Machine Learning Project
Road Segmentation

Anas Machraoui, Camil Hamdane, Nathan Girard
Section of Life Science Engineering, EPFL, Switzerland

***Abstract*—Image segmentation consists in classifying certain parts of an image as a distinguishable feature from the rest of the image. In this report, we will introduce our take on a segmentation task, using a convolutional neural network to segment roads from Earth satellite images. We made use of dilated convolutional layers to increase the receptive field of our model without giving up on high resolution which could reduce the accuracy. Also, by coupling suitable data augmentation techniques, we were able to obtain a high accuracy.**

I. Introduction

The training dataset consists of 400x400 pixels satellite images extracted from Google Maps along with ground-truths images for training where each pixel is labelled as either a road or background. Our training set was originally consisting of 100 of these images along with their ground-truths before data augmentation. The evaluation set contains 50 images of 608x608 pixels, on which predictions are made on 16x16 patches all over the image. We then use a threshold to determine whether the patch should be classified as a road or a background. In section II and III, we will explore respectively the Fractal and U-net models and their architecture. In section IV, we discuss the data augmentation and how we used it to maximize generalization. In section V to VI we present and discuss our results as a baseline for possible improvements.

II. Fractal net

Going into the project, we experimented with a model used in medical image segmentation called FractalNet, depicted by the paper of Gustav Larsson et al. [6]. On paper, the model could exceed the performance of residual networks without even augmenting the data. FractalNet uses drop-path as a co-adaptation prevention method, leading to a fractal architecture, as the model goes in many different directions. A practical advantage of FractalNet resides in its ability to transition from shallow to deep during training. Therefore, it allows for a rather quick answer when in "shallow mode" and a more precise answer when in "deep mode".

The fractal model is constructed by sequence of blocks, containing each convolution and joining layers, between which a pooling operation is done. In our case, this model consists of 4 blocks with 3 convolutional layers each in sequence, which can be seen in *Figure 1*. It begins with 16 filters in the first layer, multiplied by 2 from block to block.

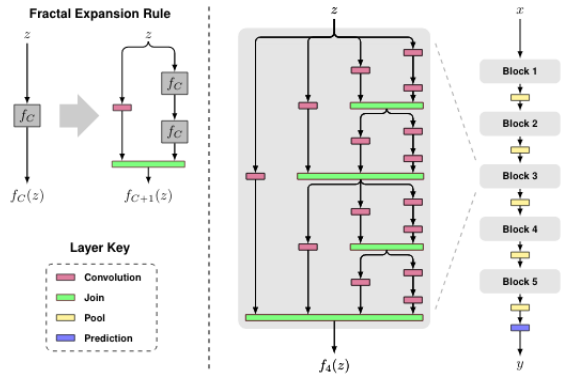


Fig. 1: Architecture of the fractal neural net.

Although it is originally designed for image classification tasks, producing a single Boolean output, we made it so the input is a 16x16 patch extracted from the images, matching the submission format, rather than inputting a 608x608 test image. It allowed us to reconstruct a whole image with predicted segmentation out of classified patches.

III. U-net

A. Original approach

U-net is a convolutional network architecture developed to produce precise semantic segmentation. The state of the art model performed pixel-wise classification upon training on biomedical images, that are usually scarcely available. Thus we have opted for it since we were provided only 100 training images and the task at hand can be performed by a classification of each pixel as road or background.

The architecture we adopted is inspired by the one created by Ronneberg et al. [5]. As depicted in *Figure 2*, the architecture resembles a U shape. The input image is fed to the network where the left side consists of a contraction downwards path. This path comprises a series of convolutions blocks where each block engulfs two 3x3 convolutions that increases the number of feature channels followed by non linear activation function (Relu). After each convolution block, a 2x2 Maxpooling operation is performed that further doubles the number of feature channels. As a result, the contraction network extracts the spatial features. The following expansion path aims at increasing the spatial resolution. This path performs deconvolutions that divide the feature channels by a factor of 2, followed by concatenation

with a feature map from the contraction path. Then a convolution block is applied. This process is repeated 4 times until a final 1x1 convolution with Sigmoid activation function is applied to yield a high resolution segmentation image. We used additional methods:

- **Dropout:** Dropout is applied after convolutions to reduce overfitting. Applying dropout also speeds up the training process. We set its rate at 0.2.
- **Batch normalisation:** Added in convolution blocks. Used for normalising the output of each activation layer which helps increasing the stability of the network.
- **Padding:** padded convolutions enables us to keep the same size of images.

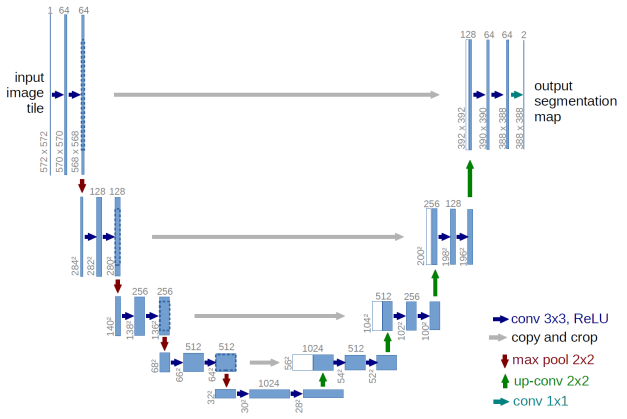


Fig. 2: Architecture of the U-net neural network.

B. Dilated bottleneck

This variant of U-net has the same downwards and upwards path but instead of applying a convolution block at the lowest level layer (bottleneck), series of dilated (atrous) convolutions with increasing dilation rate is carried out (1,2,4,8,16,32). Similar to classic convolutions, dilated convolutions apply kernels on the feature map, but on non contiguous points. Each point where the filter is applied is separated from the others by a distance called rate. This allows to extract different levels of semantic feature maps and gather more spatial information about the non immediate surroundings of pixels. In our case, this can be useful to precisely segment roads that are covered.

C. Predictions

Since we have to predict ground truths of a set of 608x608 sized images which are bigger than the training set images of size 400x400. We tried two methods.
First, we carried out mirror padding the training images to augment their size to 608x608 and feeding them to the model. Second, training the model on 400x400 images then divide each test image into 4 400x400 patches over which we make individual predictions. Then, we merge them all to obtain a final 608x608 prediction. The latter performed better on our initial tests and thus we decided to stick with it.

IV. Data Augmentation

Data augmentation is a recurring trick used in image classification and segmentation tasks as it allows the model to be trained on datasets multiple times their original size while starting with the same data. Data augmentations uses transformations such as rotations or rolling in order to present to the model images that, despite coming from the same original, are considered as separate data points and hence offer distinct learning outcomes. In this section we will first explore the concept of random data augmentation, then study the dataset in order to establish a baseline to work from. Then we will list and discuss some of the transformation we implemented in order to augment our dataset.

A. RandAugment

Data augmentation is a difficult task to tackle as it requires expertise and manual work in order to find the proper augmentation policies for each particular learning task. Learning policies have recently emerged as a way to automate the setup of augmentation policies, as it bypasses the previously mentioned requirements while crafting the perfect augmentation policy [4]. However learned data augmentation policy still can be rather prohibitive due to the large computational requirement and the need to build two separate models for the same task. We first tried a policy based on the work of Cubuk et al. [3], who presented a stochastic approach to data augmentation. Random Augmentation, or RandAugment, is presented as a compromise between learned augmentation and research for augmentation policies, using randomness to increase accuracy while having an automated augmentation, without spending much time fine-tuning the parameters of the data augmentation. RandAugment only has two hyper-parameters being N the number of successive transformation on a single image, and M the magnitude of the augmentation, driving the value of each transformation in a given interval. Since the expertise needed for augmentation of our dataset is not too demanding, we decided to keep only a selected list of transformations as follows:

- Identity
- Flip
- Rotate
- Contrast
- AutoContrast
- Roll
- Add Noise
- Brightness
- Sharpness

We got satisfying results but wanted to dig deeper into the data in order to find the proper augmentation.

B. Data exploration and a more suitable augmentation

As shown in the example of a satellite image and its ground truth in *Figure 3*, finding the roads among the image is not a simple task as some of the road can be hidden by trees, driveways, vehicles, or house roofs can be seen as

roads. In addition to that, roads are not always straight and in the same direction (horizontal/vertical for example). On the other hand, the test set consists of 50 images which differ from the training images, with a size of 608x608 pixels.

Fig. 3: A satellite image on the left and its ground-truth on the right.

An easy answer to the redundancy of the road orientation (being mostly horizontal or vertical) is the rotation. We applied rotations with different angles to our images, as seen in *Figure fig:rotations*. The rotations were done by an angle in {25, 45, 90, 180, 270}, to try and prevent overfitting to particular angles of the roads, knowing the redundancy mentioned earlier. Flipping will also prevent redundancy by providing a larger span of possible image combined with rotations.

V. Results

Model	Data Aug	Dilated Layer	F-1 (%)
FractalNet	X		75.7
U-net Litchi	X	X	77.3
U-net Lime	R	X	85.9
U-net Kiwi	N	X	87.5
U-net Melon	R	✓	87.6
U-net Banana	N	✓	88.6

TABLE I: Accuracy yielded by the different models depending on the use of data augmentation (Data Aug); dilation of the layer (dilated Layer) in the case of U-net. R corresponds to random augmentation. N corresponds to normal augmentation.

Initially, we used a modified version of the fractal model described in the paper of Gustav Larsson et al. [6] in order to use our model for image segmentation instead of classification as originally intended. Unlike U-net, this model make patch-wise prediction instead of pixel-wise prediction. The results shown in Table 1 for the fractal model are obtained after training of the model with data augmentation detailed in IV-A. However, we thought we could achieve better results with state of the art model for image segmentation: U-net. In addition, it allowed for comparison of the results between the two models.

U-net Litchi is used as a baseline U-net, without augmentation or dilated layers. The biggest improvement we saw was with the way we augmented the data. Despite being an innovative approach, random augmentation was not well suited for our segmentation task, perhaps being more suitable for image classification. We found that noise brightness and contrast changes were not significant, and in fact gave worse results than simple rotations and flips. We believe that the rotations tackling the redundancy of the road angles is the most decisive transformation in our augmentation, as having only these showed much better results than random augmentation with an extensive policy. Introducing dilated convolutional layers also had a considerable impact on the F-1 score (we observed about +1.5%). Early stopping was used to prevent overfitting when our data stopped increasing and reached a plateau. We found that this plateau was located towards epoch **50**. We can see this plateau on *Figure 4*, on which the loss stops decreasing.

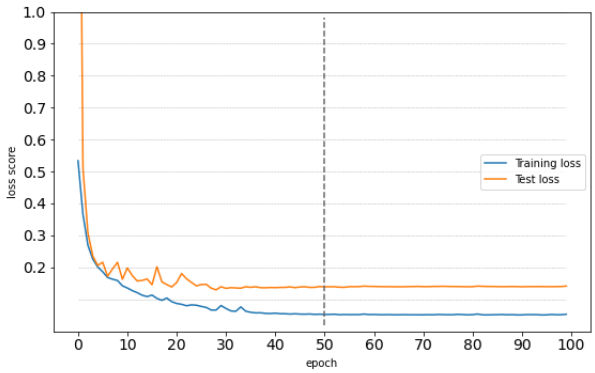
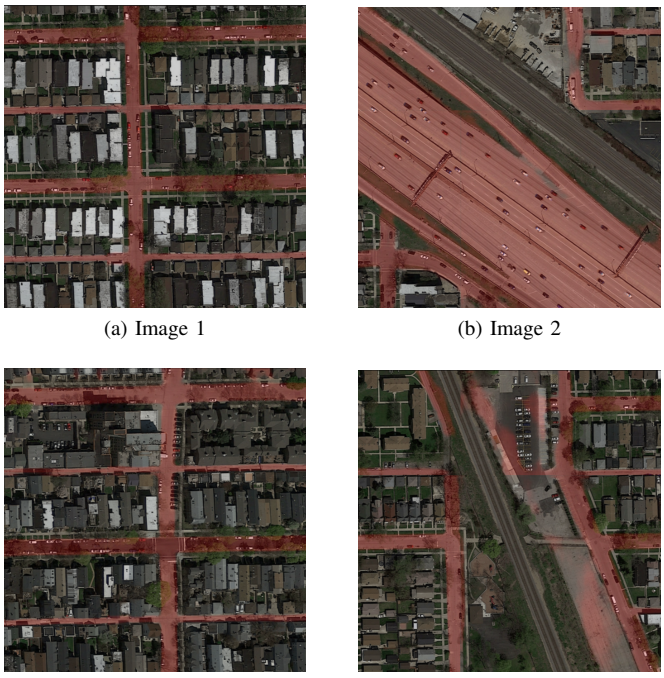


Fig. 4: Training and test loss during training of U-net Banana.

In image *5*, one can observe the predictions, with the help of a red overlay, made with U-net Banana. Those figures allows to compare the performance of our model on different configurations. We can see that the model has no issue predicting on vertical and horizontal roads as depicted in *Figure 5a*. The *Figure 5c* shows the the interest of the dilated layers, as roads hidden by trees are recognized nonetheless. Last, although the model correctly predict more complicated samples (*Figure 5b*), where regions of the image resembles to roads, its predictions are worse at times as in *Figure 5d*.

VI. Discussion

Since the fractal model make patch-wise prediction, we rather expected that the results may be worse than with U-net, which make pixel-wise predictions. Yet the results yielded with the fractal net are quite similar to the baseline U-net model Litchi, and could potentially lead to better result with proper data augmentation and fine-tuning of the model. However, since training with FractalNet on the original dataset already took 1 hour with a Google Colab GPU, we decided to continue with U-net, which required much less training time.



(a) Image 1 (b) Image 2

(c) Image 3 (d) Image 4

Fig. 5: Four test images with their prediction on overlay.

According to the results, the non random augmentation performed better than the random one. We conclude that random augmentation performs well with image classification as it adds noise, brightness and contrast. However, for our specific task, the satellite images to make predictions on, have a rather similar aspect to the training images but present different road angles and setting. Data augmentation that accounts for this variability by doing rotations and flips resulted in the model performing better. Using a dilated U-net enhanced the predictions and performed better than the classic U-net. It succeeds to capture more context on the surrounding of points in images and thus correctly predicts roads in challenging situations like for example being slightly hidden by a tree.

Since U-net is a state-of-the-art model in terms of semantic segmentation, we wanted to try other techniques, and did it with FractalNet, which turned out to have a heavy computational cost. Following the same idea, VGG16 supplemented with a classification method could be an alternative since it has shown to be an effective pre-trained model. The convolutional model VGG16 would be used here as a feature extractor, located upstream of a Random Forest for classification [1], sorting each pixel either to road or background. For the same reason we switched to U-net, VGG16 is a highly computationally costly model compared to U-net, and was thus not considered for this particular project.

VII. Conclusion and possible improvements

For future considerations, we are looking forward to explore more data augmentation techniques to perfect the model and improve the precision of predictions. Despite U-net being perfectly suited for image segmentation, we implemented state of the art augmentation techniques IV-A, which turned out to be not as effective as expected due to the nature of the segmentation task. Some augmentations such as noise and contrast were most likely suited for classification. This showcases the boundary between segmentation and classification, that we tried to cross through the implementations of a model suited originally for classification. Unfortunately, FractalNet could not compete with U-net, without mentioning the computation time. VGG-16 coupled with a Random Forest classifier uses the same idea, which could be an idea for a future semantic segmentation project. Using classification models for semantic segmentation is challenging, but shows the versatility of convolutional neural networks and the power they have to surpass human abilities with proper training.

Acknowledgements

We, as a team, thank you for reading our report. It was a difficult project as we only began two weeks before the deadline. We were on a different project at first that shown itself to be non-feasible in the time limit given of one month.

References

[1] D. M. S. Arsa and A. A. N. H. Susila. *VGG16 in Batik Classification based on Random Forest*. 2019 International Conference on Information Management and Technology (ICIMTech), Jakarta/Bali, Indonesia, 2019, pp. 295-299, doi: 10.1109/ICIMTech.2019.8843844.

[2] Piao, Shengyuan, et Jiaming Liu. *Accuracy Improvement of UNet Based on Dilated Convolution*. Journal of Physics: Conference Series 1345 (November 2019): 052066.

[3] Cubuk, Ekin D., Barret Zoph, Jonathon Shlens, et Quoc V. Le. *Randaugment: Practical automated data augmentation with a reduced search space*. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 3008-17. Seattle, WA, USA: IEEE, 2020. doi: 10.1109/CVPRW50498.2020.00359

[4] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. *Autoaugment: Learning augmentationpolicies from data.* arXiv preprint arXiv:1805.09501, 2018

[5] Olaf Ronneberger, Philipp Fischer, Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. Computer Vision and Pattern Recognition (May 2015).

[6] Gustav Larsson, Michael Maire, Gregory Shakhnarovich. *FractalNet: Ultra-Deep Neural Networks without Residuals*. Computer Vision and Pattern Recognition (2017).