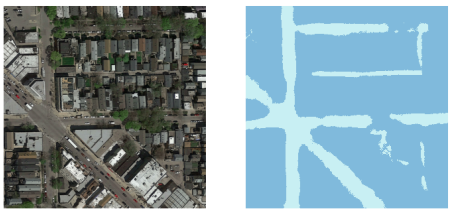

Fig. 4: Image and prediction of U-net model

road missing a piece. Through a series of consecutive correction functions, we improve the result and make the ground truth prediction smoother. This process is illustrated in Figure 5, where the left image is the raw prediction, and the right one is the corrected version. For the sake of clarity, some of the changes are emphasized via the orange circles.

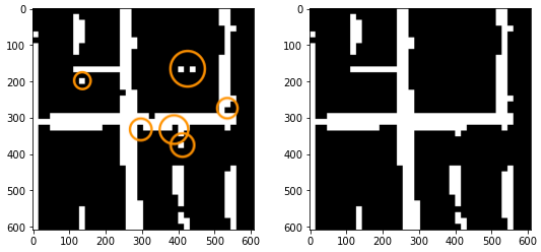

Fig. 5: Raw and post-processed prediction

VI. Evaluation of the performance

In order to validate the performance of our model, we tested it on data with known ground truth. To do so, we have split the all data in two part: the training set and the validation set. Thus 80% of the data was used to train the data and 20% was used for cross-validation.

The loss function used to train our model is a sigmoid categorical cross-entropy function which has been minimized by the Adam optimizer. The Adam optimization algorithm is an extension to stochastic gradient descent. Unlike the classical stochastic gradient, the parameter of learning rate changes during the algorithm. The initial rate is set to 0.01 and is divided by two each time the training accuracy does not evolve for five iterations.

We have chosen to compare the accuracy of different models: a naive model that always predict 0 (Naive), the CNN without data augmentation (CNN 1), the CNN with data augmentation (CNN 2) and finally the U-net with the fast.ai library.

Methods	Accuracy $\pm \sigma$
Naive	$73.9 \pm 1.4\%$
CNN 1	$90.8 \pm 0.9\%$
U-net	$91.2 \pm 0.3\%$
CNN 2	$92.1 \pm 0.7\%$

Tested model and their accuracy

VII. Discussion

We observe that the highest accuracy was achieved with CNN and data augmentation.

The following figure shows the predicted road patches on a test set image. Our model segments the image in a human-like manner. The important thing to note is that it works just as well for the roads in the center as for the roads on the border.

Fig. 6: Example of a prediction on the test set

VIII. Conclusion

In conclusion, the result provided by the model are quite impressive. CNNs make a good fit to solve this classification problem. As shown in Figure 6, results are visually satisfying and this is consistent with the reported accuracies.

REFERENCES

[1] Francois Chollet,"Xception: Deep Learning with Depthwise Separable Convolution", 2017

[2] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition, 2015

[3] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, "Inception-V4, Inception-ResNet and the impact of residuals connections on Learning", 2016

[4] Yingying Wang, Yibin Li, Yong Song and Xuewen Rong, " The Influence of the activation function in a convolution neural network model of face expression recognition", 2020

[5] Sebastian Bittel, Vitali Kaiser, Marvin Teichmann and Martin Thoma, "Pixel-Wise Segmentation of street with Neural Networks", 2015

[6] He, Zhang, Ren and Sun, "Deep Residual Learning for Image Recognition", 2015

[7] Adeodato, Vasconcelos, Santos, Arnaud, "Neural Networks vs Logistic Regression: a Comparative Study on a Large Data Set

[8] Ronneberger, Fisher, Brox, "U-net: Convolutional Network for Biomedical Image Segmentation", 2015

[9] He, Kaiming, X. Zhang, Shaoqing Ren and Jian Sun. "Deep Residual Learning for Image Recognition." 2016

[10] Long J, Shelhamer E, Darrell T (2014) Fully convolutional networks for semantic segmentation

[11] Bengio Y et al (2009) Learning deep architectures for ai

[12] LeCun Y, Bengio Y (1995) Convolutional networks for images, speech, and time-series