Road Segmentation of Aerial Images

Rasan Younis, Adam Rahmoun , Mamoun Chami, Eliot Ullmo
Kaggle group: Adam

Abstract

This paper presents a comprehensive study on image segmentation applied to satellite images obtained from Google Maps. Our goal is to classify each pixel with a probabilistic label indicating the presence of roads. For this purpose, we first augment the data by adding images and masks from different sources. We then developped various machine learning models, particularly focusing on Convolutional Neural Networks (CNNs) and advanced segmentation architectures like Dense-U-Net, PSPNet, and DeepLabV3+, while comparing them to baseline models. We got our best results by applying ensemble methods using those different models.

1 Introduction

Road segmentation from aerial satellite images is a crucial task in the field of computer vision, with significant applications in urban planning, autonomous driving, and geographic information systems. It involves dividing a digital image into several segments, each of which is made up of a collection of pixels, in order to identify the image's content and provide more straightforward examination of each segment.

In recent years, computer vision and machine learning models, have evolved significantly. This with the advancements in computing power and graphics processing units (GPUs), which enable a reduction in computational time, we are now witnessing remarkable success in image segmentation tasks. Machine learning uses Convolutional Neural Network designs to learn patterns in visual inputs and predict the classes of objects that make up an image.

In this project, image segmentation is applied to satellite images taken from Google Maps. Essentially, the task is to classify each pixel of the image with a probabilistic label p [0, 1], in order to detect which parts of the images are made of roads (p = 1), and which are made of background (p = 0). The goal is to develop a model that outputs a grayscale mask of the same size as the input image, containing the mentioned probabilities in each pixel.

We are given a training set of 144 images, along with the corresponding ground-truth masks and an evaluation set containing 144 images. Both the training and test data consist of images of size 400×400.

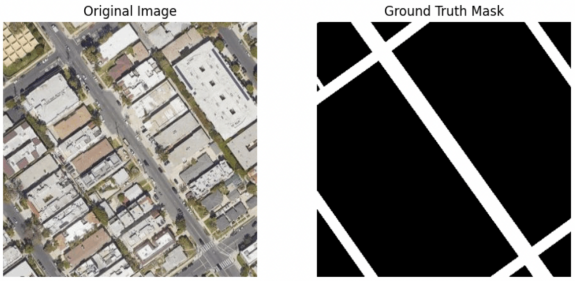

Figure 1: Satellite Image and Ground Truth Mask

2 Preprocessing

2.1 Data Augmentation

Our training dataset, as mentioned earlier, consists of 144 images. While this dataset is useful, it is quite small compared to the vast variability present in real-world road segmentation tasks. Roads can vary significantly in appearance due to differences in materials, weather conditions, lighting, and wear. Additionally, the background in satellite images can include various elements such as buildings, vegetation and more, which adds another layer of complexity that our model needs to address.

The inclusion of shadows is another factor that complicates the task. Objects like vehicles and buildings can hide or alter the appearance of roads, making it challenging for the model to accurately

detect road segments. Geographical diversity also poses a significant challenge. The training dataset might be biased towards certain geographical regions, missing out on the global diversity of road structures and environments. This can lead to a model that is not robust across different geographical contexts.

Given all these challenges, it is evident that our original dataset is not sufficient to tackle the problem effectively. Therefore, we must employ data augmentation techniques to enhance the training dataset's variability and improve the model's robustness. Data augmentation allows us to artificially increase the size and diversity of our dataset by applying various transformations, thereby helping the model generalize better to unseen data.

To further address the limitations of our initial dataset, we utilized the Google Maps API to gather additional images (Google Maps Platform, 2023), along with the DeepGlobe Road Extraction Dataset (Demir et al., 2018) and the Massachusetts Roads Dataset (Mnih, 2013). See table 3 to see the improvement.

2.2 Image processing

First, as the initial dataset is the most similar to the dataset we want to predict, we apply various transformations to fully leverage its data. We apply horizontal and vertical flips, along with an affine transformation.

Then, we split the other datasets in half. For the first half, we crop both the images and the masks to a consistent size. This step involves finding the crop that contains the most non-black pixels in the mask, ensuring that the regions of interest are preserved. And for the second half, we simply resize the image to a common size with all other images. We do that to simulate the smaller roads, since the other datasets' images are of higher resolution than the initial provided dataset.

At the end, we normalize every pixel value to the range [0, 1].

3 Architecture experiments

We explored various model architectures to identify the most effective approach. Our experimentation process was iterative with each step building on the insights gained from previous models. In this section, we present an overview of the different architectures we experimented with, highlighting the evolution of our design choices and the corresponding performance improvements.

3.1 Baseline models

3.1.1 Logistic Regression

For road segmentation, we observed that roads have a homogeneous color. To improve our logistic regression model, we extracted 12 features from each image patch: mean color values, variance of color values, mean Sobel edge magnitude, and variance of Sobel edge magnitude. These features, which combine color and texture information, improve segmentation accuracy.

Using logistic regression, we modeled the probability of a patch containing a road as:

$$P(y_i = 1) = \frac{1}{1 + e^{-x_i^\top \theta}}$$

We minimized the cross-entropy loss to find θ.

The process included loading data, splitting into training and validation sets, extracting patches and features, and training the model.

3.1.2 Patch CNN

Convolutional Neural Networks (CNNs) automatically identify key features in images, improving road segmentation accuracy compared to manual feature design.

We developed a CNN to process 16×16 image patches, using convolutional layers with ReLU activations, max-pooling, batch normalization, and dropout for regularization. Fully connected layers and a sigmoid activation function perform binary classification.

Although using patches limits the model to local features, this baseline demonstrates a good potential of CNNs and motivates exploring more advanced CNN architectures in future work.

3.2 UNET ++

UNet++ is a sophisticated segmentation architecture created to boost feature representation and guarantee stronger semantic continuity between the encoder and decoder sub-networks, hence improving the performance of image segmentation tasks. UNet++'s main novelty is the use of dense and nested skip connections, which are intended to close the semantic gap between the encoder and decoder feature maps (Zhou et al., 2018). The dense skip connections and nested architecture ensure that the model captures detailed features at multiple semantic levels, which is crucial for identifying complex road structures.

3.3 Dense-U-Net

Our second approach consisted in using a hybrid model architecture that combines DenseNet as the encoder with the U-Net framework (Ronneberger et al., 2015) for the decoder. The DenseNet architecture (Huang et al., 2017) is known for its efficient feature propagation and reduced number of parameters due to its dense connections, which enhance gradient flow and resolves the vanishing gradient problem.

The U-Net structure, integrated with DenseNet, uses upsampling and skip connections, allowing the model to capture fine-grained details by combining features from different levels of the network. To improve the performance of our model, we trained it on an expanded dataset comprising approximately 4000 images sourced from diverse geographical locations. It significantly contributed to the robustness of our model.

3.4 PSP Net

The Pyramid Scene Parsing Network (PSP Net) (Zhao et al., 2017) is a powerful semantic segmentation model which innovates thanks to its so-called Pyramide Pooling Module (PPM). The PPM's capacity to leverage the input feature map at different scales, through multiple pooling layers, allows to capture both local and global context across the input image. This is very practical for our use-case, as initial architectures had a hard-time identifying the thinner roads on satelite images. We trained the model on the complete dataset using the DiceLoss. The prediction results demonstrated that the model performed confidently in segmenting large roads. However, while it still struggled to fully segment thin roads, it was able to assign low probabilities to some pixel regions corresponding to these thin roads. This indicates a modest improvement in the model's structural understanding of road patterns.

3.5 DeepLabV3+

In order to get an even better grasp of the thin roads, we then tested an architecture based on the DeepLabV3+ deep convolutional semantic segmentation model (Chen et al., 2018). We used ResNet-152 as the backbone model and initialized the model with pre-trained weights from Imagenet (Deng et al., 2009).

Because of the model's Atrous Spatial Pyramid Pooling module (ASPP), we can capture more fine-grained information and improve the segmentation of thinner roads. Just like we expected, prediction results yielded larger regions of pixels on thin roads with higher probabilities. On top of that, this architecture segments slightly better the larger roads.

We train the model on all the datasets together, using as a loss function the DiceLoss, as the number of road pixels are scarce compared to the background pixels. We use Adam optimizer and during training keep track of important metrics to our problem such as Precision, Recall and IoU.

3.6 Ensemble methods

To further improve upon our original models, we decided to leverage the strengths of some of the best models we trained. By combining them, we aim to enhance the overall segmentation performance. This strategy allows each model to focus on its areas of expertise, thereby optimizing the training process and improving the robustness and accuracy of the final predictions.

To do so, we optimize 2 different kinds of parameters, each model's weights for the final stacked model, and the threshold that will be determining if the pixel is a road or not. We use Optuna (Akiba et al., 2019) to initialize a study that we let run for 50 trials to get the optimal parameters. Then, to get the final mask output, we sum the models' predictions multiplied by the weights that we then apply the threshold to.

4 Post-processing

We tested different post-processing methods to further improve upon our model, however, most of the experiments ended up actually lowering the model's f1-score. The issue is that most post processing methods are useful to refine the output of the model further, it helps making the lines sharper for example. However, the weakness of the model doesn't lie within the bluriness of the output, but more within its difficulty to detect the smaller routes that are in the mask, but that are also difficult to detect with the naked eye. The methods we used are median blurring, gaussian blurring, removing small connected objects, small kernel erosions (Yerram et al., 2022). We also used thresholding to convert the probability-based output of the segmentation model into a binary mask suitable for submission. Both manual and automatic thresholding methods were evaluated. Specifically, we tested Otsu's method (Otsu, 1979), which automatically determines the optimal threshold value from

the image histogram, and Li's iterative Minimum Cross Entropy method (Li and Tam, 1998), which iteratively finds the threshold that minimizes the cross-entropy between the foreground and background.

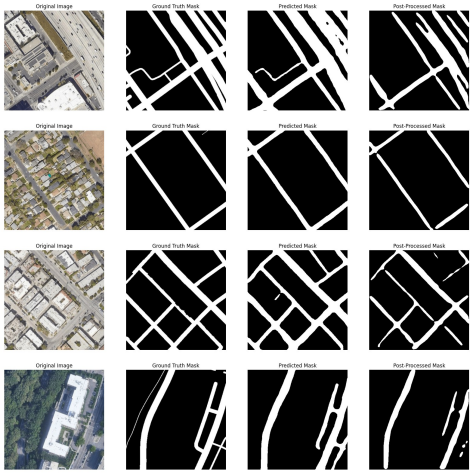

Figure 2: Satellite image, mask, predicted mask, and post-processed predicted mask using median blurring

5 Results

The results are calculated locally

Table 1: F1 Scores of Baseline Segmentation Models

Model	F1 Score
Logistic Regression	0.47
Patch CNN	0.58

Table 2: F1 Scores of New Segmentation Models with Data Augmentation

Model	F1 Score
UNet++	0.70
DenseUNet	0.69
PSP Net	0.69
DeepLabV3+	0.85

6 Discussion and Conclusion

Among our different models, the one that clearly outperformed the others is DeepLabv3+, even applying the ensemble method only slightly improved its performances. It was what we also expected since it is a model that was already proved to be very efficient in the task of Road Extraction.

The major difficulty in this particular competition was that the mask provided for images was sometimes a bit arbitrary, in the sense that some small roads will be considered as roads in the mask and sometimes not, for example in parkings.

Table 3: F1 Scores of DeepLabv3+ with changes

Change	F1 Score
With Ensemble Method	0.858
With Data Augmentation	0.853
Without Data Augmentation	0.735

Table 4: F1 Scores of DeepLabv3+ with different post-processing methods

Change	F1 Score
Median blurring	0.767
Gaussian Blurring	0.775
Removing small connected objects	0.774
Small kernel erosion	0.701

Because of the scoring metric for this competition, applying other types of post-processing apart from thresholding doesn't improve the model much. Since most post-processing methods are only useful for removing blurriness of predictions, staright-ening lines among others.

Data Augmentation is a major component of our experiments. As the data we were initially provided with was scarce, using outside datasets and applying transformations to the original dataset was a must. Training the model on an even bigger dataset should further improve upon the results.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework.

Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation.

Ilke Demir, Krzysztof Koperski, David Lindenbaum, Guan Pang, Jing Huang, Saikat Basu, Forest Hughes, Devis Tuia, and Ramesh Raskar. 2018. Deepglobe 2018: A challenge to parse the earth through satellite images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Google Maps Platform. 2023. Google maps platform. Accessed: 2023-07-29.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708.

CH Li and Peter Kwong-Shun Tam. 1998. An iterative algorithm for minimum cross entropy thresholding. *Pattern recognition letters*, 19(8):771–776.

Volodymyr Mnih. 2013. *Machine Learning for Aerial Image Labeling*. Ph.D. thesis, University of Toronto.

Nobuyuki Otsu. 1979. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241.

V. Yerram, H. Takeshita, Y. Iwahori, Y. Hayashi, M.K. Bhuyan, S. Fukui, B. Kijsirikul, and A. Wang. 2022. Extraction and calculation of roadway area from satellite images using improved deep learning model and post-processing. *J. Imaging*, 8(5):124.

Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. 2017. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890.

Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. 2018. Unet++: A nested u-net architecture for medical image segmentation.