CS-433 Machine learning

Project 2: Road segmentation

Edwin Bertschy, Tomas Valdivieso Damasio da Costa, Victor Pennacino
Department of Computer Science, EPFL, Switzerland

***Abstract*—In the field of computer vision, road segmentation holds significance in various applications such as automated driving and map generation. The versatility of segmentation techniques transcends transportation, impacting diverse domains like medical imaging and environmental monitoring, including applications like natural disaster monitoring. This study delves into the exploration of multiple neural network architectures for road segmentation. A comparative analysis of Dice and Focal Tversky losses is conducted, and techniques such as data augmentation and hyperparameter tuning with Bayesian optimization were employed. Remarkably, a pretrained U-Net model, trained with a Dice loss, demonstrated outstanding effectiveness by achieving a final F1 score of 91.3% in accurately delineating roads.**

I. Introduction

This project focuses on developing a deep learning classifier to segment roads from aerial images. The goal is to accurately label each patch of 16x16 pixels as either 'road' or 'non-road'. The dataset is made up of a training set with ground truth masks and a test set for validation. The model's performance will be evaluated using the F1 score, ensuring a balance between precision and recall. Note that the F1 score implemented in this work is derived from the cumulative count of true positives/negatives and false positives/negatives across all individual patches of the validation dataset. For clearer visualization, Figure 1 overlays true positives (green), false positives (red), and false negatives (yellow) per patch onto two aerial images from the validation dataset. True negatives have been left transparent for clearer presentation.

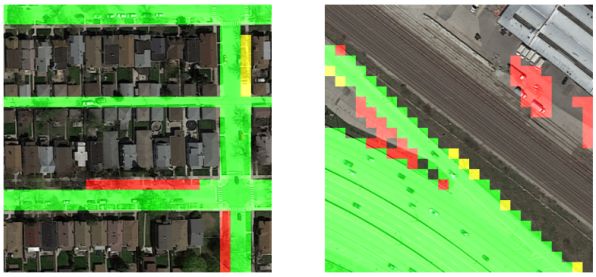

Fig. 1. Patch prediction on validation dataset after data augmentation

II. Project setup

For efficient model training, Google Colab was used for its GPU capabilities, crucial for processing our image segmentation tasks. In this project common Python modules were employed. Libraries like numpy and os, image processing modules such as matplotlib, cv2, PIL, and machine learning frameworks like torch and pandas. Alongside this, the Segmentation Models PyTorch[1] (SMP) library was employed for its high-level API, simplifying the creation of neural networks. Nine model architectures, including U-Net, all with pre-trained weights for accelerated convergence, are offered by SMP. The model development and training processes were further streamlined by its comprehensive collection of metrics and loss functions. For image augmentation, the Albumentations[2] library, known for implementing a diverse range of image transform operations for computer vision tasks, was used. Finally, Bayesian Optimization[3] was employed for hyperparameters tuning.

III. Data preprocessing

The provided dataset consisted of a training dataset and a test dataset. The training dataset comprised 100 colour aerial images, each with a size of 400x400 pixels, accompanied by their respective ground truth. The test dataset included 50 colour aerial images with a larger size of 608x608 pixels. By estimating the average resolution of multiple cars of training and test images, it was determined that the image resolution of the training and test datasets were identical. This estimation was done by measuring the average length of cars in pixels. Hence, the images in the test dataset covered a larger geographical area due to their larger size.

In the training dataset provided, the ground truth images were not strictly binary. The pixels had more than two distinct values, as can be seen in Figure 2. To create black and white image masks, a threshold of 120 (greyscale pixel intensity) was applied. In these masks, a pixel value of 255 denotes roads areas, while 0 indicates non-road areas. Finally, the size of the training images was increased to 416x416, as the models required an image size that was divisible by 32. Note that, for optimal training, the images were normalized using the previously described SMP library.

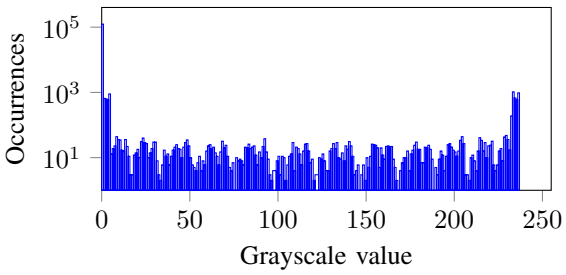

Fig. 2. Greyscale value distribution of a ground truth image

IV. Dataset splitting and early saving strategy

The dataset was structured into training and validation sets, adhering to an 80/20 split.

Representativity: A fixed seed was used to ensure reproducible results in the training/validation split. The test, training, and validation datasets were categorised to verify that the validation set was representative of the test set and that each image type was adequately represented in the training set (Table I).

Category	Neighbourhood	Complex road	Parking	Highway	Train track	Water
Train	75%	10%	35%	15%	20%	5%
Validation	82%	11%	34%	1.2%	11%	2.5%
Test	95%	26%	36%	2%	10%	2%

TABLE I

Complex roads, such as curved roads or roads with multiple connections, are underrepresented in both the training and validation sets. There are relatively few images of this type overall in our training set. This issue was addressed through data augmentation (please refer to subsection VI-A). Aside from this, the validation set appears quite representative, with a similar dataset composition to the test set. It was decided not to further modify the constitution of these sets.

To avoid overfitting, a validation-based 'early saving' strategy was adopted. This approach involved continuously monitoring the model's performance on the validation set throughout the training process. The model that exhibited the highest performance on the validation set was saved and chosen to predict the test set images.

V. The 4 starting models

Four model architectures were selected from the SMP library, and their performance was compared to choose the one most suited to our needs.

A. Model descriptions

U-Net is a popular convolutional neural network. Its architecture is characterized by a U-shaped design, consisting of a contracting path to capture context and a symmetric expanding path for precise localization.

DeepLabV3 is an advanced model for semantic image segmentation. It probes an image with filters at multiple rates and fields-of-view, enhancing the model's ability to segment objects at various scales.

FPN (Feature Pyramid Network) is a framework primarily designed for object detection that excels in recognizing objects at multiple scale levels.

U-Net++ is an iterative improvement over the classic U-Net model. It introduces a series of nested, dense skip pathways, aiming to provide more precise segmentation maps.

B. Training results with limited data

In this subsection, the outcomes of our models' training are presented, using 80 images for training and 20 images for validation.

Table II outlines the specific parameters used during the training of our models (the learning rate is denoted as lr).

Loss	lr	Optimiser	Metric	Epochs	Encoder	Weights
Dice	10^{-4}	Adam	F1 score	40	resnet34	imagenet

TABLE II

Figure 3 visually compares the performance of the four models. It allows us to have a benchmark for the initial phase of our experimentation, setting the stage to optimise a specific model in the rest of the project.

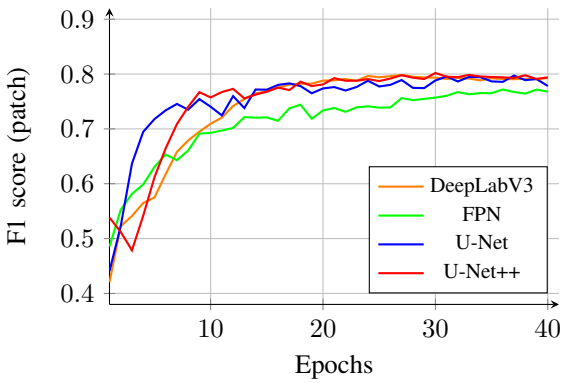

Fig. 3. Models comparison with limited data

C. Selection of U-Net architecture

Compared to the other three architectures, FPN exhibited lower performance and a slower convergence rate. Also, it was found that U-Net, DeepLabV3, and U-Net++ had similar performance levels. However, U-Net was chosen for further testing due to its ease of training and better scalability. U-Net++ faced memory issues with larger batch sizes, and both U-Net++ and DeepLabV3 training were more time-consuming.

VI. Data augmentation and learning rate

This section explores two techniques that greatly increased the model predictive capabilities. Namely, data augmentation and learning rate optimization.

A. Data augmentation techniques

To enhance the model's ability to generalize, the dataset of images was augmented using the Albumentations library. The augmentation process included the following transformations:

Flipping: Horizontal and vertical flips were applied to ensure that models could identify roads regardless of their orientation in the image.

Rotation: Images and their corresponding masks were rotated by 90°, 180°, and 270°.

Small Rotations: In addition to fixed rotations, random small rotations of up to 45° were introduced, adding more variability to the dataset and helping the models adapt to minor orientation differences.

During the experimentation, other augmentation techniques such as cropping and resizing were explored. However, it was

observed that these did not significantly enhance the model's performance. This is likely due to all aerial images provided sharing the same resolution, as previously mentioned in Section III. Additionally, rotations of underrepresented images were also explored, but no significant result improvements were observed.

B. Optimizing with Adam and Learning Rate Scheduler

The optimization strategy involved the combination of the Adam optimizer with a PyTorch learning rate scheduler. While the learning rate was adjusted by Adam, a scheduler was introduced to further halve the learning rate when the validation F1 score showed no improvement for two epochs. This approach contributed to the fine-tuning of the model and faster convergence.

C. Results

The application of a learning rate scheduler resulted in a marginal improvement of up to 1 point in performance. In contrast, the implementation of data augmentation (2'280 images) significantly boosted the models' performance, showcasing gains exceeding 10 points.

VII. Experimentation with Loss Functions

A. Description of loss functions

To enhance the model's performance, various loss functions were experimented, with particular focus on the Tversky loss and Dice loss. These losses are especially effective for addressing class imbalance.

Focal Tversky loss[4] is an advanced variation of the commonly used Focal loss, designed specifically to address class imbalance issues. By adjusting its parameters, the balance between false positives and false negatives can be controlled, making it highly adaptable to the needs of road segmentation. The Focal Tversky loss is defined as:

$$\mathcal{L}_{\text{F. Tversky}} = (1 - \text{TI})^{1/\gamma} = \left(1 - \frac{\text{TP}}{\text{TP} + \alpha\text{FN} + \beta\text{FP}}\right)^{1/\gamma} \quad (1)$$

where α and β assign weights to false positives FP and false negatives FN, respectively, TP corresponds to true positives, γ represents the focal parameter, and TI denotes the Tversky index. (Note: if $\alpha = \beta = 0.5$ & $\gamma = 1$, Focal Tversky is equivalent to the Dice Loss)

Dice loss[4] is derived from the Dice similarity coefficient (DSC). It is particularly beneficial for binary classification tasks. The Dice coefficient calculates the overlap between the predicted segmentation and the ground truth, thus directly encouraging the model to increase the prediction accuracy for the road pixels. The Dice loss is defined as:

$$\mathcal{L}_{\text{Dice}} = 1 - \text{DSC} = 1 - \frac{2\text{TP}}{2\text{TP} + \text{FN} + \text{FP}} \quad (2)$$

where TP, FN and FP corresponds to true positives, false negatives and false positives, respectively. The two losses were explored with an augmented dataset, and the parameters of the Focal Tversky loss were optimized through Bayesian optimization, as presented in Section IX.

VIII. Post-processing

Model accuracy was enhanced by averaging predictions for each test image across four rotations (0°, 90°, 180°, 270°). This method reduced orientation bias and ensured more consistent and reliable predictions, leading to an overall improvement in model performance of approximately 1 point.

IX. Hyperparameter optimization

After defining the pre-processing and training framework, the optimization of hyperparameters for the model was explored through Bayesian Optimization (BO). It was decided to run BO with Focal Tversky, as results comparable to Dice can also be explored by the BO as shown in section VII.

The following optimization problem was defined:

Maximize:

$$\max_{\alpha,\beta,\gamma,\text{lr}} \quad f1_{patch}(\alpha, \beta, \gamma, \text{lr})$$

Characteristics:

- $0.3 \leq \alpha \leq 0.7$
- $0.3 \leq \beta \leq 0.7$
- $0.5 \leq \gamma \leq 2$
- $1 \times 10^{-5} \leq \text{lr} \leq 1 \times 10^{-2}$.
- With $f1_{patch}(\alpha, \beta, \gamma, \text{lr})$ the patch based f_1 score calculated on our validation set
- α, β, γ the hyperparameters used in our Tversky focal loss
- A learning rate scheduler dividing the learning rate by 2 with patience of 2 was used
- Initialization with 2 random points, and 10 for search

This method allows us to explore a predefined hyperspace for our hyperparameters, balancing exploration and precision for the sampling process. This methodology is often used in deep learning frameworks for tuning slow training processes.

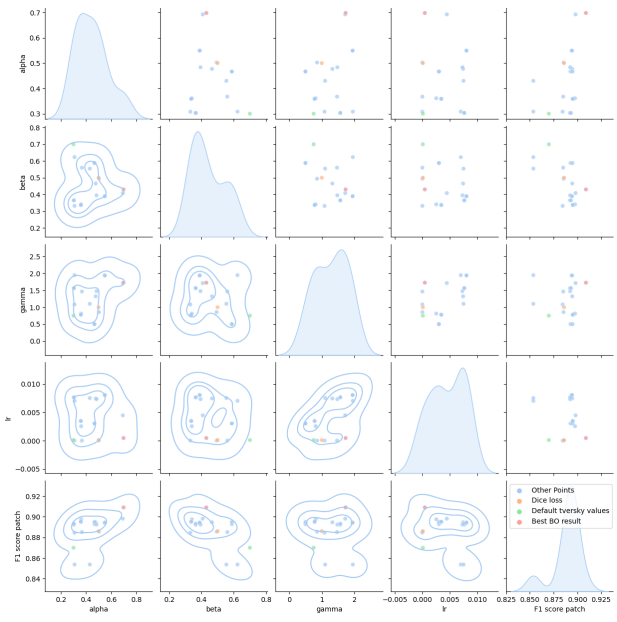

Fig. 4. Hyperspace exploration based on BO

The multidimensional parameter space explored is visualized through a pair plot, providing insights into the relationships between different parameters (Figure 4). The optimal result from Bayesian Optimization (BO), marked by the red point, is noticeably near the original Dice run when visually examined. This observation suggests that the initial choice was sound, and the default parameters were already well-optimized. The parameters used can be found in Table III.

Run	α	β	γ	lr	F1 patch	F1 AIcrowd
Dice loss	0.5	0.5	1	1e-4	0.886	0.913
Default Tversky	0.3	0.7	0.75	1e-4	0.882	0.881
Optimal BO	0.70	0.43	1.73	4.5e-4	0.906	0.910

TABLE III

As illustrated in Figure 5, the optimal parameters obtained through Bayesian optimization indeed result in a better F1 score per patch on the validation dataset but perform slightly less well on the test dataset. This is likely due to the fact that the model trained with the Dice loss generalizes better to the images in the test dataset as it does not focus on class imbalance.

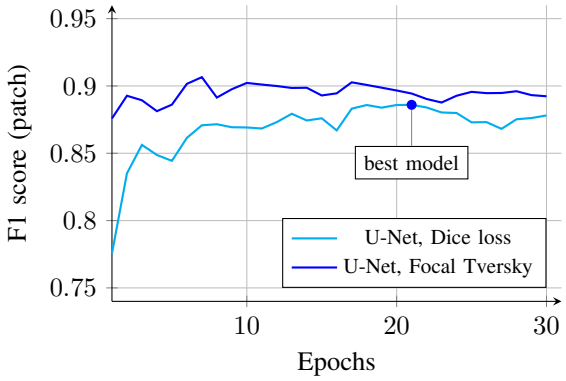

Fig. 5. Dice and focal Tversky losses comparison

X. Overall results

This section provides a summary of the overall results achieved in this project. The performance of the U-Net model with various configurations and optimizations is tabulated below, showcasing the effectiveness of different approaches in road segmentation from aerial images.

Configuration	Loss Function	Data Augmentation	F1 (AIcrowd)	Comments
U-Net Baseline	Dice	No	0.766	Baseline performance
U-Net Optimized	Dice	Yes	0.90	Improved with augmentation
U-Net with Tversky	Tversky	Yes	0.910	Tversky loss best
U-Net Final	Dice	Yes	**0.913**	With post-processing

TABLE IV
Summary of U-Net Model Performance

Table IV illustrates the incremental improvements in the model's performance through various stages of the project, culminating in the final configuration which provided the best results in terms of the F1 score.

XI. Ethical risk assessment

A. Risk context and impact

A significant ethical concern in aerial imagery-based road segmentation project is the potential bias toward urban environments. The dataset predominantly consists of urban area images, which might lead to the model underperforming in non-urban settings, such as rural areas with dirt roads.

For proof, consider the abundant use of road identification models, which have previously led to serious issues. This includes instances where secondary or decrepit roads were identified as viable, potentially endangering users' lives due to unusually increased traffic on small roads[5] or suggestion of dangerous roads. This problem is exacerbated in underdeveloped countries, from which quality satellite road data is low[6].

B. Risk evaluation and mitigation

To quantify this risk, the F1 score could be measured separately for urban and rural roads. This would enable a direct comparison of the model's performance in different environments.

For roads that were poorly classified and have a history of accidents, hand classification and human monitoring of recommended roads could still be relevant.

Should dataset limitations not be a factor, the mitigation strategy would involve diversifying the training set by primarily including a larger proportion of rural roads. Monitoring accident data to identify and address hotspots requiring special attention would also help address the issue.

XII. Discussion

The project achieved significant success in applying the U-Net model for road segmentation from aerial images, underscored by high F1 scores. Key results include the effective segmentation capability of U-Net, particularly with the optimized Dice and Tversky loss functions, and data augmentation. These results highlight the potential applications in urban planning, where accurate road mapping is critical. The model was able to capture diverse road types and conditions, which highlights the versatility of deep learning models in image analysis.

However, the study's focus on predominantly urban areas may limit the model's effectiveness in rural or varied landscapes, and its performance in challenging conditions like low-light or cloudy days remains untested. Future work should diversify the training dataset to include more non-urban environments and challenging weather conditions.

XIII. Conclusion

This project successfully developed a deep learning classifier for road segmentation of satellite images, primarily using the U-Net model with Dice and Tversky loss functions. Key improvements in the F1 score were accomplished by data preprocessing, data augmentation, and overfitting prevention strategies such as learning rate adjustment and 'early saving'.

References

[1] Pavel Iakubovskii. *Segmentation Models Pytorch*. https://github.com/qubvel/segmentation_models.pytorch. 2019.

[2] Alexander Buslaev et al. "Albumentations: Fast and Flexible Image Augmentations". In: *Information* 11.2 (2020). ISSN: 2078-2489. DOI: 10.3390/info11020125. URL: https://www.mdpi.com/2078-2489/11/2/125.

[3] Fernando Nogueira. *Bayesian Optimization: Open source constrained global optimization tool for Python*. 2014. URL: https://github.com/fmfn/BayesianOptimization.

[4] Michael Yeung et al. "Unified Focal loss: Generalising Dice and cross entropy-based losses to handle class imbalanced medical image segmentation". In: *Computerized Medical Imaging and Graphics* 95 (Jan. 1, 2022), p. 102026. ISSN: 0895-6111. DOI: 10.1016/j.compmedimag.2021.102026. URL: https://www.sciencedirect.com/science/article/pii/S0895611121001750 (visited on 12/04/2023).

[5] *Automated navigation systems are still wreaking havoc on small towns' streets*. https://algorithmwatch.org/en/navigation-systems-small-towns/.

[6] *NTSC Conducts Investigation Analysis on PT KCIC Work Train Accident*. https://en.tempo.co/read/1680327/ntsc-conducts-investigation-analysis-on-pt-kcic-work-train-accident?tracking_page_direct.