Road segementation: a Machine Learning Challenge

Léo Meynent, Vincent Tournier, Marijn van der Meer

***Abstract*—In the context of the road segmentation challenge proposed in the scope of the Machine Learning course of the EPFL on *aicorwd.com*, our team proposed a deep-learning solution based on a U-Net architecture, considered to be the current state-of-the-art for segmentation problems. By implementing pre and post-processing steps, and using the Adam optimiser with a Focal Loss, we were able to obtain a satisfactory model capable of obtaining a F1-score of 0.886 and accuracy of 0.940 on our test set.**

I. Introduction

Image segmentation is an intrinsic component of many visual comprehension systems. It consists in partitioning images into several segments or objects. It can be seen as a classification problem of pixels with a label (semantic segmentation) or partitioning of individual objects (instance segmentation) with an extra object detection step in addition to pixel-level classification. Image segmentation helps us better comprehend the content of images and is a very significant topic in image processing and computer vision. Numerous computer vision tasks, such as image compression, scene comprehension, location of objects, etc. require intelligent segmentation of an image, in order to understand what it contains and to allow easier analysis of each part. To that effect, many algorithms have been developed. But with the advent of deep learning in computer vision, many deep learning models have also emerged to comprehend what real-world object is represented by each pixel in an image. Visual input patterns can be learned through deep learning in order to predict the classes of objects that make up an image. In this project, we aimed to tackle the challenge of binary road semantic segmentation of satellite images.

II. State of the art

Before deep learning methods, classical approaches to image segmentation, as surveyed in [1], ranged from the earliest methods such as thresholding [2], histogram-based bundling [3], k-means clustering [4], watersheds [5], to more advanced algorithms such as active contours [6], graph cuts [7], conditional and Markov random fields [8], and sparsity-based methods [9]. In recent years, however as sus-mentioned, deep learning models have given rise to a new generation of image segmentation models with remarkable performance improvements [1]. Current classical deep learning architectures for image processing are standard convolutional neural networks (CNN) and their specific frameworks (AlexNet, VGG, ResNet, . . .) [10], recurrent neural networks (RNNs), long short term memory (LSTM) [11], encoder-decoders [12], and generative adversarial networks (GANs) [13]. In our case of binary semantic image segmentation, we chose an encoder-decoder architecture, more precisely a U-Net [14]. The U-Net was initially developed for biomedical image segmentation, but is now widely used outside of the medical circle because, as stated by fast.ai teacher Jeremy Howard, *”in fact, basically every Kaggle winner in anything even vaguely related to segmentation has end up using U-Net. It’s one of these things that everybody in Kaggle knows is the best practice”*.

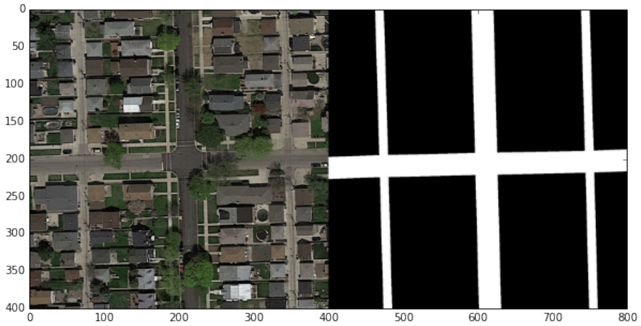

Figure 1: Example of training data. Left: satellite Google-Maps image with roads. Right: corresponding ground-truth map with road pixels in white and the rest in black

III. Models and Methods

A. Image pre-processing

Our training data consisted of a set of a hundred 400×400 RGB satellite images acquired from Google-Maps. This set came with ground-truth images where each pixel was labelled as {road=1, background=0} (c.f. Fig.1). This data was then standardised using Z-score normalisation

$$X' = (X - \mu)/\sigma$$

A test set of 50 images of size 608×608 was also provided. Those were standardised according to the mean μ and standard deviation σ of the training set. For computational reasons, training images were cut into four patches of size 200×200. The training data set was further augmented using different degrees of rotations on the patches (45°, 90°, 135°, 225° and 315°). This allowed the model to have more data to train on and learn more effectively, a strategy proposed in [14] for the original U-Net. The 45° rotation allowed us to tackle the lack of diagonal roads in the training set, while being more present in the test set. But, due to the

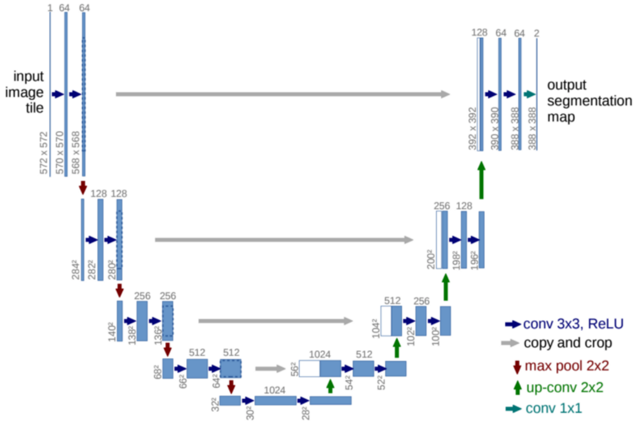

Figure 2: U-Net architecture with cross connections (example for 32x32 pixels in the lowest resolution) [14]. Cross-connections are visible in grey. Encoder/down-sampler part is on the left and decoder/up-sampler on the right.

loss of pixels in the process of handling a 45°-turned image, every 45° rotation produced only 100 new 200x200 patches, whereas full 90° rotations provided 400 of them. Overall, this augmented our training data from 100 images to 1200 samples.

B. *Model architecture*

U-Net is a CNN architecture specifically developed for biomedical image segmentation [14]. U-Nets show high performance in segmentation tasks and image processing, such as super resolution. They have been found to be particularly effective in cases where the output and inputs are of similar size. The network is U-shaped, as it is divided into a down-sampling/encoder section that forms the left side and captures context and an up-sampling/decoder for the right that enables precise localisation. The down-sampling or contracting part has a FCN-like architecture that extracts features with 3×3 convolutions. The up-sampling part uses transposed convolution, reducing the number of feature maps while increasing their dimensions. The feature maps from the down-sampling section of the network are copied into the up-sampling segment to ensure that pattern information is not lost. Lastly, a 1×1 convolution processes the feature maps to produce a segmentation map that classifies each pixel of the input image [1]. Traditional U-Net architectures result in a lack of fine detail. To avoid this, skip connections cross from same sized parts from the encoder to the decoder (grey arrows in Fig.2). Our U-Net is as shown in Fig.2, except for the input and output data size, which are of size 200×200. Drop-out probability for dropout layers was set to 0.0 everywhere as this provided the best results.

C. *Loss and evaluation metrics*

For our task of binary semantic image segmentation, we started our model with binary cross entropy (BCE) as a loss function, defined as a measure of the difference between two probability distributions for a given random variable or set of events. Our choice was based on the fact that it is widely used and said to work well for pixel level classification in segmentation [15][16]. Nevertheless, we later switched for our final model to a Focal Loss, a variation of BCE that works well for highly imbalanced class scenarios [15] such as is the case for our data, where road pixels are less common than background. Focal loss is defined as:

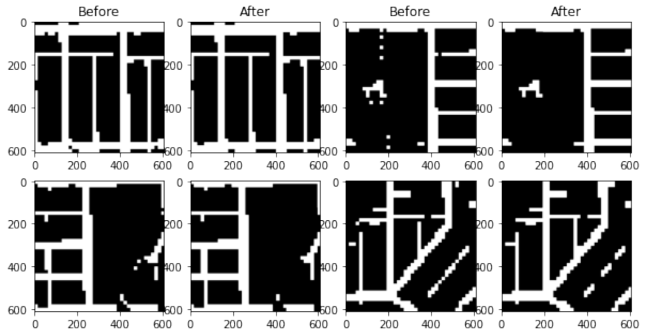

Figure 3: Examples of the effect of post-processing (see III-E) on 4 masks created by our model.

$$FL(p_t) = -\alpha_t(1-p_t)^\gamma log(p_t)$$

Where $p_t = p$ if $y = 1$ and $1 - p$ otherwise. The hyper-parameters were chosen as $\gamma = 5.0, \alpha = 0.75$ by cross-validation (see Table I). Using a focal loss instead of a standard BCE improved our score on AICrowd (F1-score from 0.842 to 0.884 see IV).

Regarding evaluation metrics, we used an F1-score and accuracy as these were used to evaluate our model on AICrowd. We expected the F1-score to be more representative of the reality of the performance of our model as the classes were highly imbalanced.

D. *Training*

The above-mentioned model was trained on a 100 epochs using an Adam optimiser, a batch size of 16 and evaluated with 5-fold cross-validation.

E. *Post-processing*

The mask produced by our model trained as in III-D were mostly of good quality, but they still featured easy to remove artefacts using some image processing techniques. Thus, we used a post-processing java program one of us coded last year for this very course, in order to remove those artefacts. We decided to remove any isolated 16 pixels wide chunk on the generated ground-truths, arguing that no road can have neither end nor beginning and simply be a simple patch in the middle of the background. To do so, we first applied an erosion on the binary mask with the structuring element being a square of 15 pixels of size. Chunks being 16 pixels large, this shrinks all the roads on the mask, and any isolated block is reduced to its center, a square 2 pixels wide only.

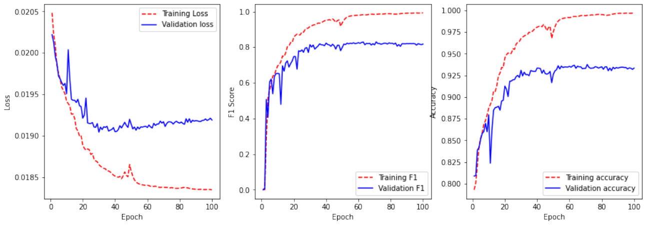

Figure 4: Focal loss (left), F1 score (middle) and accuracy (right) over a 100 epochs on the training and validation sets (20%). At the end of 100 epochs, values on training: loss 0.0184, F1-score 0.9926 and accuracy 0.9971. On validation: loss 0.0192, F1-score 0.8176 and accuracy 0.9333.

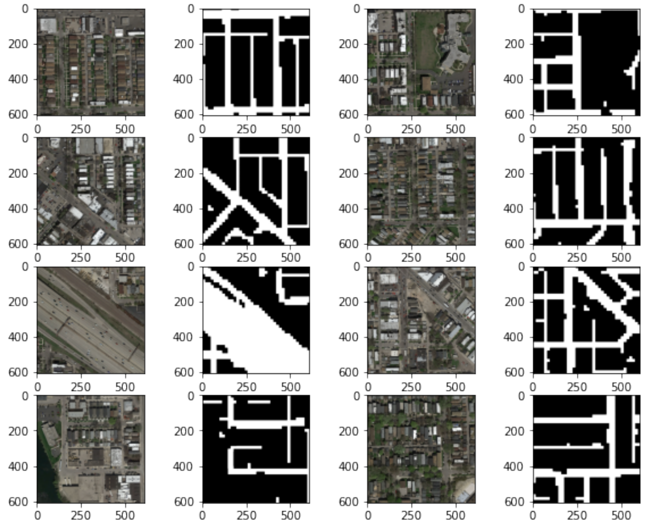

Figure 5: Predicted masks for 8 test images. Road pixels are in white (1) and the rest in black (0).

We then used a median operation with a square of size 3 as a structuring element to remove those isolated pixels, and then performed a dilation, opposite of the erosion, to bring back all roads to their original width. The result of this post-processing can be seen in Fig.3.

An unwanted side effect was the deletion of some correct pixels on diagonal roads that were predicted too thin. We worked around this issue by not post-processing the 3 most affected test pictures, i.e. pictures 47, 48 and 50. This post-processing improved our F1-score from 0.884 to 0.886 on AICrowd.

IV. Results and discussion

As a baseline for this challenge, a simple CNN model with two convolutional and pooling layers with soft-max loss was provided. This model was trained with an SGD optimiser with momentum and with a learning rate of 0.01 decaying exponentially. This model had an F1-score of **0.660** and accuracy of **0.782** on AICrowd.

During training as mentioned above in III-D, the evolution of the Focal loss, F1-score and accuracy over a 100 epochs was plotted (Fig.4). We can see that the validation metrics and loss follow nicely with approximately the same shape as the training values, but just a bit of lower performance. Overall they both seem to stabilise around 100 epochs. We trained for longer (1000 epochs) but this showed no improvement, hinting that we were just overfitting over a 100 or that we had stabilised.

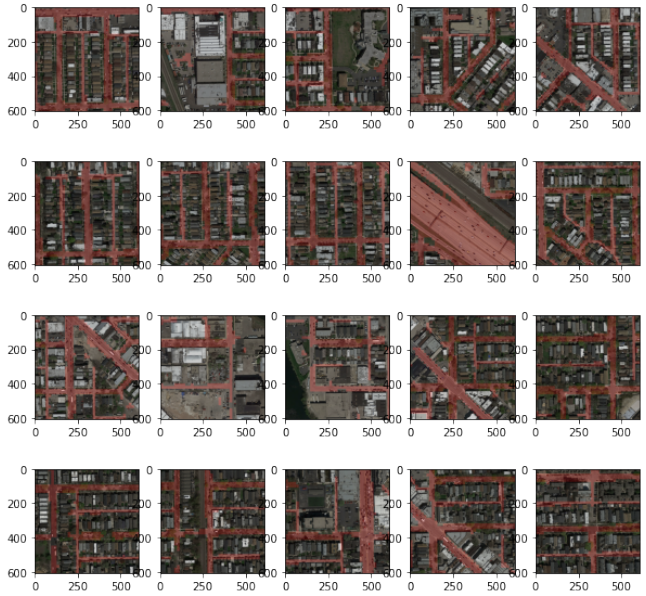

Figure 6: Predicted masks (roads in red) overlapping 20 test images by training as in III-D.

With post-processing (III-E), this provided an F1-score of **0.886** and accuracy of **0.940** on AICrowd (baseline of F1-score of 0.660 and accuracy of 0.782). From that score, we can see that our model was better than the baseline.

A few predictions on the test set can be seen in Fig.6 for masks overlapping test images and Fig.5 for images next to their predicted masks.

Visually, we can see that our model is in general very good at segmenting straight roads, but a a little less successful with oblique ones. Also, it often gives false negatives for smaller roads, especially diagonal ones, that are partially hidden by their environment such as cars or trees.

A. *Possible improvements:*

To improve our model even further, we could have, if we had had more time, performed more cross-validation on hyper-parameters such as the pixel foreground threshold for example, i.e. the percentage of pixels required to assign a foreground label to a patch. We also could have tried different optimisers in our model to see if one improved Adam's results. Sometimes, U-nets are completed with skip connections inside the encoder as well (i.e. ResNet as encoder), which is supposed to improve the performance and this could also have been explored. Finally, in classical segmentation different losses like the Dice score and Jaccard index are often used. With more time, we could have implemented those to see if they out-performed our Focal Loss.

V. Conclusion

In general, we estimate that our model is satisfactory for the task of road segmentation. Most roads are correctly classified and there is a low number of false positives. Even with a final F1 score of 0.886, our model competes well with the best ones that were at 0.909 at the time we write this report. Its main weaknesses still lie with oblique roads and partially hidden ones.

VI. Annexe

Table I

Gamma	Alpha	F1-Score
1.0	0.60	0.7964
1.0	0.75	0.0000
1.0	0.90	0.0000
1.0	1.00	0.4826
2.0	0.60	0.6901
2.0	0.75	0.7551
2.0	0.90	0.6511
2.0	1.00	0.5526
5.0	0.60	0.7918
5.0	0.75	**0.8135**
5.0	0.90	0.7991
5.0	1.00	0.6157

Table I Cross-validation for hyper-parameters of Focal Loss, F1-Score being computed as the median from 5-fold cross-validation over 20 epochs. The best combination of hyper-parameters from this is $\gamma = 5$ and $\alpha = 0.75$.

References

[1] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: A survey," 2020.

[2] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[3] Q. Wang and Z. Wang, "A subjective method for image segmentation evaluation," in *Computer Vision – ACCV 2009*, H. Zha, R.-i. Taniguchi, and S. Maybank, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 53–64.

[4] T. G. Debelee, F. Schwenker, S. Rahimeto, and D. Yohannes, "Evaluation of modified adaptive k-means segmentation algorithm," *Computational Visual Media*, vol. 5, no. 4, pp. 347–361, Dec 2019. [Online]. Available: https://doi.org/10.1007/s41095-019-0151-2

[5] L. Najman and M. Schmitt, "Watershed of a continuous function," *Signal Processing*, vol. 38, no. 1, pp. 99 – 112, 1994, mathematical Morphology and its Applications to Signal Processing. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0165168494900590

[6] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, Jan 1988. [Online]. Available: https://doi.org/10.1007/BF00133570

[7] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.

[8] A. Delaye and C.-L. Liu, "Multi-class segmentation of free-form online documents with tree conditional random fields," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 17, no. 4, pp. 313–329, Dec 2014. [Online]. Available: https://doi.org/10.1007/s10032-014-0221-z

[9] S. Minaee and Y. Wang, "An admm approach to masked signal decomposition using subspace representation," *IEEE Transactions on Image Processing*, vol. 28, no. 7, pp. 3192–3204, Jul. 2019.

[10] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, p. 1735–1780, Nov. 1997. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735

[12] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, Eds., vol. 27. Curran Associates, Inc., 2014, pp. 2672–2680. [Online]. Available: https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf

[14] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015.

[15] S. Jadon, "A survey of loss functions for semantic segmentation," 06 2020.

[16] S. Zhao, Y. Wang, Z. Yang, and D. Cai, "Region mutual information loss for semantic segmentation," *CoRR*, vol. abs/1910.12037, 2019. [Online]. Available: http://arxiv.org/abs/1910.12037