Satellite Image Road Segmentation Using Neural Networks

Mohammed-Ismail Ben Salah[1] **and Léonard Berney**[2]

[1]mohammed-ismail.bensalah@epfl.ch
[2]leonard.berney@epfl.ch

ABSTRACT

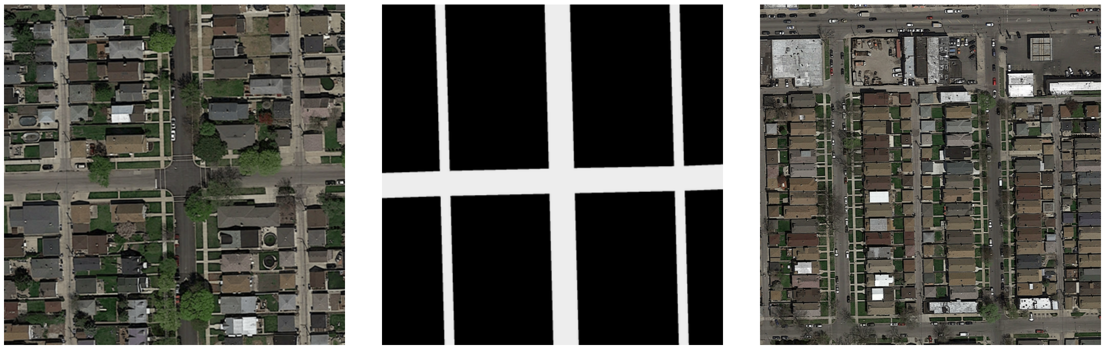
This report presents a method for detecting roads in satellite images using the U-Net neural network. In order to be able to be able to classify and segment the images,the-state-of-the -art methods are based on CNN (Convolutional Neural Networks). The problem that is presented is this project is to be able to segment images between roads and everything else. To this purpose, we looked into the current literature and found-out that one efficient neural net when it comes to images segmentation is the U-Net[1] which is used in medical image segmentation. Going from this Neural Net, we looked into how to train it efficiently and fine tune the different parameters of the model.

Introduction

Image segmentation is a computer vision field where we're looking to partition an image in segments (group of pixels) given a specified criteria (e.g. find dogs in photos). Nowadays classical techniques such as k-means,edge detection or region growing clustering are used in conjonction with machine learning based methods such as Deep Learning with CNNs (Convolutional Neural Networks). With the democratization of GPUs to execute parallel computations and in general higher computational power, the training of neural networks is today more time efficient. One special class of neural networks that is the-state-of-the-art in object recognition is CNNs,which we will use. In our problem we're looking to find roads on a dataset of satellite images. This report will present in details : 1) The dataset that was used in order to train and evaluate the neural net. 2) What was the given topology of our neural net. 3) What was the purpose of data augmentation and how it was used. 4) The results and last but not least 5) The different approaches that we used during the project to enhance the predictions that where given by our Neural Net.

1 Dataset

We are provided with a training set containing 100 RGB aerial images mainly consisting of urban areas. The labels are encoded in black and white images, where white pixels represent road and black pixels represent background. The images from the training set have a size of 400 by 400 pixels. The test set that is used to make the predictions that will be submitted to the CrowdAI platform consists of 50 608 by 608 pixels images.



(a) Train image **(b)** Ground truth image **(c)** Test image

Figure 1. Sample images from the dataset

We also trained with another dataset that is containing 449 RGB High-Resolution (2500 x 2500) images of Chicago

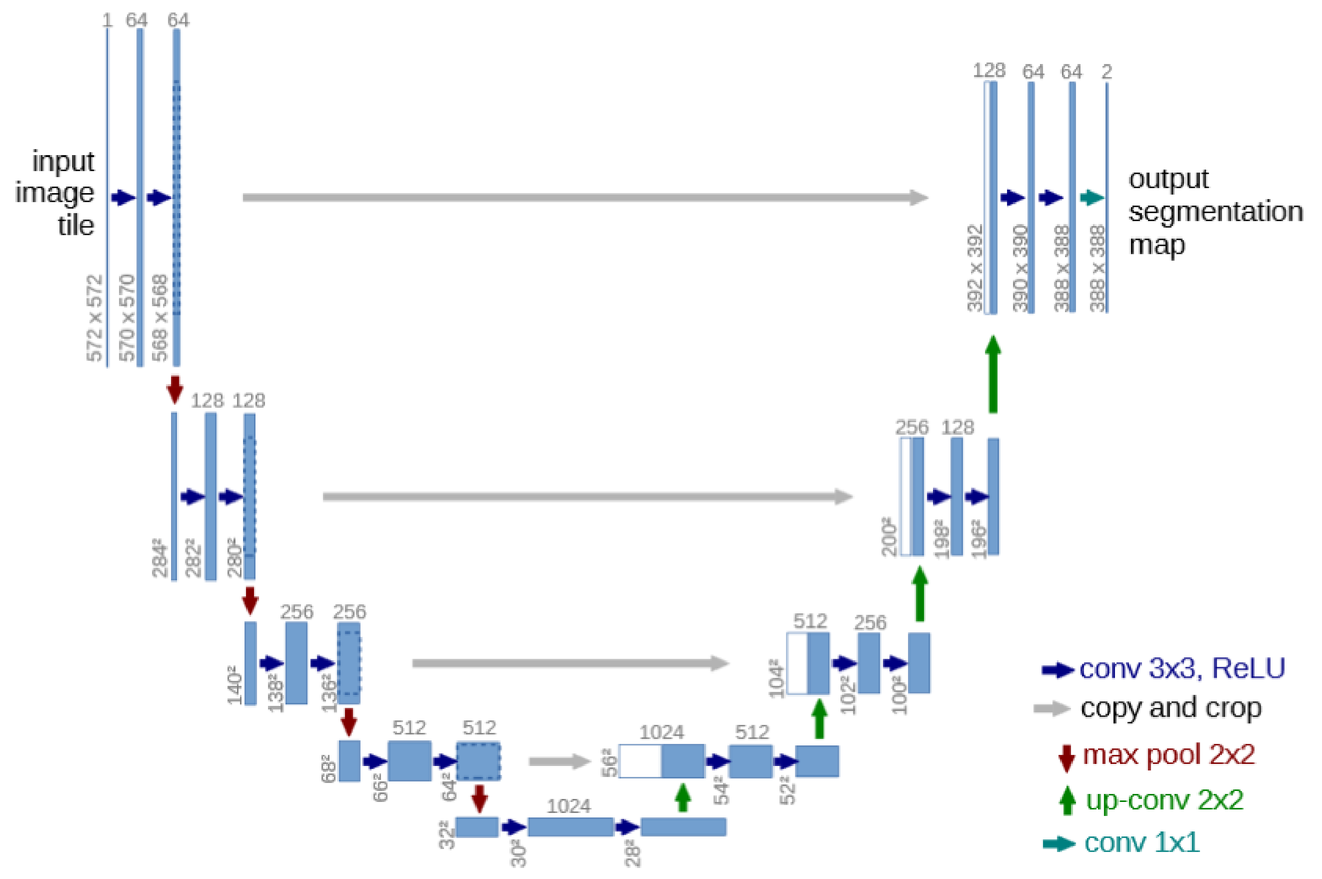


Figure 2. Architecture of the U-Net, image taken from https://arxiv.org/pdf/1505.04597.pdf

(https://zenodo.org/record/1154821.XAFSceLjIRA). The images given in this dataset are of the same size in the training and test set,which allows us to not lose information and therefore have a better training process.

Figure 3 shows some the issues that arise when labeling satellite images; some of the road is covered by trees, cars or building but is still labelled as road.

2 Model

We build a model using neural networks as they are currently the state of the art when it comes to image segmentation. They also have the advantage of being able to be trained on GPUs which can dramatically decrease the processing time. Another characteristic of neural networks is that they can greatly benefit from a large amount of training data. In our case, however, we only have access to a very limited quantity of data, which makes U-Net[1] a good candidate, as it was designed to work with low to medium amounts of data. It also has a low memory footprint and relatively fast training times compared to other convolutional neural networks. Figure 2 shows the U-Net used in the paper describing it. The only difference with our implementation is the image input size that was 572 by 572 pixels. We also use after each convolutional layer a Dropout layer in order to avoid overfitting.

3 Pre-processing

3.1 Image processing

In order to enhance the distinction between the roads and the background,it was decided to use contrast stretching as it would make road more distinguishable from the background.

3.2 Data augmentation

Even though U-Net is designed to work with low amount of data, it is still beneficial to apply data augmentation. Since neural networks detect roads by finding pattern in the images, it can easily overfit by finding patterns in the training set that are not directly related to the road, which is likely to be the case when the amount of train data is small. Before feeding the data to the neural network we randomly apply a series of linear transformation to the images:

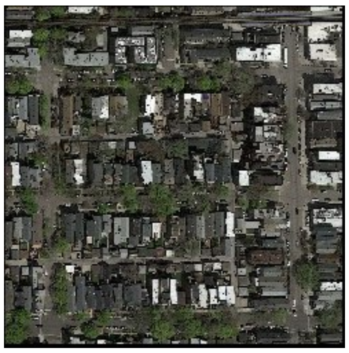
- rotation, 0 to 45 degrees
- horizontal and vertical shift, max 10% of image width
- horizontal and vertical flip
- zoom, $\pm 10\%$
- Mirroring when out of range pixels

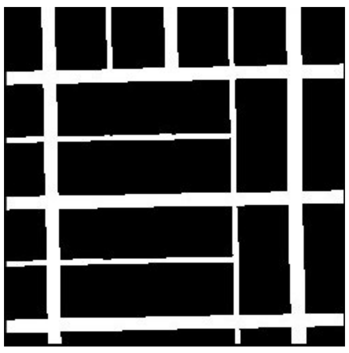
These simple transformations not only artificially create more data but also help break some of the patterns present in the original training set. The images from the training set are all very similar to the one shown in Figure 3 a). By applying rotations and flips/shifts to these image, we reduce the risks that the neural network learns to detect only road that are perpendicular to each other and run parallel to the edges of the image. Randomly zooming the images helps when not all the images have the same scale. In our case the training images all have the same level of zoom. In the test set however, images have a wider field of view which means that roads appear narrower than in the training images.

4 Training

Our neural net has the following parameters. Kernel size is (3,3),the number of input filter is 64. We take 10 % of the available dataset as validation data (without performing any data augmentation) . The optimizer used is Adam with a learning rate of 0.001. The loss function used is binary cross-entropy (as we need to make a binary decision). The model is trained for 50 epochs but typically converges after 25. It was trained on a Google Compute VM with 8 CPU,30 Gb of RAM and 4 Tesla P100 GPUs.

5 Results


(a) Validation set image


(b) Ground truth image

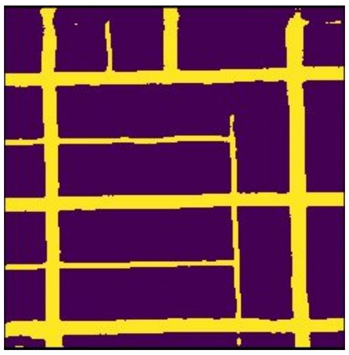

(c) Prediction

Figure 3. Sample prediction done on validation data

method	dataset	#images	f1 score
baseline	OD	90	0.869
contrast	OD	90	0.872
data augmentation	OD	1000	0.87
data augmentation	CD	10000	0.912

Table 1. F1 score obtained for every method (computed on validation data)

The baseline is defined as follow : Using the original dataset (OD) and using no data augmentation at all. As we can see constrast pre-processing enhance the score but is clearly not enough as the neural net still have the same data to train. To correct this hindrance we will use data augmentation as explained above. At least we also use the Chicago Dataset (CD) to have more original data as on which we can train and allow the neural net to have more information about the structure and the features to extract.

6 Discussion

Even though we were obtained acceptable results there are multiple things that we wanted to try but were unable to. We believe that one of the main factor that prevented us from achieving a better score was the quality of the labels in the dataset. It would be interesting to pre-process the ground truth images to remove any mislabelling of the trees as road. This would probably lead to a more accurate classification of actual road pixels. It would however create bad predictions as some of the trees actually need to be labelled as road, but given a better road pixel prediction it should be possible to post process the output images to reconstruct the missing parts of the roads.

6.1 Experimentation

In this section we discuss different experiments we conducted that we did not include in the final model because they did not improve the performance significantly.

6.1.1 Transfer learning

The concept of transfer learning is that it is possible to use a neural network trained on a problem and reuse it on a different but related problem. This method was shown to give good results while saving computation time[2].

We can split U-Net in two parts, the "down" part corresponding to an encoder and the "up" path corresponding to a decoder. We replaced the decoder part by an instance of ResNet-50[3] that was pre-trained on ImageNet and while training the neural network only the decoder part was trained. As expected it greatly decreased the training time and the results were good but not better than what we could obtain with U-Net alone. We could probably have obtained better results by training the decoder on a few epochs and then unfreezing the layers of the encoder and finish training the whole network on our dataset but ResNet-50 being more time consuming to train than U-Net we abandoned this path due to time constraints.

6.1.2 Using a different loss function

In the final model we use binary cross-entropy which is usually used when trying to maximize accuracy because they both measure the same thing. However, in our case the metric used is the f1 score which is not aligned with the binary cross-entropy, that is minimizing the loss will not necessarily give a better score. Our idea was to use the opposite of the metric as the loss as it means that minimizing the loss corresponds to maximizing the f1 score[4]. The f1 score is not differentiable but if we consider the outputs of the neural network as probabilities, it is possible to build a loss function aligned with the f1 score.

This method brings an improvement when the score is below 0.87 but when when the f1 score approaches 0.9 minimizing the binary cross-entropy was at least as good.

Conclusion and future perspectives

In final the performance of the model has been satisfactory, however it could benefit of a post-processing of the prediction in order to enforce the structural coherency and remove small clusters of false positives.

References

1. Ronneberger, O., Fischer, P. & Brox, T. U-net: Convolutional networks for biomedical image segmentation. *CoRR* **abs/1505.04597** (2015). 1505.04597.

2. Pan, S. J., Yang, Q. *et al.* A survey on transfer learning. *IEEE Transactions on knowledge data engineering* **22**, 1345–1359 (2010).

3. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. *CoRR* **abs/1512.03385** (2015). 1512.03385.

4. Eban, E. E. *et al.* Scalable learning of non-decomposable objectives. *arXiv preprint arXiv:1608.04802* (2016).