Road Segmentation Using UNet and ResNet-UNet Architectures: A Comparative Study

Quentin Chappuis, Louis Martins, Kelu Huang

ABSTRACT

In this report, we explore the implementation and evaluation of road segmentation models applied to satellite imagery. Our work focuses on two architectures: the UNet and a ResNet-UNet. These models were trained on a small dataset of annotated satellite images to classify each pixel as road or background.

I. INTRODUCTION

Semantic segmentation is a fundamental task in computer vision that involves assigning a class label to every pixel in an image. For road segmentation, this translates to distinguishing road regions from non-road areas. Applications range from autonomous driving systems to urban planning and disaster response. In this project, we aimed to design and implement a machine learning pipeline capable of accurately segmenting roads in satellite images. For this, we utilized two popular deep learning architectures: UNet and ResNet-UNet. Given the limited size of the dataset (100 annotated images), our approach incorporated several strategies to mitigate overfitting, including data augmentation and loss function optimization. This report will explain the methodology employed and review the results obtained.

II. DATASET AND DATA AUGMENTATION

A. Dataset Description

The dataset provided consisted of 100 satellite images, each accompanied by a binary ground truth mask and 50 test images. The images are of size 400x400 and 608x608 for the test images. The masks labeled each pixel as either road (1) or background (0). The 100 images were divided into training and validation to evaluate model performance effectively.

B. Data augmentation

To prepare the data for model training, we conducted data augmentation to artificially increase the size and diversity of the dataset by applying random transformations to the training data. This helps improve the robustness and generalization ability of machine learning models, especially in tasks like image segmentation.

We used the following augmentations:

1) **Horizontal flip:** Randomly flips the images and masks horizontally with a probability of 50%. This ensures that the model does not develop a bias toward specific orientations, such as preferring roads that predominantly run left to right.
2) **Rotations:** Randomly rotates the images and masks by 90 degrees. We chose a probability of 0.5.
3) **Random shadow:** We randomly add triangular zones of shadow or overlaid patterns in training data to mimic natural occlusions, such as roads covered by trees This should help the model to better predict roads that are covered by trees. We chose to set intensity to 0.6 and the probability to 0.7.
4) **Brightness and contrast:** Randomly adjusts the brightness and contrast of the image with a probability of 0.5. Satellite images are often captured under different lighting conditions. Brightness adjustments simulate variations caused by shadows, time of day, and weather, helping the model generalize to diverse illumination levels.
5) **Random resizing and cropping:** Crops a random portion of the image with probability 0.5 and resizes it to 384×384. Simulates zooming in on specific sections of the road, such as intersections or smaller features, ensuring the model learns to segment fine details and works on images of varying resolutions.

After augmentation, we had 500 images that we decided to resize to 384x384.

III. MODEL ARCHITECTURES

A. UNet

The UNet architecture, developed by Ronneberger et al. (2015), is a type of deep neural network designed in an encoder-decoder structure. It is composed entirely of convolutional layers, with the encoder using max-pooling operations to gradually downsample and condense spatial information. Conversely, the decoder employs transposed convolutional layers with strides to upsample and reconstruct the data. This design creates a bottleneck at the center of the network, forcing it to

focus on the most critical features for effective performance.

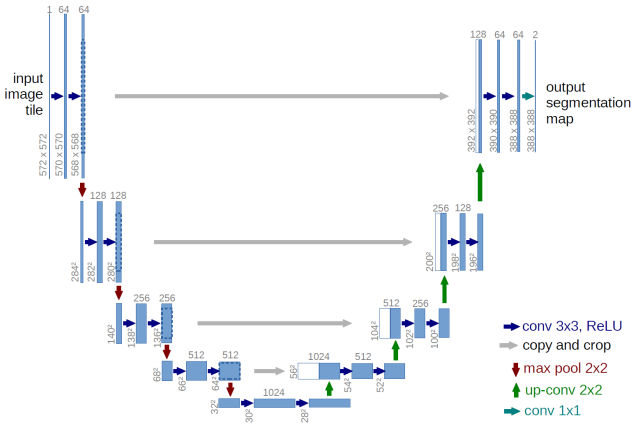

Fig. 1. UNet architecture

As illustrated in Figure 1, the UNet's architecture comprises blue boxes representing multi-channel feature maps. The number of channels is indicated at the top of each box, while the spatial dimensions are shown at the lower left. White boxes symbolize copied feature maps, and arrows indicate the different operations. The encoder, forming the left side of the "U," is responsible for feature extraction and includes four downsampling steps. At each stage, the input image—represented as a 3D matrix—passes through two 3x3 convolutional layers followed by an activation function. This is succeeded by a max-pooling layer, which reduces the image's dimensions while increasing the number of channels, allowing the network to capture low-level features of the input.

The decoder, or upsampling section, mirrors the encoder's structure with four upsampling steps. Each upsampling operation begins with a 2x2 transposed convolutional layer, doubling the spatial dimensions of the image, followed by two 3x3 convolutional layers with activation functions. Through this reverse process, the image dimensions are progressively restored while the number of channels is halved, enabling the reconstruction of the original image.

Additionally, UNet incorporates four skip connections, or "copy and crop" operations, which link corresponding downsampling and upsampling stages. These connections combine low-level features from the encoder with higher-level features from the decoder, integrating detailed and contextual information. If feature maps differ in size or channel count, the network crops them to ensure seamless concatenation. This mechanism enhances the network's ability to capture both fine and coarse details, improving overall performance.

B. ResNet-UNet

The ResNet-UNet architecture combines the strengths of ResNet, a deep residual network, with the encoder-decoder design of the UNet. The encoder uses ResNet's pretrained convolutional layers to extract hierarchical features, progressively reducing spatial resolution through downsampling. The decoder, similar to UNet, uses transposed convolutions to upsample and reconstruct spatial information, incorporating skip connections to merge encoder features for finer detail preservation. This integration of ResNet's robust feature extraction and UNet's spatial reconstruction introduces an informational bottleneck in the network, making the model focus on essential features for accurate segmentation.

ResNet introduces skip connections that allow gradients to flow directly through layers without vanishing, enabling efficient learning of deep features. U-Net, on the other hand, employs an encoder-decoder structure, where the encoder compresses spatial dimensions to extract features, and the decoder reconstructs the original dimensions while preserving critical details. The integration of these two models leverages ResNet's residual blocks in the encoder for robust feature extraction, while U-Net's decoder reconstructs the output with high accuracy.

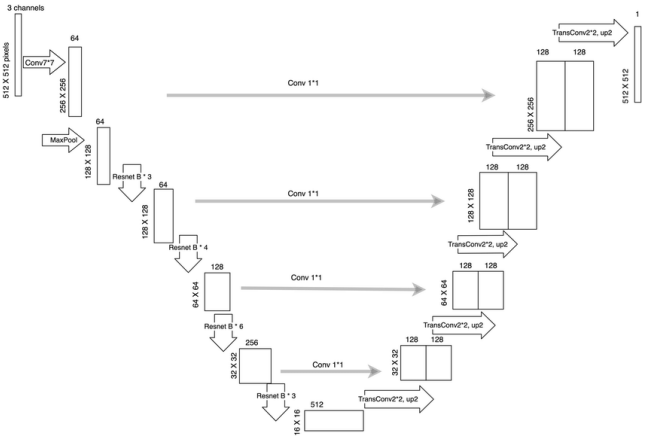

Fig. 2. ResNet-UNet architecture

Figure 2 illustrates this architecture, beginning with a 512 × 512 × 3 input image, which is processed through a 7 × 7 convolutional layer to extract 64 feature channels, reducing the spatial dimensions to 256 × 256. The encoder, depicted on the left side of the U-shaped structure, employs max-pooling operations to progressively reduce the spatial dimensions while increasing the number of feature channels, with residual blocks enhancing feature learning at each stage. At the bottleneck, the network achieves its narrowest point, with 16 × 16 spatial dimensions and

512 feature channels, focusing on extracting the most critical features.
The decoder, on the right side of the U, mirrors the encoder's structure but restores spatial dimensions through transposed convolutions, halving the number of channels at each step. Skip connections, represented by horizontal gray arrows, link corresponding layers in the encoder and decoder, allowing the model to combine low-level features from the encoder with high-level features from the decoder, enriching the reconstruction process. The final output is a single-channel 512 × 512 image, typically representing a segmentation map where each pixel corresponds to a specific class label.

IV. Training Setup

A. Hyperparameters

The models were trained using the following parameters:

- **Optimizer:** Adam optimizer, learning rate of 1×10^{-4} or AdamW with learning rate of 1×10^{-4} and weight decay of 1×10^{-5}
- **Scheduler:** Cosine annealing scheduler with $T_{max} = 10$ and $\eta_{min} = 1 \times 10^{-6}$
- **Batch Size:** 16,8,4 or 2 depending on the model.
- **Epochs:** Depends on the model and its convergence, we used 10, 50 and 100.

B. Validation

As we have 500 images after augmentation, we decided to use 85 percent i.e. 425 of them for training and thus 15 percent i.e. 75 for validation.

C. Dice loss

In road segmentation tasks, there is often a class imbalance: roads (positive pixels) are much fewer compared to the background (negative pixels). Dice Loss focuses on the overlap (intersection) between predictions and ground truth, ensuring better performance on underrepresented classes like thin roads.

D. BCE loss

BCE (Binary Cross-Entropy) Loss, is a commonly used loss function for binary classification tasks. It measures the difference between predicted probabilities (from a model) and the true binary labels (0 or 1). The goal is to minimize this loss, which encourages the model to predict probabilities close to the actual class labels.

E. Adam(W) optimizer

The Adam optimizer (Adaptive Moment Estimation) is one of the most widely used optimization algorithms in machine learning, particularly in deep learning. It is an extension of the stochastic gradient descent (SGD) algorithm that incorporates adaptive learning rates and momentum, making it well-suited for large-scale data and non-stationary problems. AdamW is a variant of the Adam optimizer that includes weight decay as part of the regularization process. Unlike Adam, where weight decay is applied to gradients, AdamW separates weight decay and gradient updates, improving optimization performance.

F. Learning rate

Learning rate may have a considerable influence on the convergence of the model. A too high learning rate may lead to unstable training. The loss may oscillate or even diverge instead of decreasing. On the other hand, too low of a learning rate may cause the training to be slow and inefficient. The model might never reach the global minimum within a reasonable time. Thus an optimal learning rate balances speed and stability. It allows the model to converge efficiently while avoiding oscillations. So we tried different learning rates, namely: $1 \times 10^{-3}, 1 \times 10^{-4}$ and 1×10^{-5}. We found that 1×10^{-4} produces the best performance of all, so that is the one we kept.

G. Learning rate scheduler

Cosine Annealing is a learning rate scheduling technique that gradually reduces the learning rate following a cosine curve. It is particularly effective for deep learning training, as it helps models converge smoothly to an optimal solution by reducing the learning rate in a cosine-shaped manner. We use it to adjust the learning rate of the AdamW optimizer. It works as follows: the learning rate starts at the initial value and gradually decreases following a cosine function. At the end of a specified time (T_{max}), the learning rate reaches the minimum value (eta_{min}).

H. Dropout

Dropout in UNet is a regularization technique used to prevent overfitting during training. It works by randomly "dropping out" (i.e., setting to zero) a subset of the neurons or activations in a layer during forward propagation. This forces the model to learn more robust and generalized features, as it cannot rely on specific neurons for predictions. We chose a dropout probability of 0.4 as our dataset is small, so there's a higher risk of overfitting. Therefore a dropout with $p = 0.4$ allows us to combat overfitting while not losing too much information.

V. Results

We uploaded our results to AIcrowd from which we obtained the following scores:

Model	Epochs	Batch	Dropout	Optimizer	Loss	F1 score	Accuracy
Baseline (CNN)	100	16	No	Momentum	Soft-Max	0.471	0.535
UNet	10	8	No	Adam	Dice	0.764	0.866
UNet	10	8	No	AdamW	Dice	0.785	0.889
ResNet-UNet	10	8	No	Adam	Dice	0.786	0.875
ResNet-UNet	10	8	No	AdamW	Dice	0.793	0.894
ResNet-UNet	**50**	**2**	**0.4**	**AdamW**	**BCE**	**0.829**	**0.908**

TABLE I
MODEL COMPARISON

We can clearly see that our UNet and ResNet-UNet far outperform the baseline model. We varied our parameters for UNet and ResNet-UNet to try to optimize the model. In the end, we found that ResNet-UNet with batch size=2, dropout probability of $p = 0.4$, AdamW optimizer with cosine annealing scheduler and BCE loss gave us the best performance of F1=0.829 and Accuracy=0.908. We decided to run it over 50 epochs as already after 10 epochs it looked more promising than the other models.

VI. Discussion

A. Challenges

The limited dataset size restricted the models' ability to generalize to diverse road environments. High variability in lighting and occlusions introduced noise in predictions.

B. Improvements

We could improve our results by:

- Expanding the dataset with more diverse road types.
- Implementing more advanced augmentation techniques, such as elastic deformations or synthetic data generation.
- Experimenting with hybrid loss functions to balance overall accuracy.

VII. Ethical Risks

A. Bias

The dataset used in this project primarily represented urban road environments, resulting in a model that performs well on city-like infrastructure but struggles with rural or less developed regions. This bias could lead to inaccurate predictions when the model is deployed in areas with limited road infrastructure, such as rural villages or forest paths. To mitigate this issue, future datasets should incorporate balanced examples of roads from diverse geographic regions, including rural, mountainous, and desert environments. Addressing such biases will ensure that the model performs equitably across all environments.

B. Privacy concerns

The use of high-resolution satellite imagery for road segmentation raises potential privacy risks, particularly if images capture sensitive or private areas. Unauthorized use of such imagery could infringe on individual privacy, especially in residential or restricted areas. To address this, strict ethical guidelines and oversight are essential. Future projects should ensure that satellite images are anonymized and appropriately filtered to exclude sensitive data. Moreover, the use of segmentation models should be restricted to ethical and legal applications, such as urban planning or disaster management.

VIII. Conclusion

This project successfully implemented and evaluated UNet and ResNet-UNet architectures for road segmentation. The ResNet-UNet demonstrated superior performance, underscoring the value of transfer learning. We achieved F1=0.829 and Accuracy=0.908 on the AIcrowd platform. Future work will focus on addressing dataset limitations and exploring advanced model architectures for even greater accuracy.

References

[1] U-Net: Convolutional Networks for Biomedical Image Segmentation, Ronneberger, Olaf and Fischer, Philipp and Brox, Thomas, Medical Image Computing and Computer-Assisted Intervention (MICCAI), pages 234-241,2015, Springer

[2] Recurrent Residual Convolutional Neural Network based on U-Net (R2U-Net) for Medical Image Segmentation by Alom, Md Zahangir and Hasan, M M and Yakopcic, Chris and Taha, Tarek M and Asari, Vijayan K ,arXiv preprint arXiv:1802.06955,2018

[3] Road Extraction by Deep Residual U-Net by Zhang, Zhixiang and Liu, Qingjie and Wang, Yunhong, IEEE Geoscience and Remote Sensing Letters, page 749-753,2018