Aerial image Road Segmentation Task using U-net variants

Mathieu Verest, Romain Frossard, Jean Ciardo
Department of Computer Science, EPFL, Switzerland

***Abstract*—The aim of this project is to concieve a classifier capable of discerning the roads from background elements using a small dataset of satellite images originating from Google maps. Both standard convolutional neural networks and U-net variants were implemented and tested. Our best result was achieved using an Attention U-net with extended data augmentation and hyperparameter tuning yielding a F1 score of 0.85 on a hidden test set from AIcrowd.**

I. Introduction

Image segmentation is a typical but important task in computer vision performed in a wide range of contexts, from medical imaging, to the task at hand : road segmentation, which has never been so topical in view of recent increase in use of efficient and accurate understanding in visual data in general.

The goal of this project is to develop a classifier capable of segmenting roads in satellite or aerial images by assigning each pixel a binary label (road = 1, background = 0).

We present different methods to tackle this problem, all based on different architectures of Convolutional Neural Networks(CNNs) which represent the state of the art[1], with data processing, augmentation and extensive hyperparameters tuning.

A. Task Specificity

The variant of the task at hand is a peculiar one. The evaluation on AIcrowd of a such a small set of images immediately brings the question of overfitting. This concern is further reinforced by the fact that the evaluation set is openly part of an available dataset quickly found when looking to augment the provided data set with existing elements. These considerations take us from any resemblance to a real case study. We therefore decide to focus our efforts in trying to optimize the F1 score without natural data augmentation and to use the CITY-OSM data set of Chicago[2] as an additional remark.

II. Baseline Model

As a baseline, to quantify further improvements, we used a simple CNN along with a more modern U-net architecture[3]. Both models were trained until overfitting with only channel normalization as processing and using the standard Binary Cross Entropy loss[4]. Using 85 images for training and 15 for validation, the vanilla CNN produced a local F1 score of 0.69 against 0.84 for the U-net. We therefore explore U-net architectures going further.

III. Data Manipulation

A. Data Exploration

The initial dataset to perform this task is composed of 100 RGB GoogleMaps-derived satellite images of 400×400 pixels primarily depicting urban environments. Each image is paired with a binary mask that serves as ground truth, where the road segments are outlined. Additionally, a test set of 50 images with dimensions of 608×608 with unknown ground truth was provided. Predictions on this test set are evaluated on the AI Crowd platform, enabling a competitive benchmarking of model performance. Training datapoints are $\sim$80% background and $\sim$20% roads, both training and test images have low geographical diversity ; appart from the empirical observations regarding the diversity of architectures, roads, aqueous and green spaces, they both consist of relatively dark images with mean RGB intensity : $\sim$30% with STD of $\sim$18% and a mean difference of $\sim$8.33e-4 across channels, which could provide an argument to support that both come from the same underlying distribution.We can see in Figure 1 that the groundtruth can be hidden by trees, train tracks or even building, which assumes the model to be complex enough for being able to handle such cases.

Fig. 1: *Left*: Satellite image from training set. *Center*: Its associated CLAHE enhancement. *Right*: Grountruth.

B. Data Processing

As stated above, the images are quite dark. To address this problem, we used contrast-limited adaptive histogram equalization (CLAHE)[5] to enhance the constrast and brightness of the images. As stated in the baseline, we also normalized the distribution across the RGB channels, to ensure that not one channel dominates the gradients[6]. On the standard U-net, from the F1 baseline score of 0.84, training similarly yielded a score of 0.86.

C. Data Augmentation

Traditionally, deep learning models (including CNNs) perform well on large datasets[7], even if the evaluated task (c.f. AICrowd) does not presuppose generalization being the main objective, a training set of only 100 images is an issue. To address this, we use synthetic data augmentation through different mechanisms. First, we add images through horizontal and vertical flips, who all are subject to rotations of $\frac{\pi}{2}, \pi$ or $\frac{3\pi}{2}$, to avoid creating images too similar to those belonging to the training set.

While taking into account the numerical superiority of straight over oblique roads (a quick survey indicated $\sim$90%), we are also aware that under-represented samples in training, like uniquely oriented roads, might be poorly recognized by the model if this imbalanced ratio is retained. To address this, we also introduce random rotations into n windows between 25°and 90°(excluding angles that would produce images too close to the original). The images rotated this way have additional 40% chances of being flipped horizontally and vertically, to ensure a maximum of diversity in samples.

Fig. 2: Description of the way that padding is implemented to ensure consistent 400x400 dimensions after rotattion.

Moreover to preserve the 400x400 pixels dimensions with rotations that are not multiples of $\frac{\pi}{2}$, we use reflected mirror padding (c.f. Figure 2) to ensure the empty space created inside the frame is filled. These implementation enable to augment and modify the distributions of certain image types at will.

IV. Model Selection

A. Metrics and Performance Evaluation

The performance of the models was assessed on a randomly selected validation set, consisting of images with known ground-truth mask labels consisting of 15% of the training set. The evaluation was based on the F1 score and accuracy, defined as follows:

$$F1 = 2 \cdot \frac{\mathrm{TP}}{2 \cdot \mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \qquad \text{Accuracy} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN}}$$

The imposed evaluation format required decomposing the ground truth and predictions masks into 16×16 patches. For each patch, if more than 25% of the pixels are classified as positive, the entire patch is considered positive. Consequently, these metrics were not computed at the pixel level but instead at the patch level. The threshold applied to the probability map output by the model was chosen to maximize the F1 score on the validation set. Additionally, the final predictions were submitted to AIcrowd for evaluation on a separate test set. The evaluation followed the same patch-based F1 score computation.

B. U-Net and Attention U-Net Architectures

As presented in the U-net parper[3], this architecture possesses key features that makes it efficient for a semantic segmentation task. The **contractive** path reduces spatial dimensions while encoding context and increasing the feature maps while the **expanding** path combine precise informations with broader leves of context by the successive up-convolution and incorporating the **skip-connections**.

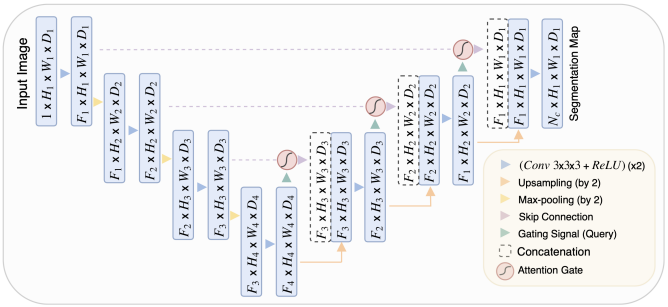

Fig. 3: Attention U-net architecture as described in Oktay et al. (2018). [8].

For segmentation task, the processing of the context by the network is key to generalization. To further explore this we also implemented an Attention U-net[8] as seen in Figure 3. This architecture includes **attention gates** at the concatenation from up and skip connections that focus on important features while suppressing the ones viewed as irrelevant, while only augmenting the memory cost minimally. We tested more complex variants such as a Recurrent Residual U-net (R2U-Net) that introduces recurrent convolutions, but as seen in Figure 6, the model performed poorly, overfitting instantly.

We performed a preliminary evaluation of the models using preprocessing and a full synthetic augmentation of the dataset.

	F1 Score	**Accuracy**
Standard Cnn	0.691	0.802
Standard U-Net	0.892	0.930
Attention U-Net	**0.911**	**0.941**
R2U-Net	0.592	0.77

Fig. 4: Comparison of U-Net variants with CLAHE, channel normalization and full dataset augmentation (parameters (2,2), c.f. train.py) .

V. Hyper Parameters

Even if among the contemporary deep learning architectures, U-nets are not considered heavyweight, as often, their size prohibited us from performing a full scale cross validation to tune all possible hyper parameters. But we considered a number of them :

A. Learning Rate scheduler

We experimented with several scheduler like ReduceLROnPlateau which reduces the learning rate when the validation loss cease to decrease, StepLR which deacreases the learning rate each fixed number of steps or even OneCycleLR which introduces a *warming up* phase, which can increase training stability[9].

B. Batch size

This parameter was constrained by computational limitations. Even on training in Google colab on a A100 GPU (40 GB of RAM), U-Net were limited to a size of 32, and Attention U-net variants were limited to a maximum of 16, sometimes 8. For comparison sake the models were trained on batches of size 8.

C. Regularization

We tested three types of regularization to try to mitigate overfitting ; Early stopping, which stopped the training if the validation loss stopped improving, Batch normalization and Dropout. Frome the baseline of Attention U-net with a F1 score of 0.911, removing the batch normalization made the models performed significantly worse, while Dropouts did not give signitificant improvement with an average F1 score of 0.908 (with parameters from 0.2 to 0.4), while weight decay also dragged the model down with an average F1 score of 0.902 (with score set from 1e-4 to 1e-5).

D. Patches

Since giving small patches to the models gave us systematically worse performances (c.f. Fig. 5), to address the fact that the images of the test set are of dimensions of 608x608 pixels while the training only of 400x400, we elected to give the model either directly 608 by 608 images, or to give it the same size of the training but to use several predictions to reconstruct the images, averaging where the patches overlap. Even with this technique, it is clear by the discrepancies of the local results and those of AIcrowd that giving directly 608 by 608 to the model is better.

Patch Size	Computation Method	Local F1 Score	AI Crowd F1 Score
400x400	Patches with averaging	0.911	0.692
608x608	Local (on full image)	–	**0.841**

Fig. 5: Example of discrepancies between the locally computed F1 score and the evaluation on AIcrowd. The model is the baseline of Attention U-net with BCE without augmentation.

E. Losse functions

Multiple loss functions exist in litterature[10] for image segmentation tasks, they can be crucial in helping the model correctly evaluating what are the elements to focus on while training. We the most used ones and tested them :

1) Binary Cross Entropy Loss: The one we used for our baslines, it is a measure of the dissimilarity between the predicted probability distribution and the true class label. It is often the default choixe in segmentation task.

$$\text{Loss } = -(y \log(p) + (1-y)\log(1-p))$$

2) Weighted Binary Cross Entropy Loss: It modifies the standard BCE loss by introducing class-specific weights, w^+ and w^-. This allows the loss function to penalize errors in underrepresented classes, reducing the effects of imbalanced classes.

$$L = -\frac{1}{N}\sum_{i=1}^{N}\left[w^+ y_i \log(\hat{y}_i) + w^-(1-y_i)\log(1-\hat{y}_i)\right]$$

3) Jaccard Loss: Synonym to the IoU metric, similar to the Dice, it is defined by the ration between the overlap of the positive occurrences between two sets and their mutual combined values.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

4) BCEDice with border penalization: This expression is a combination of the pixel-wise classification of the BCE and the imbalance handling of the Dice, which focues on the overlap between predicted and true masks. The first term being the cross entropy, the second one the Dice, handling the overlaps.

$$L = \frac{1}{N}\sum_{i=1}^{N} w_i \cdot \left[-y_i \log(\hat{y}_i) - (1-y_i)\log(1-\hat{y}_i)\right] + \frac{2\sum_{i=1}^{N} w_i y_i \hat{y}_i}{\sum_{i=1}^{N} w_i (y_i + \hat{y}_i)}$$

The weights emphasize boundary pixels by identifying regions near transitions (using average pooling) and assigning them higher weights, it ensures that the model prioritizes these singular areas. This boosts boundary accuracy in such tasks while maintaining balance across the entire image with weight normalization.

Loss Function	F1 Score	Accuracy
Binary Cross Entropy (BCE)	**0.911**	**0.94**
Weighted BCE	0.899	0.90
BCE + Penalize Border	0.9091	0.945
Jaccard Loss	**0.912**	**0.951**

Fig. 6: Evalutation of the different loss functions. The more complex ; WBCE & BCE Penalize Border are trained using a Warm Up learning rate to increase stability.

F. Optimizers

We tested three of the most common optimizers, SGD, Adam and AdamW. Over the baseline Attention U-net using the Jaccard loss we achieved scores of

G. Weights initialization

For the models working with attention, we implement two different weights initializations. The way the weights are initially set up can greatly affect the training regime in which your network will evolve in.

Kaiming initialization is tailored towards ReLU activations, it weights are drawn from a normal/uniform distribution with avariance v that depends on the number of input neuron of the layer.

The *Orthogonal initialization* uses weights that are orthogonal to each other. It tries to enforce diversity in the learned features and reduce correlation.

With a standard BCE, the *Kaimin* yielded a local F1 score of 0.913 against 0.890 for the *Orthogonal* and 0.911 without any.

VI. Results

Our best reproducible result on AIcrowd, a F1 score of 0.85, was achieved with the Attention U-net architecture described in Oktay et al. (2018). We used an initial learning rate of 2.1e-3, ReduceLROnPlateau as scheduler with patience 3 and a reduction factor of 0.9, using Adam as optimizer. We used the standard BCE loss to fine tune the model since it was easier to interpret during the process, the Jaccar being more prone to produce very polarized predictions.

A. The CITY-OSM Chicago dataset

While exploring the possibilities of natural dataset expansion, we found the source from where both the training and test set were coming from[2]. Even if it could be done quite easily, we decided not to overfit the evalutation set at all cost, having no other meaning to maximize and abstract grading criteria. We still performed a form of overfitting task, where we trained our best model on a mixture of randomly cropped and targeted cropped images, to see if the model was complex enough to recognize elements of the test set among other very similar samples, which it does. This gives obviously our best prediction on AIcrowd ; a F1 score of 0.892. Additionally, we compared to our best model the synthetic augmentation to the *true* one, additional samples coming from their natural distribution. It gave us Faire l'essai de et manière reproductible

VII. Conclusion & Discussion

In summary, our fine tuned Attention U-net gives us decent results given the small dataset provided. The U-net architecture gave a clear improvement over the standard CNN, as did the losses, while other parameters like most regularization (dropouts and weight decays) did not help us.

More complex models, wether it is variants of U-nets of the like of R2U-net or completely different architectures like Vision-transformers, might be able to capture more details or generalize better but are in general suited for datasets of larger magnitude ; 100 original data points is really small.

Additionally, artefacts in predictions such as isolated patches oustide any road context, specially over the test set, could suggest that post processing might be a lead to explore.

VIII. Ethical Risks

In this project (the same concerns extend to similar tasks) we evaluated that an ethical risk could arise with regard to **privacy**. Indeed the large scale use of satellite imagery in road segmentation can, as a side effect, record potentially sensitive or identifiable information about individuals or private structures in general, which pose a serious risk of violation of privacy. In the same vein, the fact that it was so easy to find the location of the provided set of images using a simple reverse search speaks volume. The main stakeholders that could be impacted are entities whose activities/properties are visible in the used imagery. The negative impact includes malevolent use of these data outside of the primary scope of the software such as profiling or surveillance. We consider this risk as **moderate** in severity for real-world use of similar software in view of the possibility of integrating strict data handling protocols that could greatly reduce the likelihood of occurrence of misuse. In the case or our own project, the risk is **low** both in severity and occurrence since OSM labels comply to the ODbL (Open Database Licence) and the Google maps imagery as already been downsampled with a factor of $\sim$25, which ensure that PII (personally identifiable information) are no longer identifiable.

Since we evaluated the risk to be low in the context of our project, we did not take this risk into account when training our model since at this level of detail and for the task at hand, anonymising the data (for example by blurring people, property markings of vehicles) would have not further reduced the risk.

References

[1] S. Hao, Y. Zhou, and Y. Guo, "A brief survey on semantic segmentation with deep learning," *Neurocomputing*, vol. 406, pp. 302–321, 2020.

[2] P. Kaiser, J. D. Wegner, A. Lucchi, M. Jaggi, T. Hofmann, and K. Schindler, "Learning aerial image segmentation from online maps," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 11, pp. 6054–6068, 2017.

[3] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *arXiv preprint arXiv:1505.04597*, 2015.

[4] U. Ruby and V. Yendapalli, "Binary cross entropy with deep learning technique for image classification," *Int. J. Adv. Trends Comput. Sci. Eng*, vol. 9, no. 10, 2020.

[5] W. Yussof, M. S. Hitam, E. A. Awalludin, and Z. Bachok, "Performing contrast limited adaptive histogram equalization technique on combined color models for underwater image enhancement," *International Journal of Interactive Digital Media*, vol. 1, no. 1, pp. 1–6, 2013.

[6] S. Chatterjee, D. Dey, and S. Munshi, "Chapter 2 - preprocessing and segmentation of skin lesion images," in *Recent Trends in Computer-Aided Diagnostic Systems for Skin Diseases* (S. Chatterjee, D. Dey, and S. Munshi, eds.), pp. 25–52, Academic Press, 2022.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[8] O. Oktay, J. Schlemper, L. L. Folgoc, M. C. H. Lee, M. P. Heinrich, K. Misawa, K. Mori, S. G. McDonagh, N. Y. Hammerla, B. Kainz, B. Glocker, and D. Rueckert, "Attention u-net: Learning where to look for the pancreas," *CoRR*, vol. abs/1804.03999, 2018.

[9] A. Gotmare, N. S. Keskar, C. Xiong, and R. Socher, "A closer look at deep learning heuristics: Learning rate restarts, warmup and distillation," *arXiv preprint arXiv:1810.13243*, 2018.

[10] J. Ma, J. Chen, M. Ng, R. Huang, Y. Li, C. Li, X. Yang, and A. L. Martel, "Loss odyssey in medical image segmentation," *Medical Image Analysis*, vol. 71, p. 102035, 2021.