Deep Learning for Road Segmentation

Ahmad Al-hmouz, Fares Hatahet, Basil Alajlouni, Adnan Sawalha
Deep Learning, Department of Artificial Intelligence, University of Jordan
GitHub Repository: https://github.com/KshKsh0/project-with-gitba

***Abstract*—The accurate detection of roads from satellite and aerial images is a critical task for urban planning, map updating, and transportation management. Recent advancements in deep learning, particularly with convolutional neural networks (CNNs), have demonstrated promising results in road segmentation. In this paper, we present a comparative study of several deep learning models for road segmentation, including U-Net, Residual U-Net, Attention U-Net, PSPNet and D-LinkNet. We explore the impact of data augmentation techniques and post-processing methods, specifically focusing on the application of image patching to improve segmentation accuracy. Among the models tested, D-LinkNet showed the best performance with an F1-score of 0.918, achieving 5th place in the EPFL ML Road Segmentation Challenge. Our results indicate that the integration of post-processing methods like patching significantly enhances model convergence and segmentation precision. This study provides valuable insights into selecting and refining deep learning models for road extraction tasks, emphasizing the importance of appropriate data handling and model configuration.**

Keywords—Road Segmentation, Deep Learning, Satellite Images, U-Net, D-LinkNet, Attention U-Net, Image Augmentation, Semantic Segmentation, Data Augmentation, Patch Processing

I. Introduction

Satellite images play a crucial role in understanding and analyzing the world, with key applications in urban planning, navigation, and infrastructure development [1].

One significant challenge in this area is detecting roads from these images. This task is vital for updating maps, enhancing navigation systems, and managing transportation networks. However, accurate road detection is complicated by issues such as image noise, varying road shapes, obstructions (e.g., trees, buildings), and the similarity of road textures to other materials [2].

Traditional methods often fall short of achieving the precision required for reliable detection. This project aims to address these challenges by developing a classifier for pixel-level road segmentation using advanced deep learning models. By leveraging cutting-edge architectures like U-Net [3], we seek to improve the accuracy of road detection.

This paper is organized as follows: In Section 2, we review related work and highlight prior approaches to road segmentation using deep learning models such as U-Net and D-LinkNet. Section 3 defines the problem and outlines the objectives of our study. Section 4 details the methodology, including the architectures explored, dataset preparation, and training setup. Section 5 presents the results, including model performance and key observations. Section 6 concludes the paper, discusses the challenges faced and their impact on our approach, and finally proposes directions for future research.

II. Liturature Review

Recent research on road segmentation from aerial and satellite imagery has prominently featured convolutional neural networks (CNNs). Notable approaches include the use of U-Net and D-LinkNet, which offer different strengths depending on the dataset and task complexity.

U-Net, known for its encoder-decoder architecture, has shown great success in medical image segmentation [3]. Jiang et al. (2024) applied a U-Net architecture for road segmentation in aerial images. The authors addressed challenges such as background clutter, tree and car interference, and the imbalanced nature of road vs. background data. By incorporating Jaccard loss, data augmentation, and post-processing morphological operations, they achieved an F1-score of 0.892 on the AICrowd test set. Their work highlights the importance of augmenting small datasets and refining post-processing steps to improve segmentation results [4], which inspired us in our project.

Zhou et al. (2018) introduced D-LinkNet, an architecture based on LinkNet with dilated convolutions and a pretrained ResNet encoder. Applied to the DeepGlobe Road Extraction Challenge, this method achieved an IoU score of 0.6466 on the validation set and 0.6342 on the test set. D-LinkNet's use of dilated convolutions helps capture a broader context, making it effective for complex satellite imagery with varying road sizes and urban features [5].

III. Problem Definition and Objectives

The problem we are tackling is road segmentation from satellite or aerial images. The goal is to develop a model that can automatically identify and separate roads from other elements in the image, pixel by pixel. To achieve this, we will use labeled satellite images (ground truth data) to train a classifier that recognizes roads.

The main objectives are:

- Data Preparation: Organize and format the training data (images and labels).
- Model Development: Build a model to segment roads from the images.
- Model Evaluation: Assess the model's performance using the F1 score.
- Improvement: Refine the model by experimenting with different techniques and data augmentation.

The goal is to create an accurate and reliable road segmentation model for satellite imagery.

IV. Methodology

A. Model Architectures

To address our research problem, we experimented with various kinds of models in different approaches:

1) U-net: We first expermineted with U-net, a convolutional neural network architecture designed for image segmentation tasks. It follows a symmetric encoder-decoder structure: the encoder extracts hierarchical features through

downsampling, while the decoder upsamples and reconstructs the segmentation map. Skip connections link corresponding encoder and decoder layers, allowing the network to preserve fine details and spatial information *[3]*, figure 1 shows the architecture of U-net.

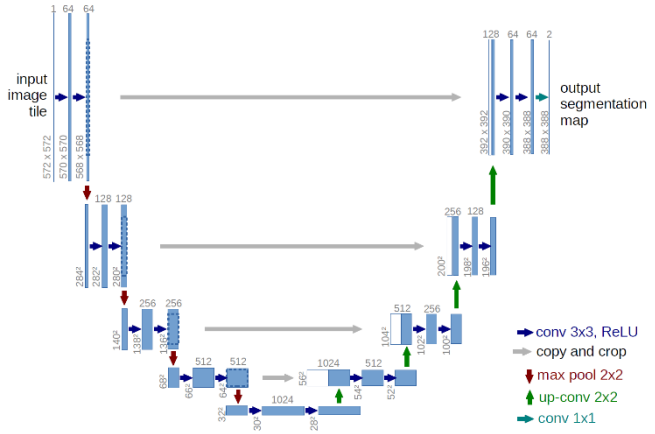

Figure 1: U-net Architecture

Then we tried different variations of U-net, including Residual U-net which is an enhanced version of the U-Net architecture that incorporates residual connections within its encoder and decoder blocks. These connections help the model learn more effectively by bypassing unnecessary layers and mitigating issues like vanishing gradients, particularly in deep networks. By combining the strengths of U-Net's symmetric structure and residual learning, the Residual U-Net achieves better performance in segmentation tasks, even when faced with noisy or cluttered images [6].

After that we tried Attention U-net which builds on the U-Net architecture by introducing attention gates (AGs) to enhance focus on relevant regions in the image. These gates learn to suppress irrelevant background information and highlight important features, such as roads in segmentation tasks, based on contextual cues. By selectively filtering features passed through skip connections, Attention U-Net improves the model's ability to capture fine details and distinguish complex structures, making it particularly suitable for tasks requiring precise segmentation in cluttered or noisy environments [7].

Then we tried Residual Attention U-Net, which combines the strengths of residual learning and attention mechanisms within the U-Net architecture to further enhance segmentation performance. Residual connections in the encoder and decoder blocks help stabilize training and capture deeper features by bypassing unnecessary layers. Meanwhile, attention gates selectively focus on relevant regions, suppressing irrelevant details [8].

In our project, we experimented with different input sizes for U-Net. For the first approach, we divided each input image into smaller patches of size 96 x 96, generating 16 patches from each original image. The second approach used the original image size of 400 x 400 without any patching. By using the smaller patches, we were able to process more detailed segments of each image during training, which significantly improved the F1-score. This patching approach allowed us to retain the original image dimensions while effectively increasing the dataset size, leading to better model performance.

2) DeepLabv3+: a state-of-the-art architecture for semantic segmentation that enhances segmentation accuracy through a combination of advanced techniques. It utilizes atrous (dilated) convolutions to capture multi-scale context effectively without losing spatial resolution. The model features an Atrous Spatial Pyramid Pooling (ASPP) module, which integrates features at different receptive fields to handle objects of varying sizes, see figure 2 for the architecture *[9]*.

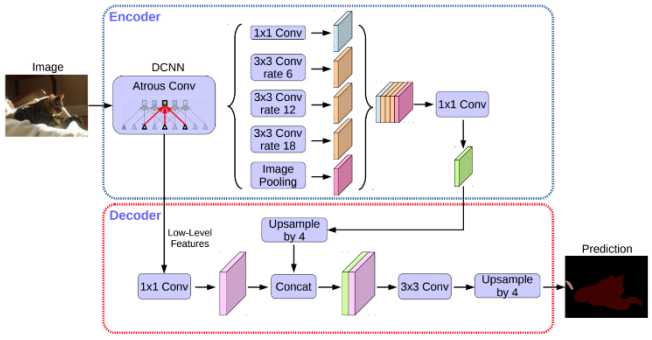

Figure 2: DeepLabv3+ Architecture

In DeepLabv3+, we applied the patching process similar to the one applied to U-net, but we unfortunately didn't get the results we hoped for, assuming it might be because of the nature of the Pyramid Pooling Module, we decided to stop using the patching process in the next models.

3) PSPNet (Pyramid Scene Parsing Network): a segmentation architecture designed to capture both local and global context by incorporating a pyramid pooling module. This module processes the feature maps at multiple scales, aggregating global information and combining it with fine-grained local details to improve segmentation accuracy, see figure 3 for the architecture *[10]*.

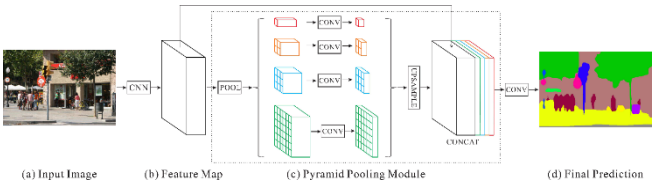

Figure 3: PSPNet Architecture

4) D-LinkNet: a segmentation architecture built on the ResNet encoder and LinkNet framework, optimized for tasks requiring accurate edge and boundary detection, such as road segmentation. The model incorporates dilated convolutions in the encoder, allowing it to capture broader contextual information without increasing the computational cost. By utilizing skip connections and an efficient decoder, D-LinkNet balances feature extraction and detail preservation, making it highly suitable for extracting long, continuous structures like roads while retaining fine details, see figure 4 for the architecture *[5]*.

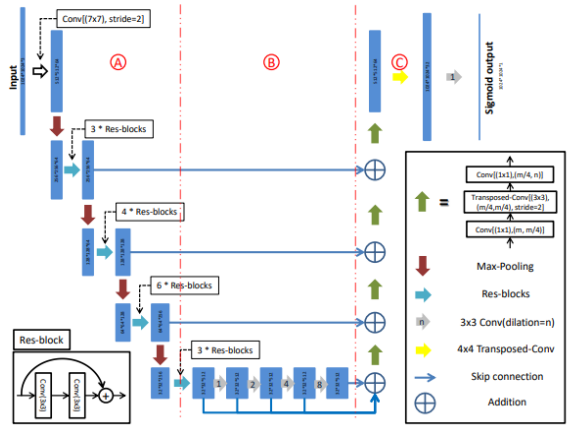

Figure 4: D-LinkNet Architecture

From the figure you can see that each blue rectangular block represents a multi-channel feature map. Part A is the encoder of D-LinkNet. We used ResNet152 as encoder. Part C is the decoder of D-LinkNet, it is set the same as LinkNet decoder. Original LinkNet only has Part A and Part C. D-LinkNet has an additional Part B which can enlarge the receptive field and as well as preserve the detailed spatial information. Each convolution layer is followed by a ReLU activation except the last convolution layer which uses sigmoid activation. Note that we also changed the input shape from 400x400 to 384x384 since the model we used does not accept inputs unless they are in a shape that is a multiple of 32.

B. *Dataset and data augmentation*

1) Dataset: The dataset in hand consists of 100 satellite images with corresponding masks, along with 50 test images. The shape of the training images is 400 x 400, while the shape of the test set is 608 x 608, and the masks are labeled as 0 for background and 1 for the road. Instead of passing the data directly to the model, we used a data generator that produces one batch per step, with a specified number of images in each batch. Additionally, data augmentation techniques were applied during batch generation, if needed.

2) Data Augmentation: To train a deep learning model, it is important to provide it with sufficient amount of inputs to obtain relevant results, and because the amount of images in the dataset we had, which was 100 images, was not enough, we decided to apply different transformations to our input images, including: Rotations by 45 °, 60 °, 90 ° and 180 °, horizontal and vertical flipping, adding random noise such as salt and pepper and Gaussian noise, and random cropping. We applied these transformations in two different approaches, Online and Offline. The offline approach was applied in order to guarantee that every single image from the dataset was transformed with every single transformation applied. The end result was a dataset of 1400 images.

C. *Training Setup*

1) Hyperparameters:

a) Learning Rate: We experimented with the following values: 10^{-3}, 3×10^{-3} and 10^{-4}.

b) Batch size: Values of 24, 16, 8, and 4 were used, depending on the model.

c) Epochs: 10, 50, 100, or 300, depending on the model.

d) Activation Function: Parametric ReLU (PReLU). The activation function was tested within the interval $[0.01, 0.3]$, and the best results were obtained with $\alpha = \{0.3, 0.15\}$.

e) Dropout: 0.5.

f) Patch Size: 96 x 96, we didn't tune it thought, we belive that tuining it might have yielded better results.

2) Validation: Without patching technique: 10%; With patching technique: 20%.

3) Loss Functions: We used the following loss functions:

a) binary crossentropy:

$$L = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)]$$

b) BCE dice loss:

$$L_{BCE-Dice} = \alpha.(L_{BCE}) + \beta.(1 - \frac{2|P \cap G|}{|P \cup G| + |P \cap G|})$$

c) BCE Jaccard loss:

$$L_{BCE-Jaccard} = \alpha.L_{BCE} + \beta.(\frac{|P \cap G|}{|P \cup G|})$$

After experimenting with these losses in several models, we found that BCE Dice loss converges the models faster.

4) Optimizers: We used the Adam optimizer.

5) Callbacks: We employed Learning Rate Scheduling with Plateau-Based Reduction, and early stopping.

V. Discussion

We noticed that the F1-score was consistently low at the beginning of our experiments, and we initially assumed that resizing the test images from 608x608 to 400x400 may have caused the poor results. Resizing could potentially result in a loss of detail, an alteration in the aspect ratio, and possible blurring or distortion of the image, all of which could affect the model's ability to identify key features. As a result, we decided to experiment with post-processing techniques to improve performance. Specifically, we applied a patching technique that involved splitting each test image into 400x400 patches, resulting in 4 smaller images from each original test image. This approach had a significant impact on the F1-score, as evidenced by the results in Table 1.

Table 1: F1-scores of models under different conditions.

Model	Post-Processing	Size	F1-score
U-net	False	400x400	0.780
U-net	True	400x400	0.852
U-net	True	96x96	0.861
ResUNet	False	400x400	0.765
Attention U-Ne	True	400x400	0.856

Attention U-Ne	False	400x400	0.794
Attention U-Ne	True	96x96	0.857
Attention Res-UNet	True	400x400	0.725
DeepLabv3+	False	400x400	0.724
DeepLabv3+	True	400x400	0.840
DeepLabv3+	True	96x96	0.774
D-LinkNet	True	384x384	0.918

As we can see from the table, post-processing significantly improved the F1-score across various models. When patching was applied, convergence during training became much faster, which also increased the training image pool to approximately 35k images. Although we applied other post-processing techniques, such as Gaussian noise reduction and morphological transformations, these did not seem to yield significant improvements in the F1-score compared to the patching technique. Nonetheless, the table highlights that post-processing, particularly patching, provided the most prominent boost to model performance.

In our experimentation with models that did not use post-processing, we tried resizing the input images. However, we found this approach led to poor results, which confirmed our initial assumption that resizing may not be a suitable alternative to post-processing in this case.

The table also shows interesting comparisons between different models. For example, we found that Attention U-net without patching outperformed U-net without patching, suggesting that the attention mechanism improved performance even in the absence of additional data augmentation. In addition, we experimented with tuning the hyperparameters of U-net, specifically adjusting the parametric ReLU and the dropout rate. We found that (α=0.3) and dropout rate of (0.5) resulted in the best performance for U-net, so we applied them across the other models, except for D-LinkNet, as we did not have enough time to test it with parametric ReLU, we instead chose ReLU for D-LinkNet due to time constraints but recognize that more extensive testing could have potentially yielded better results for this model.

Some models did not perform as well as expected. For instance, DeepLabv3+, one of the most widely used models for semantic segmentation, showed disappointing results in our experiments. Even more concerning, applying the 96x96 patching to DeepLabv3+ led to a decline in performance, suggesting that patching may not be a suitable approach for this particular model. We also briefly trained the PSPNet and ResAttention U-net models, but due to time limitations, we could not explore their full potential.

VI. Results

The best-performing model in our experiments was D-LinkNet, which achieved an F1-score of 0.918, securing us 5th place in the EPFL ML Road Segmentation Challenge [11]. We discovered this model through a GitHub repository [12], and were inspired to use it after learning that its original creator won first place in the DeepGlobe Road Extraction Challenge. During training, the final model reached a validation loss of 0.262. Figure 5 shows a visual comparison between a satellite image and the corresponding road predictions generated by our model:

Figure 5: Satellite image and corresponding road extraction prediction by the D-LinkNet model.

You can view Figure 7 that illustrates the validation loss across epochs, providing a clear view of the model's performance during training.

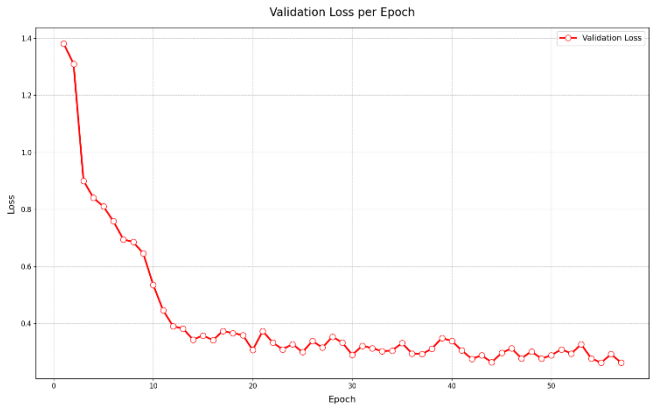

Figure 6: Validation Loss per Epoch

VII. Conclusion

In this project, we experimented with several deep learning models for road segmentation, ultimately selecting D-LinkNet as the best-performing model. With an F1-score of 0.918, D-LinkNet led us to achieve 5th place in the EPFL ML Road Segmentation Challenge. The model's architecture, inspired by the original DeepGlobe Road Extraction Challenge winner, proved effective in segmenting road features. Our experiments highlighted the importance of post-processing, particularly patching, which significantly improved F1-scores and accelerated model convergence. Despite some challenges with models like DeepLabv3+, the results demonstrated that careful experimentation with post-processing techniques and model selection is crucial for optimizing segmentation performance.

A. *Challenges*

One of the primary challenges in this project was the lack of time, which limited our ability to conduct a broader range of experiments. This constraint led us to focus on models that yielded the best results. Additionally, a lack of experience in this domain contributed to inefficiencies, as time was spent experimenting with approaches that we would have avoided—such as resizing images instead of exploring post-processing techniques.

Moreover, the small size of the training dataset presented a significant challenge, so we had to try various data augmentation techniques. Simply applying random augmentations without careful consideration would not have been ideal for model performance, as not all augmentation methods are beneficial for every model. We learned that selecting and fine-tuning the right augmentation strategies is crucial for enhancing model accuracy and ensuring better generalization, especially with limited data.

B. Future Work

- We observed that the model sometimes struggles to detect roads covered by shadows, as seen in figure 6. To address this, we believe adding a shadow effect augmentation to the training images could help the model better predict these shadowed areas.

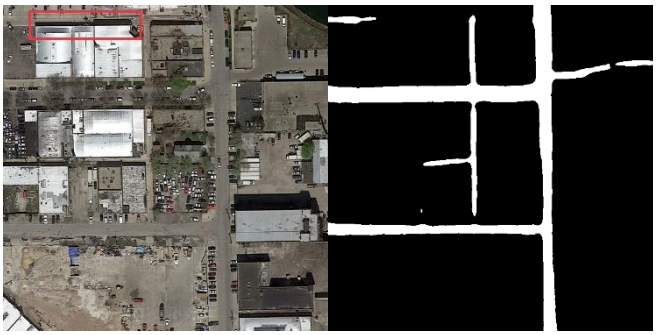

Figure 7: Model missing roads due to shadow coverage.

- Further tuning of the D-LinkNet hyperparameters might lead to improved results.

- Some roads were not detected due to obstruction from trees, as shown in figure 7. We believe acquiring more diverse training data could help tackle this issue. Additionally, experimenting with different models, such as the Dual Attention Dilated-LinkNet [13], may allow the model to focus more on road features and less on irrelevant details.

Figure 8: Roads not detected due to tree obstruction.

VIII. REFERENCES

[1] admin, "Applications of Satellite Imagery," 9 February 2024. [Online]. Available: https://satpalda.com/blogs/applications-of-satellite-imagery/.

[2] Q. Zhu, Y. Zhang, L. Wang, Y. Zhong, Q. Guan, X. Lu, L. Zhang and D. Li, "A Global Context-aware and Batch-independent Network for road extraction from VHR satellite imagery," *ISPRS Journal of Photogrammetry and Remote Sensing*, pp. 353-365, 2021.

[3] O. Ronneberger, P. Fischer and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *Medical Image Analysis*, vol. 34, p. 1–12, 2016.

[4] L. Jiang, R. Luo and D. Wang, "CS-433 Project2: Road Segmentation for Aerial Images," 2024.

[5] L. Zhou, C. Zhang and M. Wu, "D-LinkNet: LinkNet with Pretrained Encoder and Dilated Convolution for High Resolution Satellite Imagery Road Extraction," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Salt Lake City, 2018.

[6] Z. Zhang, Q. Liu and Y. Wang, "Road Extraction by Deep Residual U-Net," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 5, pp. 749 - 753, 2018.

[7] O. Okta, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz, B. Glocker and D. Rueckert, "Attention U-Net: Learning Where to Look for the Pancreas," *arXiv preprint arXiv:1804.03999*, 2018.

[8] Z.-L. Ni, G.-B. Bian, X.-H. Zhou, Z.-G. Hou, X.-L. Xie, C. Wang, Y.-J. Zhou, R.-Q. Li and Z. Li, "Raunet: Residual attention u-net for semantic segmentation of cataract surgical instruments," in *International Conference on Neural Information Processing*, 2019.

[9] L.-C. Chen, Y. Zhu, G. Papandreo, F. Schroff and H. Adam, "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018.

[10] H. Zhao, J. Shi, X. Qi, X. Wang and J. Jia, "Pyramid Scene Parsing Network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[11] E. ML, "Class Project 2 Road Segmentation," AICrowd, 2024. [Online]. Available: https://www.aicrowd.com/challenges/epfl-ml-road-segmentation.

[12] Big-BangBang, "DeepGlobe-Road-Extraction-Challenge," Github, 18 January 2024. [Online]. Available: https://github.com/Big-BangBang/road_extraction/blob/f876382158ddca95b8709a1aabed39f4ae7ea551/README.md. [Accessed 10 December 2024].

[13] L. Gao, J. Wang, Q. Wang, W. Shi, J. Zheng, H. Gan, Z. Lv and H. Qiao, "Road Extraction Using a Dual Attention Dilated-LinkNet Based on Satellite Images and Floating Vehicle Trajectory Data," *IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING*, vol. 14, 2021.

Our code for D-LinkNet Model is available on our GitHub repository, which can be accessed at https://github.com/KshKsh0/project-with-gitba.