An attention based U-Net Model for Road Segmentation
Group IDA - Project 2B

VIEGAS MILANI, Adriano (338788) SOARES, Diogo (346798) KREMER, Iris (337635)
adriano.viegasmilani@epfl.ch diogo.sousasoares@epfl.ch iris.kremer@epfl.ch

***Abstract*—The recent rise of systems relying on georeferenced data has drastically increased the need of providing maps of human-inhabited regions. We already have performant satellites that provide us with an immense amount of image data, but segmenting roads on these images is a tedious task for humans. Luckily, this task can be assisted by machine learning models such as convolutional neural networks. In our project, we are addressing this problem using several models, with the goal to explore existing solutions for the road segmentation task and identifying the most promising approaches.**

I. Introduction

Convolution neural networks (CNNs) have proven to be very efficient when it comes to the analysis of images. While traditional simple CNN architectures already perform decently, recent work has tremendously improved the information extraction process in images through techniques with new kinds of layers and improved layer structures. As our task is to perform road segmentation on satellite images from Google Maps, we aim at exploring some of these recent techniques and compare them to traditional CNN methods.

We start by exploring the data provided to us in section II. Since our dataset is really small, we need to augment it. Our augmentation procedure is described in section III. We then present four classifiers in section IV, each of which use a different approach to solve our task. We then tune and experiment with them according to our descriptions in section V, and compare their performances using binary accuracy and f1-score as evaluation metrics in section VI.

II. Data Exploration

Our training dataset is composed of 100 satellite images, each with 400x400 pixels, coupled with 100 ground truth images as the labels. Conversely, the testing test entails 50 satellite images with 608x608 pixels. This dataset contains roads of various types, with different directions and some obstacles, but since they are satellite images, perspective is always top-down, which removes any kind of possible stretching and shearing of the roads in the images.

As we can see in figure 1 some trees overlap with the roads, small roads are partially hidden by buildings, and there are sudden road color changes, which makes the classifier's work harder. Additionally, it might also be the case that the roof of a building has the same color as the road, or that there are parking spots as seen in the second image.

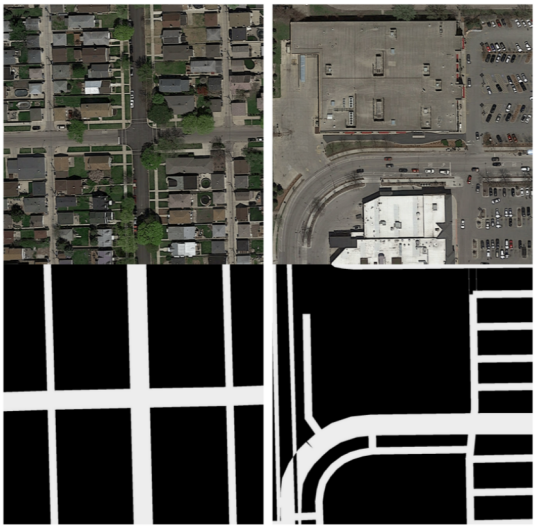

Figure 1: Satellite Images and their corresponding Ground truth

The ground truths are given pixel wise, in binary format. The corresponding ground truths are impervious to the trees and obstacles, therefore representing the roads correctly. Moreover, as we expected, the buildings are not represented as foreground. There is a more undefined line when it comes to parking spots and wide roads, with only some parts considered roads. This issue can be seen in the top left of the fourth image in figure 1.

III. Data Augmentation

In computer vision, it's common to use data augmentation when working with small training datasets. Our case was no different, so we decided to do the following augmentation procedures: Mirroring, and Rotation. Random Cropping could cause issues with resolutions in some models, so we did not implement it. Furthermore, some other data augmentation techniques were considered such as adding noise to the image and altering the tint, that being said we decided not to use them since our main focus was performance for this task specifically and not robustness.

Due to the size limits described below, we separate input images in 4-9 patches for U-net and TransUnet, and in 625 patches of 16x16 for Baseline and Dilnet, and run the model on these patches. It's worth noting that some patches overlapped, however we considered the cost of overlapping to be smaller than the benefit it brings of averaging the estimation in overlapping areas.

IV. Models and Methods

We explored several CNN architectures, which are presented in the following subsections. Our goal is to compare traditional approaches with different improved versions, and identify the best model for our task.

A. Baseline CNN

Traditionally, convolution neural network (CNN) architectures are used for image classification and segmentation. There exists a plethora of architecture variants to choose from. Therefore, solving our task using at first a simple CNN provides an essential baseline in our project.

Hereafter, we present the architecture of our simple baseline convolution neural network, with pooling and fully connected layers.

This CNN was built based on the code originally provided for the challenge, with very little modification compared to the original provided architecture.

Layer	Kernel Size	Stride	Output Size
input	-	-	16X16X3
conv_1	5X5	1	16X16X32
pool_1	2X2	2	8X8X32
conv_2	5X5	1	8X8X64
pool_2	2X2	2	4X4X64
fc_1	-	-	1X1X512
output	1X1	-	1X1X1

B. DILNET

For CNNs a central aspect of network architecture has remained largely in place. An issue with standard convolutional networks for image classification is that they progressively diminish resolution, until the image is represented in a tiny feature spaces that does not retain much spatial information. Despite the good results of traditional CNNs, the disregard for spatial acuity can prevent these models from achieving even better results. In the end, taking this into consideration can help in the analysis of complex scenarios, where multiple objects and configurations must be accounted for. One possible solution is to introduce dilation rates in multiple layers. This way, when applying the convolution, we increase the receptive field of the higher layers, compensating for the reduction in receptive field induced by removing subsampling.

We followed the theoretical derivation from [1] to produce the following architecture.

Layer	Kernel Size	Stride	Dilation Rate	Output Size
input	-	-	-	16X16X3
conv_1	2X2	1	1	16X16X16
conv_2	2X2	1	1	16X16X16
conv_3	2X2	1	2	16X16X32
conv_4	2X2	1	2	16X16X32
conv_5	2X2	1	4	16X16X64
conv_6	2X2	1	2	16X16X64
conv_7	2X2	1	1	16X16X64
global pooling	16X16	-	-	1X1X64
output	1X1	-	-	1X1X1

C. U-net

U-net is one of the most commonly used deep learning segmentation network, with the original paper currently having more than 34k citations according to google scholar. It was originally created for medical imagery [2], but is used for any kind of image segmentation. U-net is another solution for the problem The objective behind U-net was to solve three main issues: being able to train a model without a lot of available data, being able to assign class labels pixel wise instead of a single label for an entire image, and retaining spatial information. This is what we want for our task.

U-net's architecture, seen in figure 2, is composed of two main parts, a contractive path and an expansive path, also called encoder and decoder. They are very similar but opposed, the first part is a traditional architecture for CNNs. The second part enables localization of labels by expanding the result of the encoder. In more details, the contractive path comprises a series of four blocks containing 2 convolutions with a kernel of size 3, followed by max pooling with a pool size of 2. Each block has twice the dimensionality of the previous, and drop-out layers are added at the end of the path to perform implicit data augmentation and thus making the model more robust. Then, there is a last block without max pooling, before entering the expansive path. The decoder is also a series of four blocks that contain an up-sampling by using transposed convolution followed by 2 convolutions with a kernel of size 3. These convolutions are applied to the concatenation of the up sampling with the result of the convolutions at the same step in the encoder, this relation can be seen by the gray arrows in figure 2.

Thus, U-net takes as input a square image of at least 16 by 16 and up to 572 by 572 pixels [3], and returns a similarly shaped output, where each pixel is labeled, with the corresponding class in one hot encoding with confidence levels.

The version of U-net we use, that is adapted from Vidushi Bathia's implementation [4], is essentially the same, but with relu activation functions in each layer.

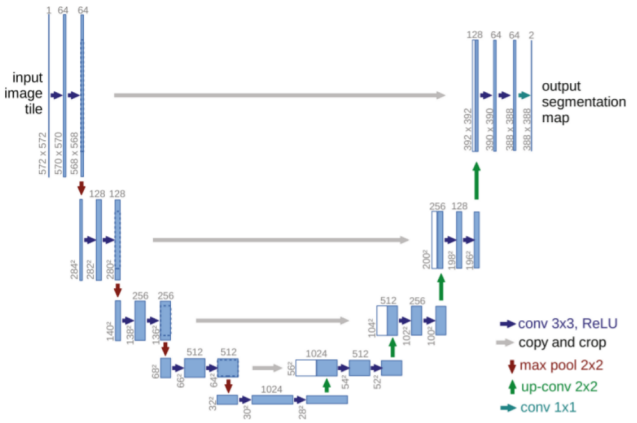

Figure 2: U-net Model architecture as presented by Ronneberger et al. [2]

D. TransUNet

Attention was first introduced in the famous paper "Attention is all you need" [5] in the context of NLP, and researchers soon realised it could be applied to any type of sequential data, including images.

The idea behind attention is to tell the network which part of the data is important to which other part, i.e. essentially where to look for information. Mathematically, it is defined by the formula

$$softmax(\frac{QK^T}{\sqrt{d_k}}) * V$$

where Q, K, V are the matrices of queries, keys, values respectively (one row per token), and d_k is the number of columns of K. Concretely, Q and K are multiplied to obtain the importance score of the key token for each query token. This matrix is divided by $\sqrt{d_k}$ for more stable gradients and normalized with a softmax. These are weights for the matrix of values, which represent the absolute importance of each token [6], [7]. When the set of tokens of the query and the keys are the same, it is called self-attention, otherwise it is cross-attention.

Although the attention computations are really heavy when applied to images, recent work in computer vision has made huge progress by using this technique. One approach was taken by J. Chen et al. [8], in which they implement TransUNet, a U-net model for medical image segmentation with a separate pipeline of attention layers. This pipeline maps the output of the last encoder layer into a latent space on which self-attention is computed, then feeds the attention map into the first decoder layer of U-net, as shown in figure 3.

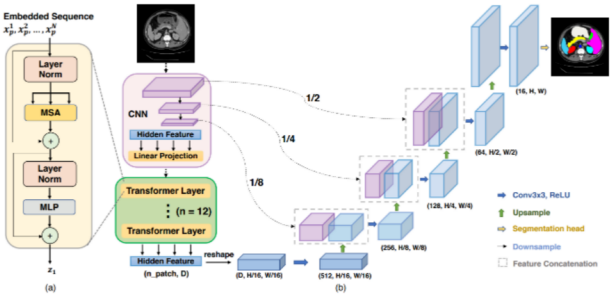

Figure 3: TransUNet Model architecture as presented by J. Chen et al. [8]

As this apparently efficient architecture caught our attention, we decided to implement our version of TransUNet for our road segmentation task, by augmenting our U-net implementation IV-C with the attention pipeline.

In our implementation, we made some major changes[1] to the architecture proposed by the original paper. First, due to computational power limits, we kept our version of TransUNet lightweight, so we reduced the number of transformer layers and the size of the latent space on which they perform. Secondly, we drastically modified the shape of the latent space. In the paper, they operate on a flattened 2D patches space with added positional embedding, which we did not know how to reproduce. So instead, we chose to keep a simple multidimensional latent space.

V. Experiments

A. Post-processing

After a certain model performs a prediction, the prediction result can be improved using post-processing techniques.

Empirically, it's often visible that the road classification shows some continuity, so we change classifications that do not possess 4-connectivity with the same label to the other label. This way, a patch denoted as road in the middle of background patches will turn into background and vice-versa. This can be seen in figure 4.

Figure 4: Applying Post-processing 4-connectivity erosion on DILNET output

[1]Refer to the comments in the code of our TransUNet implementation for more details on the changes made to the original TransUNet architecture.

B. Hyperparameter tuning

We performed a grid search to choose the best set of hyperparameters for each model. Due to timing constraints, a 2-fold cross-validation was used, favoring tuning more parameters over having more folds. Furthermore, we decided to tune on a random subset of our dataset (including augmented data) only. Since our data is quite homogeneous, the low number of folds and subset should not cause problems, although we often did observe quite a large difference in evaluation metric values between the two folds during tuning. We decided to only tune the learning rate for all models and the batch size + patch size for U-net and TransUNet, as there was no time left to tune the remaining parameters. Tuning is thus not perfect, and some hyperparameter choices may not push the models to their fullest. Still, table II shows the final chosen parameters

U-net Classifier		
Learning Rate	Batch Size	F1-score
0.001	16	0.8979
0.001	8	0.9065
0.001	4	0.8786
0.01	8	0.0856
0.003	8	0.7845
0.0003	**8**	**0.9081**
0.0001	8	0.8273

Table I: Some meaningful results of the tuning for the U-net model, after already selecting a patch size of 400.

Classifier	Patch Size	Batch Size	Epochs	lr
CNN Baseline	16	32	40	0.0005
DILNET	16	32	40	0.0005
U-net	400	8	40	0.0003
TransUNet	320	8	40	0.0003

Table II: Parameters chosen after tuning.

VI. Results

To evaluate the performance of our models, we consider the binary accuracy and the f1-score as metrics, with the f1-score being the most significant important metric.

We used CNN as the baseline model without data augmentation or post-processing for comparison with the extended models. Table **??** shows the comparison with results obtained for other models using post-processing and the augmented dataset. The entries are the best performance of our models, evaluated over the test set by Aicrowd.

Classifier	Binary Accuracy	F1 Score
CNN Baseline	0.812	0.702
DILNET	0.855	0.755
U-net	0.914	0.855
TransUNet	0.933	0.877
TransUnet + U-net	**0.936**	**0.883**

VII. Discussion

U-net and TransUnet have significantly better results than the other models, with TransUnet reaching a f1 score of 0.933. The differences in prediction style between the two can be seen in figure 5. For some images, U-net estimates better than TransUnet, so we decided to make an ensemble of the two models using the average of their confidence values pixel wise. This increased our results to 0.883 f1 score.

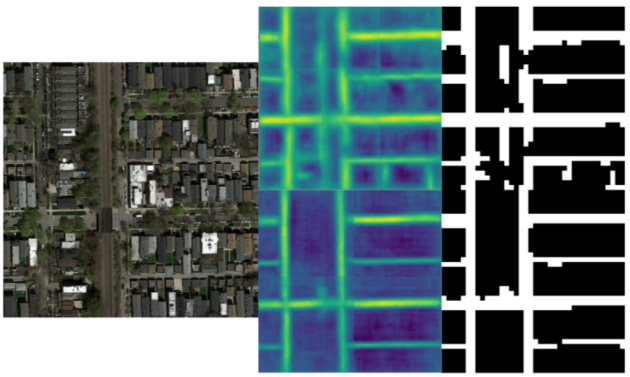

Figure 5: Predictions of a satellite image, pixel-wise and with 16*16 patches. U-net predictions are on the top row and TransUnet predictions on the bottom row.

VIII. Outlook

Although we obtained very satisfying results with our best model, there are several aspects we can still improve. First, doing a more profound tuning would have allowed more precise hyperparameters and thus better results, especially for TransUnet, where we know the patch size is not optimal. More advanced post-processing using machine learning models or mathematical filters might also improve our results. It is also possible to train deeper and more complex models than ours, such as those presented by S. Minaee et al. in their overview of the state of image segmentation [9], or with more generated data. In theory this could give better and more robust results, however the project timeline coupled with our hardware limitations did not allow us to do so.

References

[1] T. F. Fisher Yu, Vladlen Koltun, "Dilated residual networks," *arXiv preprint arXiv:1705.09914*, 2017.

[2] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[3] S. Piao and J. Liu, "Accuracy improvement of unet based on dilated convolution," in *Journal of Physics: Conference Series*, vol. 1345, no. 5. IOP Publishing, 2019, p. 052066.

[4] Vidushi Bhatia, "U-net implementation from scratch using tensorflow," https://medium.com/geekculture/u-net-implementation-from-scratch-using-tensorflow-b4342266e406, online; accessed 20th of December 2021.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[6] Jay Alammar, "The illustrated transformer," http://jalammar.github.io/illustrated-transformer/, online; accessed 23rd of December 2021.

[7] Yasuto Tamura, "Multi-head attention mechanism," https://data-science-blog.com/blog/2021/04/07/multi-head-attention-mechanism/, online; accessed 23rd of December 2021.

[8] J. Chen, Y. Lu, Q. Yu, X. Luo, E. Adeli, Y. Wang, L. Lu, A. L. Yuille, and Y. Zhou, "Transunet: Transformers make strong encoders for medical image segmentation," *arXiv preprint arXiv:2102.04306*, 2021.

[9] S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.