Augmentation Techniques for Large Image Segmentation with Partially Available Labels

Valentin Bieri, Onat Vuran, Selin Barash, Ahmet Alper Özüdogru
Department of Computer Science, ETH Zurich, Switzerland

***Abstract*—Creating road segmentation maps from aerial images remains to be an important task in today's world. Satellite imagery serves as a crucial resource for generating current road maps. When undertaken manually, the data labelling process not only demands substantial manpower but also leaves room for human errors. Implementing automated techniques for road segmentation can mitigate these challenges by eliminating the necessity for manual labelling. In this study, we work on the CIL-Competition-2023 dataset and offer a comparative benchmark of existing methods, combining them with strategies of learning from partially labelled data and applying test time augmentations to improve the performance.**

I. Introduction

Semantic understanding of satellite images - and in particular the problem of road segmentation - consistently attracts significant research interest. The availability of accurate road detection algorithms enables more inexpensive maintenance of continuously evolving high-quality maps, which is in the interest of many industrial, public, or even military applications.

Since the rise of deep neural networks for image segmentation [1] and in particular convolutional neural networks (CNNs) [2], a new state of the art for image segmentation has been established. For generic image segmentation, various open-source libraries implement different approaches and neural network architectures for the task. One popular example is the segmentation-models-pytorch (SMP) library [3], which provides a collection of encoder-decoder models [4]. In these models, the first part - the encoder - generates image features and passes them to the second part - the decoder - where they are used to create segmentation maps. Typically, encoder as well as decoder are divided into stages, whereas each stage of the encoder passes features to its counterpart in the decoder.

The task of road segmentation from satellite image patches is no exception to this development. Popular segmentation benchmarks [5][6][7] demonstrate the strong performance of deep CNNs in the field. One crucial limitation of both benchmark and proposed methods in the field is that they usually do not consider the global consistency of the predictions. When dividing large satellite images into small patches, relevant context is lost across neighboring patches. To investigate the benefits of context-aware models, we merge the given competition image patches into two large satellite images (see 1). We observe that the patches were seemingly randomly split into train and test set, which replaces the original per-pixel binary segmentation task for satellite image patches with an impainting task, where we have to predict the missing segmentation maps for areas within a very large image.

The goal of this work is to empirically demonstrate the numerous advantages of a more global, context-aware perspective on road segmentation from aerial images. The underlying generic task of large image segmentation with partially available labels is - despite its many valuable applications - not an equally deeply studied field of research. A very simplistic approach to deal with the image size is to train a classifier for small, computationally feasible patch sizes and then apply the classifier to each patch independently. One can extend this approach to arbitrary complexity by extracting patches with a sliding window scheme at multiple scales and rotations. In this work, we demonstrate that even very simple patch extraction schemes provide significant improvements over the common non-overlapping patch grid.

In our case, a (known) part of the image includes expert annotations. Whereas this setting seems somewhat artificial in our context, we see a number of highly relevant related problems. An example is human-assisted semi-automatic image segmentation, which is not only used in the creation of large segmentation datasets but also in applications where human judgment is necessary in critical parts of the image (e.g. medical field). To make use of the additional information in these partial labels, we propose a training scheme particular to this task. That is, we train models on patches with only partially available labels, aiming to improve the accuracy of the non-labeled image regions. To that end, the loss of the unknown area is simply masked.

In a series of experiments, we test the effectiveness of this simple training scheme and consistently observe benefits over the standard version. We compare our models in a benchmark with two contemporary approaches. The first one is a vanilla U-Net model that is only trained with patches from the CIL-Competition-2023. This baseline was provided by the competition already and reproduced by us in a first step. Its second, improved version differs in using more data, which enables transfer learning to some extent. Finally, we present an improved architecture based on the SMP library and test it using the proposed training and inference

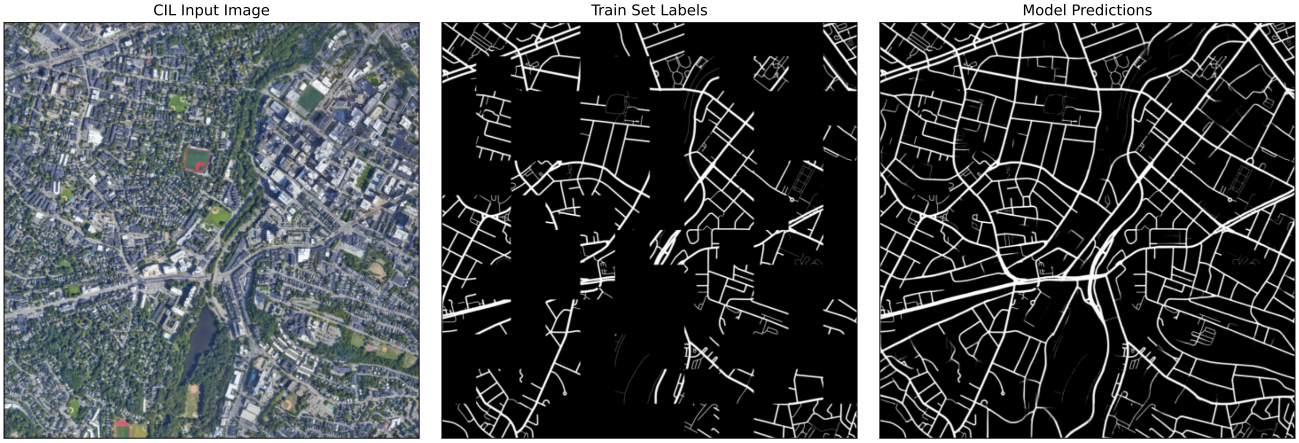


Figure 1: One of the two assembled input images, the given train set labels, and the corresponding model predictions from left to right.

schemes, observing small but consistent improvements over the standard training techniques with the same models.

II. Method

We had two main goals and therefore developed two different methodologies for the task. Our first goal was to create a strong second baseline using popular proven methods in learning segmentation maps for satellite image patches. The second benchmark was designed to create a competitive candidate for our solution. We compare this second benchmark with the first baseline which is already provided as an example in the competition.

As an improvement to the second benchmark we also considered using a discriminator loss to further train a pretrained model because we observed that our model predictions were easily differentiated with the ground truth prediction maps. The discriminator loss was summed up with the dice loss with a weight of 0.01.

The second goal was to solve the reformulated large satellite image segmentation problem with learning using partial labels and applying test time augmentations at inference time. Learning with partial labels also facilitates in inference with test time augmentations. Finally, we used an ensemble of our best performing models.

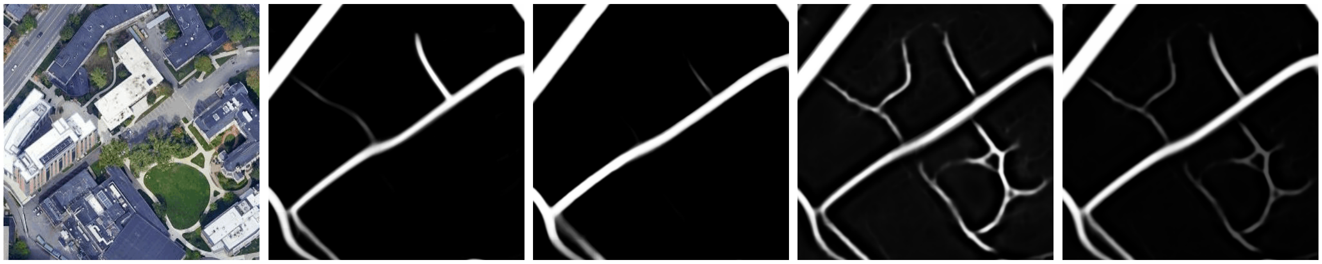
A. *Second Benchmark*

A way to improve the vanilla U-Net was to improve its architecture. We extended the vanilla version by using a performance proven architecture as a backbone in its encoder. This also allowed us to use pretrained weights learned in ImageNet Classification problem. To use these backbones as encoder we extracted features from several levels of the model and used them in the skip connections in the U-Net architecture. The backbones were combined with a convolutional decoder. We also tested with slightly different model types such as DeepLabV3+(ref), Unet++. We used smp library implementations for all models[add ref]. The vanilla U-Net benchmark was trained with Binary Cross Entropy loss, we observed that using a dice loss [8] was more performant for our task experimentally. The dice loss (1) between the prediction and the groundtruth is computed by dividing the intersecting area by the total area, and has been used widely for comparing images. In the following equation p_i are the predicted probabilities, the y_i values are the ground truth labels for each pixel, and ϵ is used for numerical stability.

$$\mathcal{L}_{dice}(p, y) = 1 - \frac{\sum_{n=1}^{N} p_n y_n + \epsilon}{\sum_{n=1}^{n} p_n + y_n + \epsilon} \tag{1}$$

CIL-Competition-2023 data consisted of 400x400 RGB satellite image patches with their pixel-wise ground truth binary road segmentation maps. 144 of these pairs were training images and the remaining 144 were test images. We extended this data by creating datasets by scraping. We scraped data from Bing and Google Maps, and combined them with already publicly available Roadtracer and EPFL datasets.

Several augmentations were applied in training time using the albumentations library [9]. The augmentations consisted of random flips, rotations, shadow additions, color adjustments, and noise with different probabilities for each augmentation. By introducing these diverse transformations, the model is exposed to a broader range of variations in the data, effectively simulating different real-world scenarios. This process encourages the model to learn more robust and generalized features from the training data, making it better equipped to handle variations, noise, and distortions that it may encounter during inference on unseen data. The augmentation significantly improves the model's ability to



(a) Satellite Image Patch (b) Unet Prediction (c) Unet++ Prediction (d) DeepLab Prediction (e) Ensemble Prediction

Figure 2: Model Segmentation Patch Predictions Examples

generalize well to different situations, which can lead to better overall performance and enhanced predictive capabilities. We see that these also help the model to gain robustness in its predictions, which is useful for test time augmentations.

B. Learning with Partial Labels

Learning with partial labels is a training recipe, in which we dynamically sample 400x400 image patches from the large satellite image and its corresponding ground truth segmentation map with some unknown label areas due to their corresponding patches belong to the test set. We train the model with random samples taken from this large satellite image. The sample is passed through the model, and the resulting segmentation map prediction is further masked with a loss mask to only make the model focus only on the loss calculated on some part of the image. The prediction is always masked with zeros always on the areas with unknown labels and sometimes on the areas with known label to make the model stochastically on a specific randomly chosen area.

This method allows models to see the connections between neighbor patches during training and use this information to come up with more compatible predictions. This training recipe can be applied to any model with no modification required on the model side.

C. Test Time Augmentations

In this project, we took a novel approach for performing test-time augmentation (TTA) to enhance the predictive performance of our road segmentation model.

Initially, we assembled all the provided training and test images to form two large satellite images/ each consisting of a 12x12 grid of 400x400 RGB images. This allowed us to process and analyse these images in a more cohesive and holistic manner, as opposed to treating each image patch individually. In particular, the predicted labels at the sides/corners of an image are usually not predicted accurately, since without seeing the surroundings of a pixel, it is hard to say if it is part of the road or not. To overcome this, we added the following strategies. We took patches from the large images, shifting a window by approximately 1/5 of the image size. The overlapping patches provided more confident labels making the regions central part of some patch. We also only took the labels from a parameterized central square and ignore the remaining predicted labels outside this square. Moreover, we applied various transformations to the large images, including rotation, scaling, and flipping. By using these transformations we generated a diverse set of augmented samples for our model to make predictions on. This significantly improved the model's ability to capture a wider range of variances that might be present in the test data.

After predicting the labels for each patch, we assembled them back together to form a large segmentation map prediction for the entire large satellite image. We used an ensemble approach to combine these predictions, averaging the predictions from different transformation methods to create a single, more accurate prediction for each large image.

Finally, to generate the results for the individual test images, we mapped the predicted labels from the large 12x12 image back to their original locations in the smaller images.

D. Data

1) Bing: We scraped approximately 9.2k additional satellite image patches and their corresponding road maps through the Bing API. We cropped the Microsoft logo and resized images to 400x400 to comply with the competition. Since competition data was focused on urban areas, we only scraped cities worldwide with populations exceeding 800k. We also filtered out, regions with almost no road pixels.

2) RoadTracer: In their road segmentation model evaluation, Bastani et al. [10] compiled an extensive dataset consisting of high-resolution satellite images and ground truth road network graphs encompassing the urban core of forty cities in six different countries. Named the "RoadTracer" dataset, each city's region covered an approximate area of 24 square kilometers around the city center. The dataset featured satellite imagery sourced from GoogleMaps and OpenStreetMap at a resolution of 60 cm/pixel.

3) EPFL: We utilized the dataset from a road segmentation challenge hosted by EPFL, which is accessible at https://github.com/arthurbabey/road66/tree/master. The dataset comprises 340 labeled images.

4) Google Maps: A total of 22.8k additional 600x600 images were acquired using the Google Maps API. These images were collected from the 20 most populated cities in the USA, including prominent urban centers such as New York, Houston, and Philadelphia. To maintain consistency with the given images in the competition, we made an effort to keep the same zoom-level and carefully selected cities with similar urban structures. This approach ensures that the augmented dataset complements the original dataset, creating a diverse and relevant training set for the model to learn from.

III. Training & Experiments

During training, we had two stages. The first stage was pretraining in which we use all datasets, and a fine tuning stage in which we split the CIL-Competition-2023 into 5 folds and finetune the best model obtained in the pretraining stage with each fold. Even though training recipe is slightly different for each model, we generally used 200 steps of learning rate warm up from 1% the learning rate up to the original value. Later used cosine annealing scheduler to decrease it. Learning rate was selected to be 0.0007 for the pretraining phase and 0.0001 for the finetuning phases. We used ImageNet classification task trained weights in backbones. Best model was decided by comparing the IoU(Intersection over Union) scores on the validation set of different epochs. The discriminator was trained during the pretraining phase along with the model with a 0000.5 learning rate. During finetuning, last checkpoint of the discriminator in pretraining phase was used for all folds. We used a simple PatchGAN [11] discriminator architecture.

We made experiments with different selection of backbones such as efficient-net [12], regnety [13], with different model architectures such as DeepLabV3+[14], U-Net[15], U-Net++[16]. We also added each improvement to the training recipe sequentially to demonstrate each of the steps effectiveness separately.

To generate predictions out of different folds we averaged their segmentation prediction maps. This approach is also used in the ensemble where we averaged every models' folds with equal weights. We applied thresholding with 0.5 to each folds' outputs to convert them to binary but did not do so for the ensembled predictions.

IV. Results & Conclusion

Table I depicts or method comparisons. The patch F1 score is based on whether the average label on 16x16 grid in the segmentation map prediction aligns with the average label in the ground truth. The gray background models corresponds to our benchmarks, with higher scoring being the second benchmark. By looking at the results, we can conclude that using improved architecture with a state-of-the art backbone brings a huge leap to the first baseline performance. However, we did not see a big difference between different model types, they were quite close in performance.

Architecture/ Backbone	Scraped Data	ImageNet Weights	Dice Loss	Discriminator Loss	Partial Labels	Test Time Augmentations	PatchF1 Score
Unet/ No Backbone	X	X	X	X	X	X	0.86380
Unet/ timm-regnety-080	Y	X	X	X	X	X	0.91466
Unet/ timm-regnety-080	Y	Y	X	X	X	X	0.9202
Unet/ timm-regnety-080	Y	Y	Y	X	X	X	0.93488
DeepLabV3+/ timm-regnety-320	Y	Y	Y	X	X	X	0.93677
DeepLabV3+/ timm-regnety-320	Y	Y	Y	Y	X	X	0.93742
DeepLabV3+/ timm-regnety-320	Y	Y	Y	Y	Y	Y	**0.94671**
Unet/ timm-regnety-080	Y	Y	Y	X	Y	Y	**0.94637**
UnetPlusPlus/ efficientnet-b7	Y	Y	Y	X	X	Y	**0.94611**
Ensemble: Unet + UnetPlusPlus + DeepLabV3+	Y Y Y	Y Y Y	Y Y Y	XXY	Y X Y	Y Y	**0.94811**

Table I: Results
Light gray: First Benchmark
Dark gray: Second Benchmark

During our experiments we observe that using an ImageNet weights initialization for the backbone speeds up the training greatly. Maybe with longer training, the same score could be reached but it wouldn't be an efficient use of resources so we didn't test it. Considering the effect of loss between the two benchmarks and better performing models, we observed that binary cross entropy loss always provided slower training which created inefficiency in training. We found that dice loss provides a more generalizable model with faster training.

We noticed that augmenting the DeepLab model with a discriminator bolstered its capacity to accurately predict narrow roads, as depicted in Figure 2d, by boosting the model's assurance in these zones. However, this also resulted in an increase in false positives. While the Unet and Unet++ outputs were fairly similar, they identified differing minor roads within varying images (Figure 2b, 2c), and had a sharper prediction of the main roads than the DeepLab model, affirming that their inclusion still offers value. By combining these three models, we achieved an ensemble that could reliably predict minor roads while maintaining commendable performance on main roads. These visual analyses align with the observed test scores in Table I.

Our results indicate that the biggest improvement comes from the test time augmentations along with learning with partial labels. When we applied partial label learning task to an already pretrained model and then generate the predictions we observed that model gains a lot of robustness and better prediction capabilities.

References

[1] T. Panboonyuen, K. Jitkajornwanich, S. Lawawirojwong, P. Srestasathiern, and P. Vateekul, "Road segmentation of remotely-sensed images using deep convolutional neural networks with landscape metrics and conditional random fields," *Remote Sensing*, vol. 9, no. 7, 2017. [Online]. Available: https://www.mdpi.com/2072-4292/9/7/680

[2] P. Kaiser, J. Wegner, A. Lucchi, M. Jaggi, T. Hofmann, and K. Schindler, “Learning aerial image segmentation from online maps,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. PP, pp. 1–15, 07 2017.

[3] [Online]. Available: https://github.com/qubvel/segmentation_models.pytorch

[4] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” 2014.

[5] W. G. C. Bandara, J. M. J. Valanarasu, and V. M. Patel, “Spin road mapper: Extracting roads from aerial images via spatial and interaction space graph reasoning for autonomous driving,” 2021.

[6] [Online]. Available: https://www.isprs.org/education/benchmarks/UrbanSemLab/2d-sem-label-potsdam.aspx

[7] [Online]. Available: https://www.isprs.org/education/benchmarks/UrbanSemLab/2d-sem-label-vaihingen.aspx

[8] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. J. Cardoso, “Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations,” in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Springer International Publishing, 2017, pp. 240–248. [Online]. Available: https://doi.org/10.1007%2F978-3-319-67558-9_28

[9] [Online]. Available: https://albumentations.ai/

[10] B. F., “Roadtracer: Automatic extraction of road networks from aerial images,” 2018.

[11] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” 2018.

[12] M. Tan and Q. V. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” 2020.

[13] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, and P. Dollár, “Designing network design spaces,” 2020.

[14] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” 2018.

[15] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” 2015.

[16] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, “Unet++: A nested u-net architecture for medical image segmentation,” 2018.