CS-433 : Road Segmentation

Amine Youssef, Aït Lalim Adrien Omar, Liess Gröli
EPFL Lausanne, Switzerland

***Abstract*—In this project, we address the task of segmenting satellite imagery in pixel order, a critical challenge for applications such as autonomous navigation and urban planning. Leveraging deep learning techniques, we implement and evaluate the U-Net and Nested U-Net architectures, known for their superior performance in semantic segmentation tasks. These models incorporate encoder-decoder structures and skip connections, enabling precise spatial localization while maintaining computational efficiency. Through extensive experimentation and optimization, including data augmentation and hyperparameter tuning, our best model achieves an F1 score of 0.908 and an accuracy of 0.949 on the test set.**

I. Introduction

The objective of this project is to develop a machine learning model capable of segmenting roads from aerial satellite images. This task falls under semantic segmentation, where each pixel in an image is assigned a specific label—in this case, "road" or "background." Semantic segmentation has rapidly advanced thanks to fully convolutional neural networks (FCNs) such as U-Net, a popular architecture initially designed for biomedical image segmentation. U-Net's encoder-decoder structure with skip connections enables efficient feature extraction and precise localization, while its enhanced variant, the Nested U-Net, improves performance with dense skip connections for better gradient flow and feature reuse. These features make both architectures highly suitable for road segmentation challenges, especially when dealing with occlusions caused by trees, shadows, or other obstacles. The training dataset for this project consists of 100 aerial satellite images, each in RGB format with a resolution of 400×400 pixels. Each image is paired with a grayscale mask where white pixels represent roads, and black pixels indicate the background. The testing dataset includes 50 aerial images with a higher resolution of 608×608 pixels. Both datasets are sourced from Google Maps. By experimenting with U-Net and Nested U-Net architectures, optimizing hyperparameters, and employing advanced data augmentation techniques, this project aims to achieve accurate and robust road segmentation.

II. Data Augmentation

Before committing to a specific model architecture, we first analyze the dataset to determine effective augmentation methods that can promote broader generalization and avoid overfitting. Because the labeled set has only 100 samples, using them as-is may fall short for training a deep neural network to achieve robust performance. To address this challenge, we design and implement the following data augmentation pipeline, by applying:

- **Resize**: All input samples are resized to a fixed dimension of 384×384 pixels where 384 is the largest number smaller than 400 and divisible by 32. This preprocessing step ensures uniformity across the dataset, which is crucial for compatibility with downstream model architectures and facilitates batch processing.
- **Shifts**: To simulate positional variations in the dataset, each sample undergoes a random shift in both the horizontal and vertical directions. The translation offsets (Δx and Δy) are sampled uniformly from a range of $[-30, 30]$ pixels [4].
- **Rotations**: We rotate each image by all the selected angles from the set $\{45°, 90°, 135°, 180°\}$. This operation enhances the model's rotational invariance, allowing it to recognize patterns regardless of their orientation [4].
- **Flips**: To further increase diversity, each sample undergoes:
  - **Vertical Flip**: A flip along the horizontal axis is applied.
  - **Horizontal Flip**: A flip along the vertical axis is also applied. These flips help account for mirrored patterns or symmetries that may appear in the data.
- **Brightness**: To simulate real-world lighting variability, the brightness of each sample is scaled by a random factor drawn from a uniform distribution in the range $[0.6, 1.4]$. Factors below 1 darken the image, while factors above 1 brighten it. This augmentation improves robustness to variations in illumination.

Fig. 1. A sample batch of augmented data

III. Models

Now that we have taken care of augmenting our data, we can start designing the model we will be using. Like said in the introduction, we decided to use both U-net and U-net++(also

called nested U-net) as our models. Now let's delve on how each of them is designed.

A. U-Net:

UNet is highly effective for road segmentation due to its encoder-decoder architecture, which efficiently captures both high-level contextual information and fine-grained spatial details essential for accurately delineating roads. The encoder path progressively reduces the spatial dimensions while extracting robust features, enabling the model to understand complex road patterns and varying environmental conditions. Crucial to UNet's success are its skip connections, which directly transfer feature maps from the encoder to the decoder, preserving spatial information lost during downsampling. This ensures precise localization of road boundaries and helps the network maintain high-resolution details in the segmentation output. Additionally, UNet's ability to handle class imbalances through tailored loss functions makes it well-suited for distinguishing roads from diverse backgrounds, resulting in reliable and accurate road segmentation in various scenarios.

We propose a configurable U-Net architecture that allows you to choose three key parameters: how many downsampling/upsampling layers D the network has (but you have to make sure that the dimensions of the picture are dividable by 2^D), the number of sequential convolutional blocks L at each scale, and the base number of channels C after the initial input layer. These three parameters can be tuned to balance computational constraints with the complexity of your segmentation task [3].

In the classic U-Net, you typically have a fixed number of layers, a fixed number of convolution blocks per scale, and a set channel progression. Our version removes these restrictions. First, you decide how many downsampling steps D you want. Each downsampling step halves the spatial resolution and doubles the channels, extracting deeper features at each step. Next, at each scale of the network, instead of just two convolutional blocks, you can specify L the number of consecutive convolutional operations. This repetition can lead to richer feature representations, though it also increases computation. Finally, the base channel count C sets how many feature maps the very first convolution layer will produce; this value then grows (in the encoder) or shrinks (in the decoder) as you move through the network.

As in a standard U-Net, each downsampling layer in the encoder has a corresponding upsampling layer in the decoder. Skip connections transfer high-resolution spatial detail from the encoder directly into the decoder, refining the upsampled representations. The final output layer returns to the original image resolution.

B. U-Net ++:

UNet++ enhances the standard UNet architecture, making it even more adept for road segmentation by introducing nested and dense skip connections that facilitate more refined feature fusion across multiple scales. This improved connectivity allows for better gradient flow and deeper supervision, enabling the network to capture intricate road structures and subtle variations in road appearance more effectively. The dense skip pathways in UNet++ help integrate multi-scale contextual information, which is crucial for accurately segmenting roads in complex urban environments with varying textures, shadows, and occlusions. Additionally, the architectural refinements of UNet++ lead to smoother and more precise road boundary predictions, reducing segmentation errors and enhancing overall accuracy. These advancements make UNet++ particularly suitable for demanding road segmentation tasks where high precision and robustness are required. [5] [1]

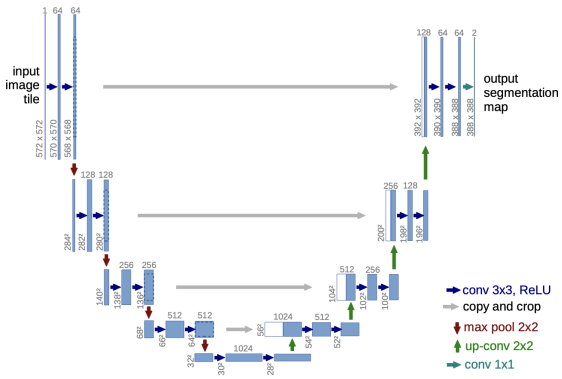


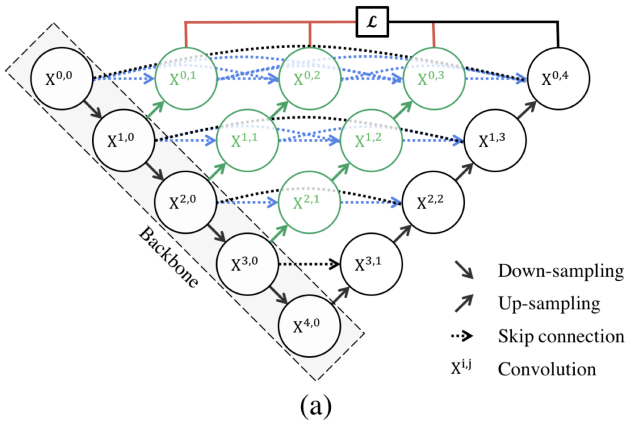
Fig. 2. An illustration that explains how U-Net is designed with L=2, C=64 and D=4. Each convolutional block in our proposed architecture is designed to extract and refine feature representations at a given scale. The block begins by receiving a feature map $X \in \mathbb{R}^{C_{in} \times H \times W}$ where C_{in} and H, W represent the spatial dimensions. We first apply a 2D convolution using a 3×3 kernel with stride 1 and padding 1 to preserve the spatial resolution. This would transform X in $Z \in \mathbb{R}^{C_{out} \times H \times W}$ s the chosen number of output channels determined by the current scale and the base channel count C. To promote stable and efficient training, we optionally include a normalization step, such as Batch Normalization. This normalization mitigates internal covariate shift and improves gradient flow. Finally, we apply a nonlinear activation function, typically a Rectified Linear Unit (ReLU). This introduces nonlinearity, enabling the network to learn complex mapping functions. By repeating this sequence of operations L times at each scale, the network incrementally refines and enriches its internal feature representations, ultimately aiding in more accurate segmentation results. [3]

IV. Training & Validation

A. Training

After creating our models and applying data augmentation, the next step is to choose the optimizer, scheduler, and loss function. We chose the Adam optimizer with a learning rate of 10^{-4} and a weight decay of 10^{-5} for regularization. For the loss function, we selected Dice Loss [2] since we noticed that road classes only represent 18.7% of the total classes.

With the optimizer, scheduler, and loss function in place, we can proceed to train our model. In training, we have two choices: the first one in which each epoch consists of randomly sampling 150 batches (each of size 8) from our augmented dataset, which serve as the training data for that epoch. This process is repeated for a total of 50 epochs. This option consumes a lot of memory since we need to augment all our initial data at first. Our second training method is where we would extract 20 images that we are gonna then augment; we



(a)

Fig. 3. An illustration that explains how U-Net is designed with D=4 [5]. Each convolution blocks is the designed the same way as described in Figure 2 for U-Net model. For each $x^{i,j}$ we would be concatenating all their connected layers then applying convolution on the concatenated layers so that we get the number ofc channels of the level we are on.

would train with them, but this method has also a drawback since we are feeding our model the same images, it may overfit because of the little number of images we are using.

After training our model, we can proceed to predict on the test set and submit our results to AIcrowd. First, we need to resize the test images which are originally 608×608 using the same scaling factor applied to the training images, which would make the new size $\frac{608384}{400} = 584$. Next, we will divide each resized image into four smaller images of size 384×384, since our model was trained on images of this size. Once predictions are made on these smaller images, we will merge them back together and resize the resulting image to match the original test image size of 608×608.

B. Validation

For validation, we handpicked four different images, chosen specifically because they feature varied road types, including large roads, diagonal roads, and complex road structures. These images were used to test our model at the end of every epoch. We used the F1 score, a commonly used metric, as our primary performance measure, applied to both the predicted images and their ground truth.

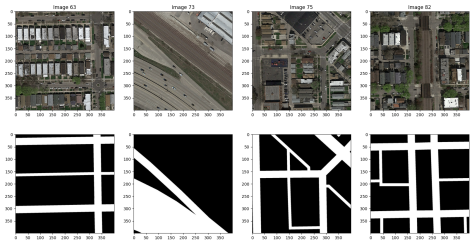


Fig. 4. Validation set that was used to validate our results

V. Results

We tried different parameters with the two models, implemented different parameters, and then recorded the validation f1 score (we should note that in all these we are using 4 layers).

TABLE I
THIS TABLE RECORDS ALL THE RESULTS BY THE DIFFERENT MODELS THAT WE TESTED

Model name	C	L	F1-score	AI crowd F1	submission ID
U-Net	64	1	0.811	-	-
U-Net	64	2	0.896	0.897	#278026
U-Net	64	4	0.911	0.908	#278023
U-Net	128	2	0.909	0.876	#278025
U-Net++	64	1	0.763	-	-
U-Net++	64	2	0.927	0.902	#277592
U-Net++	64	4	0.866	-	-
U-Net++	128	1	0.670	-	-

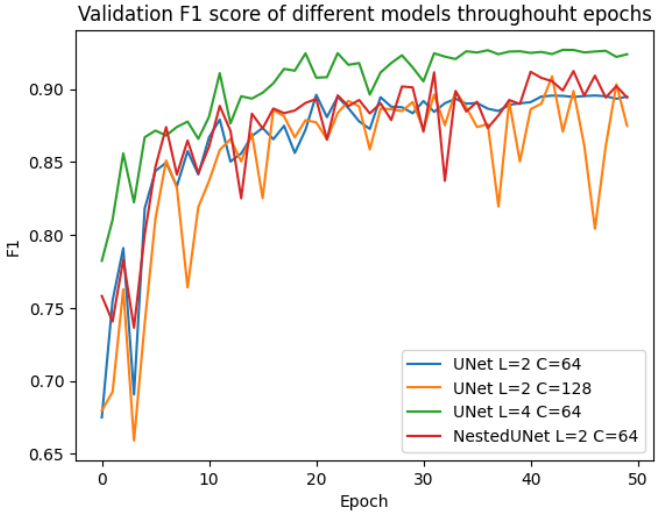


Fig. 5. Validation set f1 score evolution through the epochs for the models that gave us the best results

The U-Net with four levels ($L = 4$) and 64 base channels ($C = 64$) achieves the highest overall F1 scores on AIcrowd (0.908 Submission ID: #278023), surpassing all other configurations. Among the remaining models, the Nested U-Net (often referred to as U-Net++) also performs strongly (F1 score on AIcrowd SubmissionID : #277592), though it does not quite match the peak performance of the deeper U-Net. Which is consistent with the results we got in Figure 5, as we can see the F1 score of the U-Net with $L = 4$ and $C = 64$ has the best results and doesn't fluctuate as much as the other models. Following this model, the Nested U-Net validation seems to be better than the other models.

VI. Conclusion

In this project, we explored and compared the performance of U-Net and Nested U-Net architectures for the task of road segmentation from satellite images. Through optimization, including tailored data augmentation and hyperparameter

tuning, our best configuration achieved an F1 score of 0.908 and an accuracy of 0.949 on the test set. These results demonstrate the effectiveness of U-Net models for demanding semantic segmentation tasks, particularly when enhanced with dense connections and diverse augmentation pipelines. While the performance is satisfactory, there remain potential areas for improvement, such as integrating advanced preprocessing techniques or leveraging pre-trained models for similar tasks. Additionally, expanding the dataset could further enhance the robustness and generalization of the models.

References

[1] Renjie Li et al. *A Comprehensive Review on Deep Supervision: Theories and Applications*. 2022. arXiv: 2207.02376 [cs.CV]. URL: https://arxiv.org/abs/2207.02376.

[2] Vishal Rajput. *Robustness of different loss functions and their impact on networks learning capability*. 2021. arXiv: 2110.08322 [cs.LG]. URL: https://arxiv.org/abs/2110.08322.

[3] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. arXiv: 1505.04597 [cs.CV]. URL: https://arxiv.org/abs/1505.04597.

[4] Kai Sheng Tai, Peter Bailis, and Gregory Valiant. *Equivariant Transformer Networks*. 2019. arXiv: 1901.11399 [cs.CV]. URL: https://arxiv.org/abs/1901.11399.

[5] Zongwei Zhou et al. *UNet++: A Nested U-Net Architecture for Medical Image Segmentation*. 2018. arXiv: 1807.10165 [cs.CV]. URL: https://arxiv.org/abs/1807.10165.

VII. Ethical

Situation 1: Ethical Risk Recognized.

This project has squabbles where road segmentation models may be used for surveillance or breach of privacy of individuals. This risk affects the occupants of the regions that have been captured in a satellite image and governments or organizations that want to adopt the technology. The potential negative impact is that these models can serve to keep records of how individuals move, monitor the activities on a private property(interrogation on the non-consenting individual), which amounts to privacy infringement. Although this risk of occurrence relies on the policies of the organizations using the technology, the degree is great, because the consequences are even worse — large scale destruction of privacy and trust in AI systems.

In order to evaluate this risk, we examined how the other existing technologies are doing in regard to satellite images that have been collected, examined the policies provided for the privacy of the geospatial images and data and territory, and always considered the social responsibility of the AI technology in surveillance scenarios. No direct metrics were measured, but during the research, the developments showed that some abuse has occurred in the same situation in the past.

In mitigating this risk, we included some measures aimed at the ethical use of the segmentation models. For example, we made it clear in the project report that the technology should be used with respect to planning and navigation that could suggest unwanted uses that would violate privacy. However, assessing this risk fully is not immediately achievable in our project as it necessitates the enforceability of law and the compliance of end-users to privacy standards. As such, the main limitation is the lack of regulation over how the models were put to use once the models were made available to the public.