Project 2 : Road Segmentation

Ahmed Ben Romdhane
Axel Dinh Van Chi
Jasmine Nguyen-Duc
Machine Learning course, EPFL, Switzerland

***Abstract*—In this work, the aim is to identify roads in satellite images. The two neural network architectures UNet and ResNet, inspired by state-of-the-arts, are compared when training the model for this road segmentation task. Because the ratio for roads is significantly smaller than that of non-road objects in the acquired images, the F1 score defines the efficiency of the model. To optimize this score, data augmentation, as well as hyper-parameter tuning are performed over the different models, to finally achieve a F1-score of 90.8% on the test set.**

I. Introduction

Semantic image segmentation enables the labelling of each pixel of an image. Here, the aim is to detect roads, so the label values for each pixel are either 1 for roads or 0 for everything else, which defines the background. Road detection can be useful when studying traffic management, city planning, and road monitoring. The efficiency of two different optimized convolutional neural network (CNNs) architectures, UNet and ResNet, are compared for this specific task. The effectiveness of the model was evaluated based on the F1 score metrics, which is defined as a harmonic mean between precision and recall and takes values between 0 and 1. This is done as to better take into account the heavily unbalanced ratio between the roads and non-road class labels. To obtain optimal results, data augmentation is performed with rotation and cropping. Important hyper-parameters, such as the learning rate and the weight decay, are tuned to acquire a more precise prediction. Moreover, various schedulers and optimizers are compared once the best architecture is established with respect to the F1 score. Finally, adequate post-processing is attempted after the prediction is made to smooth the pixels of the output image.

II. Models and Methods

A. Neural Networks as models

1) Convolutional Neural Networks for segmentation: The field of computer vision and object detection draws heavily on deep learning and the use of deep CNNs in particular. Over time, many successful models derived from this architecture have demonstrated their effectiveness for pixel classification problems such as segmentation [1]. Among these are the UNets and Resnets, which we have explored in this report. The main advantage of these models over more standard machine learning approaches is their ability to learn the most appropriate features for the problem, eliminating the need for manually crafted features, something that generally requires a great deal of expertise in the field [1].

2) UNets: Initially developed for biomedical segmentation purposes by Ronneberger et al. [2], the UNet quickly established itself as a reference model for segmentation. The architecture of the final model used in this project is illustrated in Figure 1 and is largely inspired by the original UNet model [2]. It is indeed composed of two symmetric paths: the contractive and the expansive paths. Similar to a convolutional network, the contracting path aims at creating feature maps while gradually reducing spatial information. Each step of this path consists in applying two consecutive 3 x 3 convolutions (one-padded here to keep the image dimensions constant), each followed by a rectifying Linear Unit (ReLU) function and a max pooling operation for down-sampling (2 x 2 window with stride 2). The expansive block ultimately restores the original size of the image from the feature maps. It combines the features obtained from the contractive layer with spatial information in order to achieve a precise segmentation. This is done through a serial application of up-sampling (via a so-called ”up-convolution” operation which is a 2 x 2 convolution) that halves the number of channels, followed by a concatenation with the corresponding part in the contractive bloc and a 3x3 convolution then ReLU applied twice. Finally, a 1x1 convolutional layer followed by a sigmoid maps each original pixel to its probability of being a road. Moreover, we apply Batch Normalization after each of the 3 x 3 convolutions and before applying the ReLU function. Essentially, this operation consists in normalizing the layers inputs and was shown to reduce internal covariate shift, as well as increasing the training speed and acting as a regularizer [3]. Lastly, following the model described by Silburt et al. [4], a dropout layer (with probability $p = 0.5$) is added in the expansive path, in order to avoid over-fitting (Figure 1).

3) ResNets: The use of deep neural networks can sometimes lead to higher training errors and a degradation problem may occur. The ResNet is a popular neural network which consists of a series of stacked residual units and is often chosen to address this issue. It has a similar backbone to that of the UNet, but each unit has an additional skip connection. These skip connections are known to facilitate information propagation through the addition of the identity mapping of the input with output at the end of each unit. Two ResNets models are tested and optimized for the segmentation task, with and without a dropout step implemented in the middle of each unit.

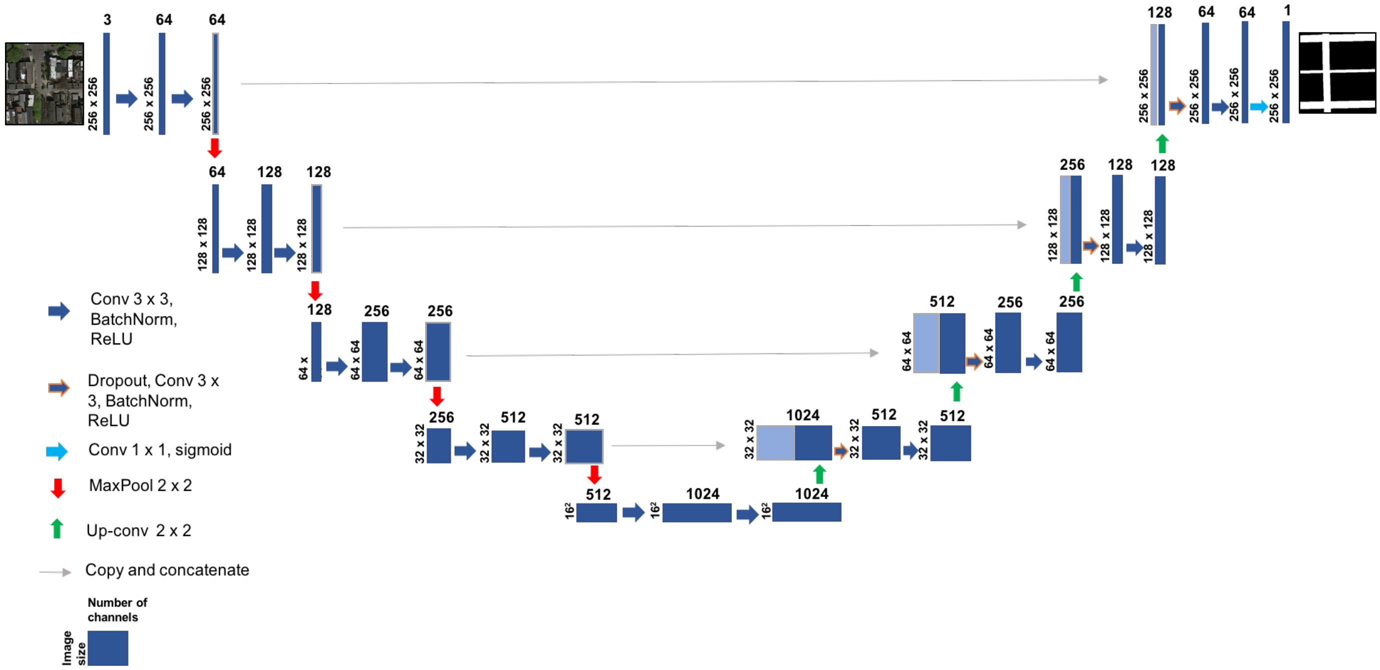

Fig. 1. Architecture of the final UNet model used to obtain the best F1-score.

B. Pre-processing

1) Exploration of the dataset: The provided training dataset consists of 100 coloured satellite images that are associated with their corresponding ground truth (**GT**) masks. While the pixels of the masks are required to have the binary values of 1 or 0 depending on whether or not they are associated to a road (labeled pixels), this is not the case for the given ground truths masks. Hence, the pixel values that exceed the chosen threshold of 0.3 times the maximum pixel value are changed to 1 and the rest are set to 0. Once the masks are modified to have only binomial values, the data is split into three groups : a training set, a validation set and a test set. The ratios for these groups are 0.8, 0.1 and 0.1 respectively. The training set is used to train the model, the validation set gives insight on the performance of the predictions throughout the epochs and the local test evaluates the model performance on unseen data. Finally, the final F1 score is attributed based on the results obtained on the 50 additional images provided without any given mask.

2) Augmentations: Data augmentation is the key to increasing the size of the training data, which is essential to improve the performances of the model during the training step. In this work, each image was cropped into five 256 x 256 sized images. These would correspond to the four corners plus the centre and offer 5 new images per initial one to feed the model. Moreover, the original image would then be rotated at the chosen degrees of 90, -90, 45 and -45, before extracting a centre crop of size 256 x 256. Hence, starting with 100 training images of size 400x400, we end up, after the augmentation, with 9 new resized images derived from each of the original image in the train set. In total, 900 images of size 256 x 256 are produced from this procedure.

C. Training

Training a deep neural network is difficult because of the many parameters to consider. This is a time consuming and computationally demanding process. Moreover, the optimization is challenging because the problem to be optimized is not convex and is very likely to contain several minima. The training was carried out using the free GPU provided by google collab, which drastically improves the training speed. Here, we discuss the choices made for the different parameters.

1) The models: Four models have been explored throughout this report. In the following sections we will denote them as follows:

- **UNetDOBN**: UNet described in section A.2 and shown in Figure 1, which corresponds to our final model.
- **UNetBN**: UNet with batch normalization, without dropout.
- **ResNetDOBN**: ResNet with batch normalization and dropout.
- **ResNetBN**: ResNet with batch normalization, without dropout.

Each of the models was subsequently optimized with a different loss function, in order to finally select the one with the best score evaluated on the validation set.

2) Loss functions: These are the three loss functions that were used for the optimization of the neural nets:

- Binary Cross Entropy (BCE) loss: Widely used in the context of binary classification (pixel-wise here), this

quantity gives a measure of the difference between the true distribution of the data and that predicted by the model. It is therefore a question of minimizing this difference.
- Dice Loss: Another popular metrics for segmentation tasks based on the Dice coefficient, which provides a measure for the similarity between two sample images.
- BCE-Dice loss: A combination of the two described losses that provides the stability of the BCE to the standard Dice loss.

3) Activation function: The selected activation function is the rectified linear activation function (ReLU), a piecewise linear function that returns the input directly if it is positive, otherwise zero. It has become the default activation function for many types of neural networks. It is favoured to sigmoid and hyperbolic tangent, even though they are also very commonly used. The main problem with the latter is their saturation for very high and low values, as these functions have an S-like shape. ReLU is thus the preferred activation function for this deep neural network [5].

4) Optimizers: The chosen optimizer is Adam (Adaptive momentum estimation) as it is often favoured over other stochastic methods in many studies. It is said to combine the advantages of AdaGrad and RMSProp which are extensions of stochastic gradient descent. Also, this method computes individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients. [6]

5) Schedulers: Schedulers are beneficial as they modify the value of the learning rate throughout the epochs. Two different schedulers are used in this work :

- Reduce On Plateau: As the name suggests, when the loss ceases to decrease for a certain amount of epochs (patience threshold), the learning rate is reduced by a chosen factor. The patience is set to 10 and the factor is 0.5.
- Step LR: The learning rate is reduced by a certain factor after a fixed number of epochs. Here, it is divided in half every 20 epochs.

6) Learning rate tuning: As described in the beginning of part C, the landscape of the loss function is expected to have several local minimas and, as such, the learning rate is perhaps the most important parameter to tune for deep learning models: if too high, there is a risk of overshooting the minimum, the opposite case, convergence can be extremely slow. A successful method consists in varying the learning rate at each mini batch ($e.g.$ exponentially) between two extremums values (for example 10^{-7} and 10) over one epoch [7]. Subsequently plotting the losses according to the learning rates most often results in a curve very similar to Figure 2. A good pick is the learning rate that offers the greatest decrease in loss ($i.e.$ steepest gradient) but which remains lower than the learning rate corresponding to the minimum.

7) Weight decay: The Adam optimizer allows the decay of the weights as an option. Since Adam is used instead of Vanilla SGD, one must keep in mind that the weight decay does not correspond to the L2 regularization. This method was used with a weight decay value of 10^{-3}, however this did not seem to significantly improve the F1 scores.

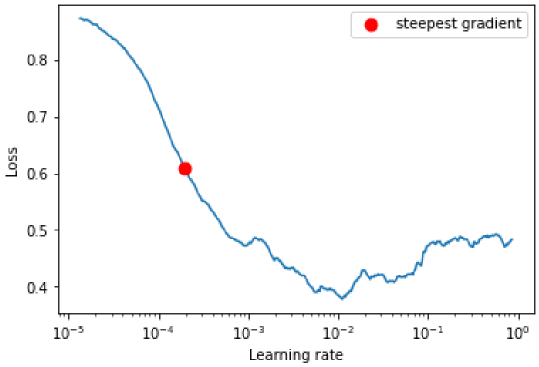

Fig. 2. Loss with respect to the learning rate. The red dot specifies the learning corresponding to the steepest change in loss.

III. Results

Table I shows the F1 scores obtained on the local test set (see $B.1$) for the different models described in $C.1$ ($e.g.$ UNet BN + DO refers to **UNetDOBN**). The scheduler ReduceOnPlateau is applied to obtain these results.

		UNet		ResNet	
		BN+ DO	BN	BN + DO	BN
Dice Loss	LR	9.6e-5	9.6e-5	3.6e-5	3.6e-5
	F1 score	0.9096	0.9085	0.853	0.853
BCE Loss + Dice Loss	LR	1e-4	5e-5	1e-4	5e-5
	F1 score	0.9090	0.8957	0.8773	0.8449
BCE Loss	LR	4.75e-05	4.75e-05	1.6e-05	3.25e-05
	F1 score	0.9044	0.8896	0.8527	0.8354

TABLE I
F1 score and learning rates (LR) for the various loss functions and network architectures considered

Table II shows the F1 score results when the patch size used for the data augmentation is increased from 256 x 256 to 320 x 320. Both the local test set (coming from the initial training set) and the provided test set are predicted on. Only the best network (**UNetDOBN**) and two best loss functions (BCE-Dice loss and Dice Loss) are selected. Step LR is the scheduler used to obtain these results.

Loss function	Network	F1 score on local test set	F1 score on the provided test
BCE Loss + Dice Loss	UNet BN + DO	0.967	0.908
Dice Loss	UNet BN + DO	0.925	0.908

TABLE II
F1 score and learning rates (LR) for the two best loss functions and **UNetDOBN** when the patch size is increased to 320 x 320

IV. Discussion

The UNet network turned out to be more efficient than the ResNet network for the segmentation task. When observing

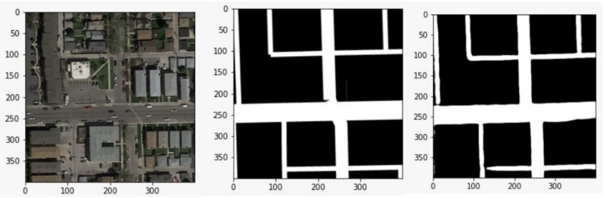

Fig. 3. Satellite image number 99 from the training set (left) with its ground truth (middle). The image (right) shows the prediction made by our best model (UNet BN + DO)

the validation loss throughout the training of the ResNet model, one can notice that the curve behaves like a parabolic function. The loss decreases exponentially, reaches a plateau and increases exponentially again. This is a signature of overfitting, which is an issue because the model adapts too well to features that are only present in the training set. The implementation of a Dropout step in the core of each residual unit seems to improve the results, which is consistent with previous studies that suggest that Dropout inhibits overfitting.

Regarding the loss functions, the F1 score reached with Dice Loss seems to overtake that of the BCE Loss and the BCE Loss + Dice Loss when the performance is assessed on the test images obtained by the splitting of the training set. However, when predicting on the provided test set, the score corresponding to the BCE Loss + Dice Loss is the highest.

The networks tested give very satisfactory F1 score results on the test set arising from the splitting of the provided training set. The performance decreases however when predictions are made on the given test set. It is provided separately from the training set and consists in satellite images that have sizes of 608x608. This means that the predictions are made on pictures that are bigger than the ones the model trained on.

To improve the predictions on bigger images, the cutting of the test set was attempted. Each picture could be divided into four smaller images (304x304) that would be predicted on separately. After the predictions are made, the four images would be put together again to reform the 608x608 sized image. This method did not perform well, and even decreased the performance accuracy. The issue was that the model would lose the general context of the picture, and the transition from one subimage to another could be discontinuous. Another strategy to improve the predictions is to modify the size of the patch size used for the cropping during the feature augmentation. Logically, an increased patch size should improve the model, but there is a tradeoff to take into account with respect to the similarity between the patched images when the size is too big. The patch size is thus modified to 320x320 to train the final model (versus 256x256 used previously) and the two best loss functions (Dice/BCE Loss and Dice Loss) combined with the best network (**UNetDOBN**) are selected. The results show a significant improvement on the test sets for both loss functions (see table II).

V. Conclusion

In summary, the UNet has been proven a powerful convolutionnal neural network for the semantic segmentation of roads from satellite images. Our ResNet architecture does not perform as efficiently, even though the results are also very decent. Given initially 100 images (of dimension 400x400) to train on, the model was able to predict with a F1 score of 90.8% on 50 images (of dimension 608x608). To obtain such a model, the images of the training set were first cropped and rotated so that we end up with 900 images (of dimension 320x320) in total. Adam is combined with different scheduler techniques to optimize the model. The sum of the Dice Loss and BCE Loss as well as the Dice Loss alone both lead to the same F1 score when predictions are made on the provided test set.

References

[1] F. Sultana, A. Sufian, and P. Dutta, “Evolution of Image Segmentation using Deep Convolutional Neural Network: A Survey,” *Knowledge-Based Systems*, vol. 201-202, p. 106062, Aug. 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950705120303464

[2] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” *arXiv:1505.04597 [cs]*, May 2015, arXiv: 1505.04597. [Online]. Available: http://arxiv.org/abs/1505.04597

[3] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” *arXiv:1502.03167 [cs]*, Mar. 2015, arXiv: 1502.03167. [Online]. Available: http://arxiv.org/abs/1502.03167

[4] A. Silburt, M. Ali-Dib, C. Zhu, A. Jackson, D. Valencia, Y. Kissin, D. Tamayo, and K. Menou, “Lunar Crater Identification via Deep Learning,” *Icarus*, vol. 317, Mar. 2018.

[5] J. Brownleer, “A gentle introduction to the rectified linear unit (relu),” https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/, January 2019, accessed on 17-12-2020.

[6] ——, “Gentle introduction to the adam optimization algorithm for deep learning,” https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/, July 2017, accessed on 17-12-2020.

[7] S. Gugger, “How do you find a good learning rate,” https://sgugger.github.io/how-do-you-find-a-good-learning-rate.html, March 2018, accessed on 17-12-2020.