Exploration of different CNN approaches for Satellite Imagery Road Segmentation

Lucas Falardi, Max Krähenmann, Enea Peter, Virgilio Strozzi
Group: FibLouis
Department of Computer Science, ETH Zurich, Switzerland

***Abstract*—This project endeavors to explore novel techniques in the domain of computer vision and Convolutional Neural Networks (CNNs). Within this paper, we present ZoomNet and Ensemble Bootstrapping, two CNN-ensemble-based approaches for the task of aerial image road segmentation. ZoomNet combines multiple same model architectures trained on different sizes (zoom levels) of image data, while Ensemble Bootstrapping trains together different models and recursively add each averaged prediction as an extra channel to train the same models in new epochs. We analyze the performances achieved by these methods on the public Kaggle competition score [1], by training them on two different datasets (ETH-Data and Map-Data). Moreover we compare our new approaches to 6 current state-of-the-art models.**

I. Introduction

High-resolution aerial images serve as valuable sources of information for numerous applications, including urban planning, traffic management, mapping, and navigation systems. However, the conventional manual processing of this data is not scalable due to the overwhelming workload, particularly when dealing with vast areas. Traditional image processing techniques, such as filtering or edge detection [2], can be utilized, but they demand specialized knowledge and are highly sensitive to specific datasets and variations. In contrast, Convolutional Neural Networks (CNNs) have emerged as a dominant and efficient technique for computer vision tasks, including the processing of aerial images.

This paper focuses on the task of image road segmentation using CNNs. The main objective is to create an accurate pixel-wise mask for aerial images, effectively distinguishing between road and non-road areas. This information can be further combined with other data, for instance, to produce precise territorial maps.

Aerial image road segmentation presents various challenges, including complex road networks with varying geometries, the coexistence of multiple road types constructed from different materials, resulting in distinct appearances on the images. Additionally, the resemblance between roads and other linear structures, such as rivers, railroads, or the diverse landscape, adds to the complexity of this task. Numerous researches and approaches have been attempted in recent years, such as [3], [4], [5]. In this paper, we present a comparison of the results of our new proposals with different baselines, contributing to the objective of improving the efficiency and precision of road segmentation and providing interesting ideas for newer works.

II. Models and Methods

A. Dataset

The data provided for this project consisted of 288 400×400 pixel RGB satellite images acquired from Google Maps [6]. For half of them (144 images), a 400x400 ground truth mask was also provided. This mask assigned a value $v \in [0,1]$ to each pixel, representing the probability of the pixel belonging to road area ($v = 1$) or background ($v = 0$). This set of images serves as a training set.

As for the other half of the images, referred to as the test set, no ground truth is provided. The objective is to predict whether each patch of 16x16 pixels of the image contains at least 25% of pixels representing the road area.

B. Data Acquisition & Preprocessing

Given that the provided data is very limited, we collect additional training data from Google Maps instead of relying on data augmentation. We study the given areal images and assume that they are taken from major US cities. Therefore, we select 13 of the largest cities in the US to add data from. The selected cities are Boston, Los Angeles, Washington, Houston, Philadelphia, New York, Chicago, Phoenix, Jacksonville, Columbus, Charlotte, Seattle, and Indianapolis. We select the cities to represent different geographical regions of the USA. Subsequently and we retrieve between 1500 and 6000 images per city using the Google Maps API.

For each city, a rectangle area is defined to collect images, either including all images contained in the square or skipping some images, depending on the city's size. This is done to maintain comparability in the number of images between cities. Each retrieved image consist of both the satellite view and a corresponding map image. To ensure consistency with the already provided data, the map images are customized using a specific map style through the Google Cloud Platform. The map style is designed to color all streets in black and leave the rest of the area in white. To use the map images, the colors have to be inverted to match the ground truth images provided.

An example of the images gathered can be observed in Figure-1. To ensure quality, only images where at least 10% of the area consist of street pixels are retained. The zoom level is also adjusted to be consistent with the training images (zoom level 18). All collected images are of size 400x400x3 pixels. In total, the dataset created from Google

Maps (**Map-Data**) images consist of 43926 satellite images and their respective map images. These images from Google Maps are used to train the models, while the data provided by the project serves as the validation set. Some of the validation images are retained for the final test set. Ultimately, all the images provided by the project are utilized to fine-tune our models. We also make use of the provided images to train another copy of each model to provide a baseline and an argument of our choice to gather more data.
Apart from the preprocessing described for the collected images, no further preprocessing is conducted on the given data.

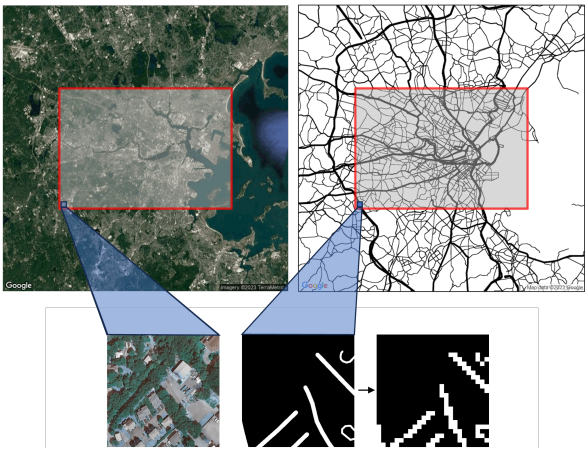

Figure 1: Data Generation from Satellite Imagery and Maps Data from Google Maps. The figure illustrates the process of obtaining satellite images and map images for the same location, both saved as 400x400 pixel images. The map images are generated with a personalized style and require color inversion to serve as ground truth for their respective satellite images. Moreover, from the resulting ground truth, additional images are created with lower resolutions. In these images, 16x16 patches are colored white only if at least 25% of the pixels from the ground truth image are also white.

C. Baseline Models

In order to assess the performances of our proposed models, we provide a comparison with some of the models that have been used on segmentation tasks in the past, and some state-of-the-art models that currently get deployed. Based on the resources that we had, we choose to evaluate the performance of the following models:

- **UNet** [7]: The architecture consists of a contracting path to capture context and a symmetric expanding path that enables precise localization. Furthermore skip connections are added between the encoder and the decoder and play a crucial role in enabling efficient and effective information flow during the process of image segmentation. The architecture employed here consists of 4 encoder and decoder layers.
- **UNet++** [8]: A more powerful version of the UNet. The architecture is essentially a deeply-supervised encoder-decoder network where the encoder and decoder sub-networks are connected through a series of nested, dense skip pathways. These re-designed skip pathways aim at reducing the semantic gap between the feature maps of the encoder and decoder sub-networks.
- **LinkNet** [9]: The architecture follows the philosophy of the UNet, but extends on it by using residual network blocks for the encoder.
- **PAN** [10]: This architecture leverages a pyramid pooling module to capture multi-scale contextual information from the input feature maps, allowing the network to have a broader perception of the scene. Additionally, PAN introduces an attention mechanism that recalibrates the feature maps to focus on more relevant regions, enhancing the discriminative power and improving the segmentation accuracy.
- **MiniUNet** [7]: The architecture is the same as the one of the UNet, but the input size is 208x208x3. Therefore, the model is trained only on quarters (overlapping with each others on 8 pixels) of the original 400x400x3 image and the inference is performed by unifying the quarters predictions for each picture.
- **DeepLabv3+** [11]: Expanding on the DeepLabv3 architecture, which deploys spatial pyramid pooling, DeepLabv3+ adds a decoder module. This allows to recover more detailed object boundaries.
- **SegFormer (MIT-B0)** [12]: A tranformer based architecture, which consists of a hierarchical Transformer encoder and a simple MLP decoder head, without attention-mechanism. The model is pre-trained on ImageNet-1k.

D. Proposed Models

ZoomNet: The architecture can be visualized in Figure 2. The idea of the model is to combine the weighted predictions of two same UNet's architectures, trained separately on different input sizes. The specific architectures are the same UNet (input size 400x400x3) and MiniUNet (collage of the four quarters of original image of input size 208x208x3) of the previous section II-C. The output is then evaluated depending on the hyperparameter $\alpha \in [0,1]$ as follow: $ZoomNet(in) = \alpha * MiniUNet(in) + (1-\alpha) * UNet(in)$
The main motivation behind this architecture is that the two UNets should learn slightly different ways to predict the label for each image, since the MiniUNet only has access to a quarter of the original image at each prediction (a different zoom level) while the UNet has a more wide view having access to the full image. The advantage of such architecture is that different same architectures trained on different smaller zoom-levels can be used together to produce a more accurate prediction of a wider image, therefore allowing re-utilization of existing models and architectures at the cost of a slower inference and a preprocessing cut in pieces of the original dataset.

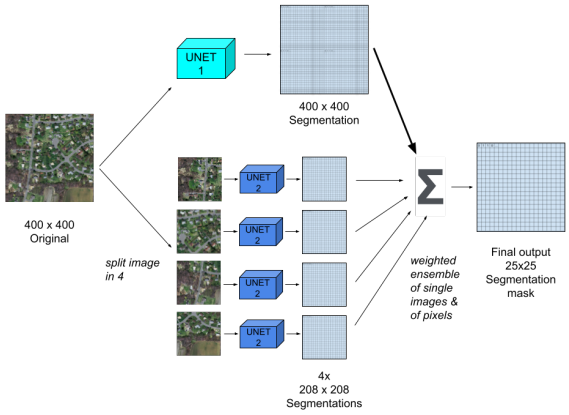

Figure 2: Sketch of the ZoomNet network. We distinguish the two UNet models, one in light blue and one in dark blue

Ensemble Bootstrapping: This approach doesn't specifically refer to any architecture, but can be seen as a technique to enhance the performance of singular baseline models. The idea can be inferred from Figure 3 and looks as follow: first a set of different models is trained, with the goal of producing accurate predictions on the training data. Once these predictions are produced, an ensemble is created and the predictions are averaged. The averaged predictions then get added as an extra channel to the input images, and the models get trained again. Once trained, the same process can be used for further rounds of training. The idea behind this concept is to try and force the models to learn higher level features by providing them with an easy guess about the general structure that the predictions should have.

One can also apply this concept on a model scale, by simply adding the predictions of a model to the input of the next training round, the goal being to essentially refine the predictions and learn more details.

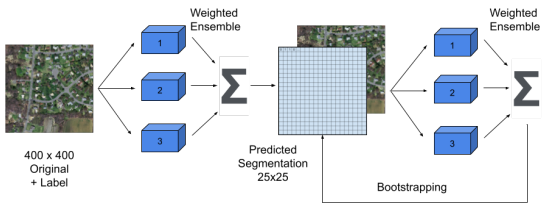

Figure 3: Sketch of the Ensemble Bootstrapping process. The results of different models are averaged and then used again to train the models

E. Training

Depending on the architecture of each model presented in the section II-C and the section II-D we employ different strategies. First we distinguish two types of train for each model: the first is made on the labeled dataset provided in the task (**ETH-Data**), containing 144 images, with a split of train 0.9 and validation 0.1, and the second on the 43926 images augmented dataset (**Map-Data**) using the *ETH-Data* as a validation set.

During both types of training, we minimize the *Dice Loss* on the prediction of each model and we keep the model's checkpoint with the smallest loss on the validation dataset across the epochs of training.

To allow reproducibility, we fix the seed of each possible generator to 420. Moreover we make use of the Adam Optimizer [13] for all of the models and set the initial learning rate to 0.0001 for all the models. The SegFormer makes an exception, where an initial learning rate of 0.00006 is used as suggested in the original paper [12]. We fix a batchsize of size 16 for each model.

When training the models on the *Map-Data*, we use 15 total epochs for all the models and finetune for 20 epochs on *ETH-Data*. The MiniUNet model is an exception, where the number of total epochs is set to 5 due to the natural larger size (4 times) of the training dataset and it is finetuned for only 5 epochs on the *ETH-Data* for the same reason. When training the models on the *ETH-Data*, we simply train across 40 total epochs.

All the meaningful hyperparameters of the models are derived empirically and reported on the results.

F. Post Processing

As a post process for the output of each model, we split the output labels of each model in 625 patches of 16x16 pixels and evaluate a mean m of the pixel values $p_{i,j} \in [0, 1]$ for each patch. If $m > 0.25$ then all the pixels in the patch are set to a value 1, otherwise 0.

This postprocess strictly follows the task description of the problem.

III. Results

We show the results of our models in the Table I. Along the proposed models we test six baselines. Every model is trained and tested on Kaggle on both the given dataset (*ETH-Data*) and the augmented dataset (*Map-Data*). The *Ensemble* entry relates to a majority voting ensemble of the six baselines. In the Ensemble Bootstrapping section, we report both the scores of the ensemble as well as the scores of the single models trained alongside in this approach. The number of parameters of the ZoomNet and the Ensembles are omitted since we make use of multiple neural networks. Lastly, we highlighted on the table the best scores in both datasets.

In Figure-4 we visualize two examples of the segmentation results produced by different models of the ZoomNet approach.

IV. Discussion and Conclusion

Road segmentation presents inherent challenges, even for human classifiers, as it is not always straightforward to

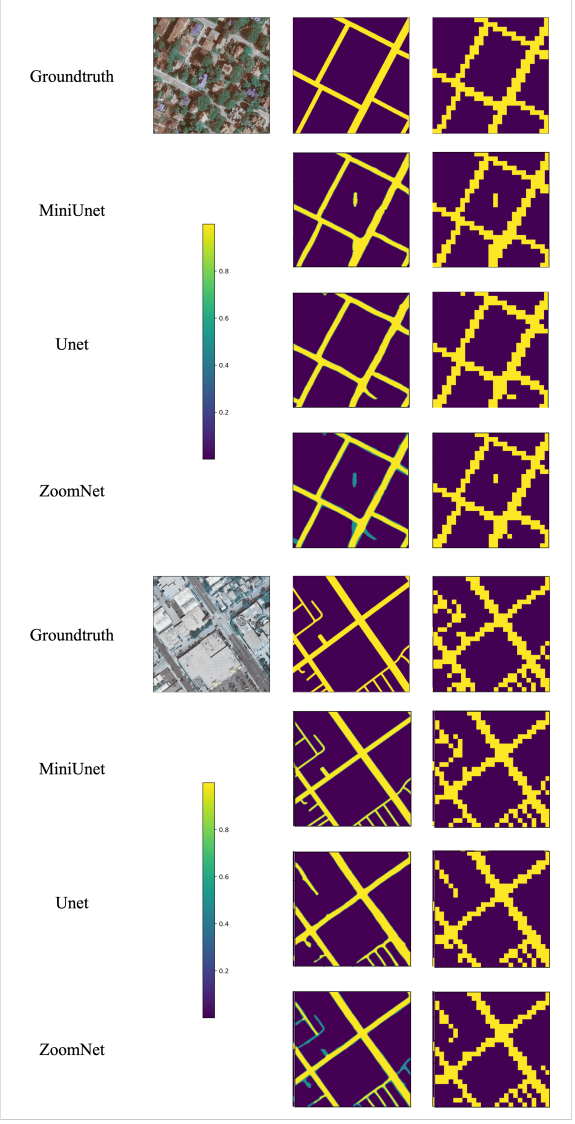

Figure 4: Visualization of Segmentation results from different models of the ZoomNet approach. On top the results for the image 'satimage_36' are displayed, while on the bottom the results for the image 'satimage_6'. The top section displays the satellite image alongside its corresponding ground truth and the 16x16 pixels patched-ground as in the task description. For each below row, the segmentation outcomes of the MiniUNet, UNet, and the combined result ZoomNet are displayed

determine what should be classified as a road. Furthermore, the ground truth data sourced from Google Maps may not perfectly reflect the road layout, leading to potential misclassifications of streets or non-street areas. Therefore, achieving a flawless image segmentation model may be considered impractical. In our research, we explored various approaches aiming to leverage the strengths of different models and combine their outputs to achieve improved results.

The results in Table-I show that using a larger dataset improves the performance of all the models significantly, suggesting a requirement for more data than *ETH-Data*.

Analyzing the performances of the Ensemble bootstrapping model we do not find a desired outcome. The UNet++ model, when utilized individually, outperforms the Ensemble Bootstrapping model. However, it is worth noting that the models trained within the Ensemble Bootstrapping does perform better than when trained individually as a baseline model. Throughout training, the individual models exhibits an enhancement in their predictions, while the ensemble does not demonstrate significant improvement. We speculate that this lack of improvement in the ensemble may be attributed to the single models converging on similar solutions, thereby limiting the overall predictive gain achieved through combining their outputs. It would be worthwhile to explore whether conducting the same approach without the ensemble, employing a single model, would yield better results, for example focusing only on the UNet.

On the other hand, the ZoomNet model demonstrates promising outcomes when compared to the baselines. As depicted in Table-I, the combined predictions of ZoomNet surpasses those of UNet and MiniUNet when used individually. We hypothesize that the UNet model might excel at capturing the broader context of the image, while the MiniUNet could focus more effectively on intricate details. Moreover, is worth noticing that the MiniUNet can be trained on a larger dataset due to the split of each image in quarter. To further enhance the results, it could be worthwhile to investigate the possibility of dividing the model further and incorporating these additional predictions into the ensemble.

Despite the Ensemble of baseline models producing the overall best predictions, our two new approaches provide valuable insights that warrant further investigations.

References

[1] "ETHZ CIL Road Segmentation 2023," https://www.kaggle.com/competitions/ethz-cil-road-segmentation-2023, [Accessed 31-07-2023].

[2] T. Peli and D. Malah, "A study of edge detection algorithms," *Computer Graphics and Image Processing*, vol. 20, no. 1, pp. 1–21, 1982. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0146664X82900703

[3] H. Ghandorh, W. Boulila, S. Masood, A. Koubaa, F. Ahmed, and J. Ahmad, "Semantic segmentation and edge detection approach to road detection in very high resolution satellite images," *Remote Sensing*, vol. 14, no. 3, 2022. [Online]. Available: https://www.mdpi.com/2072-4292/14/3/613

[4] A. Bimanjaya, H. H. Handayani, and R. F. Rachmadi, "Extraction of road network in urban area from orthophoto using deep learning and douglas-peucker post-processing algorithm," *IOP Conference Series: Earth and Environmental Science*, vol. 1127, 2023.

[5] D. Griffiths and J. Boehm, "Improving public data for building segmentation from convolutional neural networks (cnns) for fused airborne lidar and image data using

active contours," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 154, pp. 70–83, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0924271619301352

[6] Google, "Google maps platform," https://developers.google.com/maps, accessed: [22.06.2023].

[7] 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*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham: Springer International Publishing, 2015, pp. 234–241.

[8] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, D. Stoyanov, Z. Taylor, G. Carneiro, T. Syeda-Mahmood, A. Martel, L. Maier-Hein, J. M. R. Tavares, A. Bradley, J. P. Papa, V. Belagiannis, J. C. Nascimento, Z. Lu, S. Conjeti, M. Moradi, H. Greenspan, and A. Madabhushi, Eds. Cham: Springer International Publishing, 2018, pp. 3–11.

[9] A. Chaurasia and E. Culurciello, "Linknet: Exploiting encoder representations for efficient semantic segmentation," in *2017 IEEE Visual Communications and Image Processing (VCIP)*, 2017, pp. 1–4.

[10] H. Li, P. Xiong, J. An, and L. Wang, "Pyramid attention network for semantic segmentation," *CoRR*, vol. abs/1805.10180, 2018. [Online]. Available: http://arxiv.org/abs/1805.10180

[11] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," *CoRR*, vol. abs/1802.02611, 2018. [Online]. Available: http://arxiv.org/abs/1802.02611

[12] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," in *Advances in Neural Information Processing Systems*, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021. [Online]. Available: https://openreview.net/forum?id=OG18MI5TRL

[13] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Exploration of different approaches for Satellite Imagery Road Segmentation

Authored by (in block letters):
For papers written by groups the names of all authors are required.

Name(s):	First name(s):
Falardi	Lucas
Krähenmann	Max
Peter	Enea
Strozzi	Virgilio

With my signature I confirm that
- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	Signature(s)
Zürich, 31.07.2023	Lucas Falardi V. Strozzi

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.