CS-433 Machine Learning Project 2 - Road Segmentation

Ilieva Nadezhda, Krsteski Stefan, Tashkovska Matea
Department of Computer Science, EPFL, Switzerland

***Abstract*—Road segmentation is a key challenge in digital image processing and computer vision and it plays an important role for numerous applications, such as driver assistance, traffic surveillance and vehicle guidance. The aim of this project is to develop a model that can accurately segment roads from satellite images. To address this, we employ two models renowned for their proficiency in this domain: U-Net and DeepLabV3. Additionally, we enrich our primary AIcrowd dataset with two external datasets. Our approach also involves preprocessing, normalization, data augmentation, and postprocessing to enhance models' performance. Our top-performing model reached an F1-score of 0.923, highlighting the effectiveness of the DeepLabV3 model, the use of varied datasets, and sophisticated data processing methods for the satellite images.**

I. Introduction

In the evolving landscape of digital image processing and computer vision, image segmentation stands as a challenging problem. Road segmentation involves identifying and isolating parts of an image that represent roads. This project focuses on segmenting roads from satellite images, which is crucial for various applications like driver assistance, traffic surveillance, and vehicle guidance. More specifically, we propose a comprehensive approach to address this problem by utilizing state-of-the-art architectures such as U-Net and DeepLabV3. These models are widely recognized for their effectiveness in image segmentation tasks, making them well-suited for our objective. Our goal is to provide valuable insights and advancements in the field of satellite image analysis through the utilization and comparison of these architectures with diverse datasets.

II. Data

This section presents a detailed overview of the datasets we use in our project. This includes the primary dataset obtained from AIcrowd, as well as two external ones.

A. AIcrowd Dataset

The train dataset comprises 100 RGB satelite images with a resolution of 400×400 pixels. These images are provided with corresponding ground truth masks where road pixels are labeled as 1 and non-road pixels as 0. It is important to note that some pixels in the masks have values between 0 and 255, rather than being strictly 0 or 255, which are later adjusted for more accurate classification. The test dataset contains 50 RGB satelite images at a higher resolution of 608×608 pixels.

B. External Datasets

1) Massachusetts Roads Dataset: In our project, we also make use of the Massachusetts Roads Dataset [1], which consists of 1500×1500 pixel RGB satellite images with their corresponding masks. First, we carefully selected 200 images from the Massachusetts Roads Dataset that are visually similar to our primary dataset. To adjust to the size of the images in the AIcrowd dataset, we split each of the selections into 9 smaller segments. Finally, we reviewed these segments and eliminated the ones that had little or no road, resulting in a total of 1333 useful images.

2) Kaggle Dataset: Additionally, we utilize a dataset from Kaggle [2], that consists of 400×400 pixel RGB images of two American cities - Boston and Los Angeles. These images are extracted using the Google Maps API. We kept only the satellite images from Los Angeles, since we found them to be more visually similar to our primary dataset. Moreover, we kept only the images with at least 10% roads. This enriched our data with an additional 7470 images.

III. Data Processing

In this section, we outline the various data processing steps used in our project, including preprocessing, image normalization, data augmentation, and postprocessing, which are crucial for improving our models' performance.

A. Preprocessing

Since some of the ground truth images had pixels that were not purely black (0) or white (255), we initially focused on transforming them into a binary format. To resolve this, we employed a thresholding approach: pixels with values below 127 were converted to 0 (non-road), while the rest were turned to 255 (road). This eliminated the gray pixels and ensured a clear binary distinction in our ground truth images.

B. Data augmentation

In our project, we experimented with different data augmentation techniques to increase the robustness and diversity of our model. This involved applying a series of random transformations to both the satellite images and their corresponding ground truth masks, ensuring that these modifications were consistently mirrored across both of them. Transformations in our data augmentation process included random resized crop, along with random horizontal and vertical flips, as well as random rotation. For the images alone, we experimented with color jitter, adjusting brightness, contrast, saturation, and hue to represent different lighting conditions.

C. *Image normalization*

To ensure that the pixel values of the images were on a consistent scale, we applied image normalization. We normalize each channel independently, based on the mean and standard deviation across images in the training data.

D. *Postprocessing*

To improve the quality of the predicted labels, we used traditional image processing techniques from the family of morphological operations, such as erosion and opening. Morphological erosion shrinks bright regions and enlarges dark ones, whereas morphological opening removes bright spots (reffered to as 'salt') and connects small dark cracks. We experimented with various combinations of shapes and sizes for the structuring elements (or kernels), enabling us to achieve optimal image clarity via thinning lines and eliminating isolated white spots.

IV. Models and Methods

In this section, we describe the various models and methods employed in our project. Initially, we established a baseline using a Convolutional Neural Network (CNN) model, by modifying the provided script from the project description. This baseline served as a reference point for subsequent experiments.

A. *U-Net*

U-Net, originally developed for biomedical image segmentation [3], has gained popularity in various image segmentation tasks due to its effective design. As illustrated in Figure (1), the U-Net architecture is composed of three key components: the encoder, the bottleneck and the decoder.

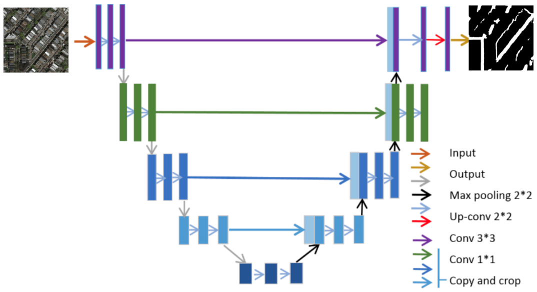


Figure 1: U-Net Architecture

The encoder is responsible for capturing the context of the input image. This part of the network consists of several convolutional layers, each followed by a pooling layer. The convolutional layers help in extracting feature maps from the image, while the pooling layers progressively reduce the spatial dimensions of these feature maps. This approach allows the encoder to extract important features from the image, focusing on capturing broad contextual information.

The bottleneck is the transition zone between the encoder and decoder paths. It contains a few convolutional layers that process the deepest, most abstract representations of the input data. In the context of road detection, the bottleneck aids in synthesizing the high-level features extracted by the encoder, serving as a component for understanding complex patterns, such as differentiating roads from other similar structures, such as houses with gray roof.

The decoder, or the expansion path, follows the bottleneck. Its primary function is to gradually recover the spatial resolution that was reduced by the encoder. The decoder uses a series of up-sampling layers, which increase the dimensions of the feature maps.

The U-Net architecture incorporates skip connections, a feature that connects layers in the encoder to corresponding layers in the decoder. These connections facilitate the transfer of fine-grained details and spatial information directly across the network. Moreover, skip connections mitigate the vanishing gradient problem by providing an alternate pathway for the gradient flow during backpropagation.

More specifically we used the U-Net model from the `segmentation_models_pytorch` library [4]. We chose this model due to its flexibility and the availability of various encoders. We employed EfficientNetB3 [5] as an encoder, taking advantage of pre-trained ImageNet weights. This approach allowed us to leverage the power of transfer learning, utilizing a network trained on a diverse dataset to improve our model's performance on satellite images.

B. *DeepLabV3*

After extensive experimentation with U-Net architectures, we shifted our focus to DeepLabV3 [6] with a ResNet50 [7] as a backbone. DeepLab models, are a series of deep learning architectures designed to tackle the problem of semantic segmentation. The architecture is shown in Figure (2).

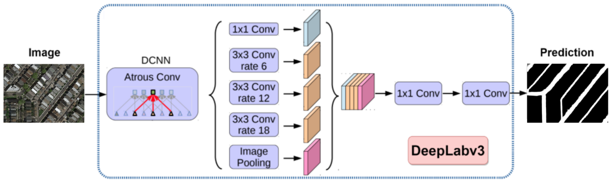


Figure 2: DeepLabV3 Architecture

The process begins with an input image passing through a deep convolutional neural network (DCNN), in our case ResNet50, which is responsible for extracting features. Early layers detect simple elements like edges, while deeper layers recognize more complex patterns.

The network then uses atrous (dilated) convolutions. Atrous convolutions are a specialized type of convolution that allows the network to capture a broader context. They have a wider field of view, allowing the network to integrate information from a larger portion of the image without losing resolution. Unlike regular convolutions, atrous convolutions have a larger receptive field. They can cover a broader area of the input without increasing the number of parameters. This is done by inserting 'holes' between the pixels in the convolutional filter. For example, a 3×3 atrous convolution might cover the same area as a 5×5 regular convolution but still only use 9 parameters. The comparison between the two is shown in Figure (3).

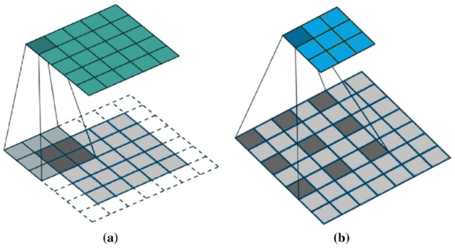


Figure 3: Normal Convolution vs. Atrous Convolution

The following component is the Atrous Spatial Pyramid Pooling (ASPP) module. This component applies atrous convolutions at different scales, enabling the network to capture details ranging from very fine to broader aspects. ASPP resembles a set of different magnifying glasses, each revealing different levels of detail - some show very fine details like road width, while others reveal the bigger picture like neighborhood layouts. This view is critical for accurately identifying elements like roads, buildings, and trees in satellite photos.

The features gathered at these multiple scales are then merged using 1x1 convolutions. This step combines the detailed and broader context information into a unified feature map. This unified feature map is used to generate the segmentation map, which classifies each pixel into different categories. The map is up-sampled to match the original image's dimensions.

Specifically we used the implementation by PyTorch [8], which can be found at the official PyTorch website. Additionally, we leveraged the pre-trained DeepLabV3 model with weights trained on the Pascal VOC 2012 dataset [9].

V. Experimental Setup

We divided the dataset into training and validation sets, with 80% for training and 20% for validation. For testing our models' performance, we used the test dataset and submitted our predictions on the AIcrowd platform. The models' performance was primarily evaluated using the F1 score. In our training process we used the Binary Cross-Entropy with Logits Loss. The models were trained using the Adam optimizer with a learning rate of 3e-5. To convert the model outputs to binary predictions, a Sigmoid function was applied, with a threshold set at 0.6, which was determined empirically based on performance metrics observed on the validation set. We utilized Wandb [10] to track all of our experiments in order to easily analyze and organize them.

VI. Results

Table (I) shows the F1-scores of different models evaluated on the AIcrowd platform. As expected, the baseline CNN model gave us unsatisfactory results which we used as a reference for further improvements. The introduction of the UNet architecture with an EfficientNet encoder, significantly improved our results, with an F1-score of 0.860. This result highlights the UNet's superiority in handling the complexities of image segmentation tasks. Adding data augmentation to the images, specifically horizontal and vertical flips and random rotations up to 10 degrees, further improved the F1-score. This due to the increased diversity and robustness of the training dataset, which allows the model to learn more general features. Furthermore, the final stride in our experiments with the UNet architecture involved applying postprocessing techniques on the model trained with data augmentation. These postprocessing techniques included morphological operations, more specifically an opening with a 3×3 square structuring element, followed by an erosion with a 9×9 square structuring element. This resulted in an F1-score of 0.891. The use of morphological image processing methods resulted in a more precise and coherent representation of roads by removing noise, smoothing edges, and filling in gaps in the segmented areas.

Model	**F1-score**
Baseline CNN	0.541
UNet	0.860
UNet w/ Data Augment.	0.880
UNet w/ Data Augment. & Postprocess.	0.891

Table I: F1-score of the baseline and UNet variants' performance tested on AIcrowd

In pursuit of enhanced performance beyond what was achieved with UNet, our attention shifted to the DeepLabV3 architecture in order to further improve our results.

The results in Table (II) showcase the F1-scores for various DeepLabV3 models trained on different datasets and evaluated on the AIcrowd platform. All results presented in the table were obtained using data augmentation and postprocessing. The augmentation process included horizontal and vertical flips, as well as random rotations up to 10 degrees. Similarly as before, the postprocessing involved morphological operations, as applying an opening with a 3×3 square structuring element, followed by an erosion step using a larger 9×9 square element. The initial implementation of the DeepLabV3 model achieved an F1-score of 0.898, which is an improvement over the best results obtained with the UNet model. Because of the better performance, we decided to continue developing and optimizing the DeepLabV3 architecture in our next experiments.

Model	**F1-score**
DeepLabV3 (AIcrowd dataset only)	0.898
DeepLabV3 (w/ AIcrowd & Massach.)	0.900
DeepLabV3 (w/ AIcrowd, Massach. & Kaggle)	0.923

Table II: F1-score comparison of DeepLabV3 models trained on different datasets and tested on AIcrowd

All models that utilize external datasets went through a two-phase training process. First, we trained each model on both the external dataset(s) and our main AIcrowd dataset. Then, we took this trained model and further fine-tuned it using only the AIcrowd dataset images. This step-by-step training method enabled us to adjust the model to our main dataset, while also benefiting from the variety and diverse characteristics in the external datasets. The integration of the Massachusetts dataset marked an improvement in the model's performance and achieved an F1-score of 0.900. This indicates that the Massachusetts dataset contributes valuable data for our model. More specifically, we saw an improvement in the model's performance especially in its ability to accurately predict curvier and thinner roads, which were not as common in the train dataset from AIcrowd. An even more significant increase is observed when we used both the Massachusetts and the Kaggle datasets combined with the train images from AIcrowd giving us the best F1-score of 0.923. This improvement shows the importance of diverse and comprehensive training data in enhancing our model's generalization capabilities and developing a highly accurate model.

In Figure (4), we show a pair of satellite images from the validation set along with their corresponding ground truth data and the predictions generated by our model. Comparing our model's predictions with the ground-truth images, we can observe that our model's predictions align quite well with the actual road layouts. This indicates that our model effectively distinguishes the roads from their surroundings.

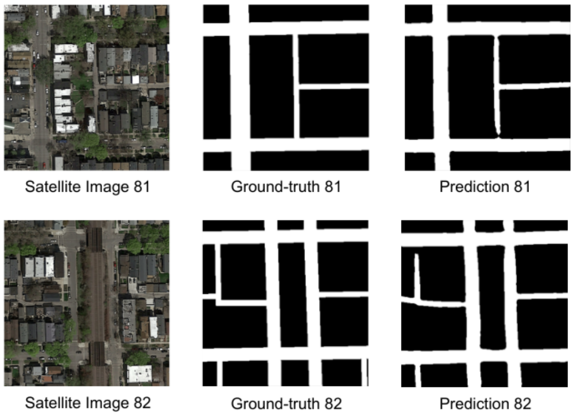


Figure 4: Road Segmentation Comparison - Satellite image, Ground-truth and Model prediction

VII. Discussion

The most significant improvement in our project came from the adoption of the DeepLabV3 architecture with ResNet50 as backbone, particularly when trained on a combination of the AIcrowd, Massachusetts, and Kaggle datasets. This model demonstrated the importance of diverse training data in improving accuracy and generalization capabilities. Another critical aspect was the incorporation of postprocessing techniques and data augmentation. These methods significantly enhanced the quality of predictions by refining the segmentation maps and increasing the robustness of the model against variations in satellite images. The use of morphological operations like opening and erosion proved effective in cleaning up the segmented road images and led to more precise prediction.

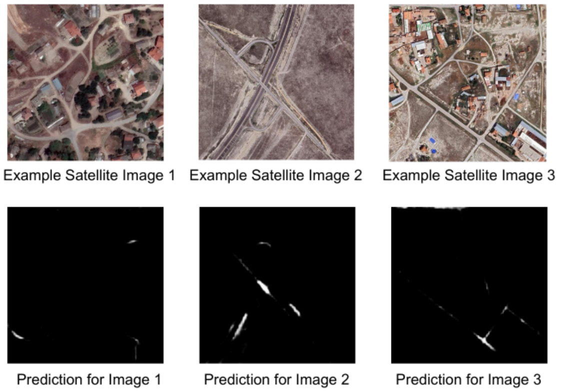
While advanced models like DeepLabV3 showed promising results, they also come with higher computational demands. This aspect poses a challenge, especially in scenarios where resources are limited. Future work could explore optimizing these models for greater efficiency or investigating other architectures that balance performance with computational load. Future studies could also focus on expanding the dataset with images from different geographical locations and conditions to further enhance the model's generalization capabilities. Additionally, incorporating Focal Loss as an improvement to our model could be beneficial. Focal Loss is an advanced variant of the standard cross-entropy loss, designed to address class imbalance by focusing more on difficult, misclassified cases. Such an adaptation could enhance the model's ability to accurately segment roads in complex satellite images, potentially leading to further improvements in performance.

VIII. Ethical Risk Assessment

In this section we describe a potential application of our trained model and we analyse one ethical risk that comes with it. Namely, navigation systems and map applications could utilize road segmentation models to improve routing and estimate travel times. The intended end-users in such use cases would include, but not be limited to: individual drivers and commuters, commercial drivers, delivery services, and emergency responders.

There are several risks that could arise, including privacy and security risks, as well as dependence on quality and up-to-date satellite data. However, in this analysis, we will focus on the risks related to bias and fairness. One potential risk we have identified is related to the training data. The model may not accurately identify roads in certain areas, which could result in a system malfunction. Specifically, if some regions are underrepresented or not included in the training set at all, the model's performance would be poor, rendering the application useless. In our specific use case, users of the application may experience potentially incorrect, non-existent, or less efficient navigation route suggestions.

To evaluate the risk, we conducted research on the training data and compared the model's performance. We found that the training data used for the road segmentation model was biased towards urban areas and lacked diversity in terms of geographical locations and road types. As a result, the model is more likely to inaccurately identify roads in rural or underrepresented areas. To validate this, we tested the model's performance on a manually selected sample of satellite images. As depicted in Figure (5), even without comprehensive metrics, we can visually confirm a significant degradation in performance.


Example Satellite Image 1 Example Satellite Image 2 Example Satellite Image 3

Prediction for Image 1 Prediction for Image 2 Prediction for Image 3

Figure 5: Performance on underrepresented areas

Ideally, one would include a diverse training set to mitigate this issue. However, due to time and resource constraints, it was not possible to collect and label a sufficiently diverse dataset. As a result, we acknowledge the limitations of the model in accurately identifying roads in underrepresented areas and the potential negative impact it may have on users in those regions. To address this, one could include a disclaimer in the application, which would notify users about the limitations of the model and encourage them to use their own judgment when following navigation suggestions in areas with limited training data coverage.

References

[1] V. Mnih, "Machine learning for aerial image labeling," Ph.D. dissertation, University of Toronto, 2013.

[2] L. Timoth, "Road segmentation - Boston & Los Angeles," https://www.kaggle.com/datasets/timothlaborie/roadsegmentation-boston-losangeles, 2023, Kaggle.

[3] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 2015, pp. 234–241.

[4] Segmentation Models PyTorch, "Unet," 2023, [Online; accessed 21-12-2023]. [Online]. Available: https://segmentation-modelspytorch.readthedocs.io/en/latest/docs/api.html

[5] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," 2019, cite arxiv:1905.11946Comment: Published in ICML 2019. [Online]. Available: http://arxiv.org/abs/1905.11946

[6] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR '16. IEEE, Jun. 2016, pp. 770–778. [Online]. Available: http://ieeexplore.ieee.org/document/7780459

[8] PyTorch Contributors, "Deeplabv3," 2023, [Online; accessed 21-12-2023]. [Online]. Available: https://pytorch.org/vision/main/models/deeplabv3.html

[9] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results," http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[10] L. Biewald, "Experiment tracking with weights and biases," 2020, Software. [Online]. Available: https://www.wandb.com/