

Road Segmentation in Aerial Images

Stefan Jokić, Yan Walesch, Thomas Zhou

Group: SupportVictoryMachines

Department of Computer Science, ETH Zurich, Switzerland

Abstract—Road segmentation in images is an important computer vision task commonly tackled with encoder-decoder convolutional networks. Weaknesses in high-level semantic representations and their spatial reconstruction have motivated state-of-the-art architectures such as LAnet. We modify the LAnet architecture to better suit our problem, and present a novel post-processing layer for road segmentation. We apply pre-training, data augmentation, and test-time augmentation for more robust predictions. Experiments show that our approach improves performance over our baselines and achieves a public mean F1 score of 0.92147.

I. INTRODUCTION

As a fundamental topic in computer vision, image segmentation has been a widely researched topic in the last decades. The special case of road segmentation consists of the classification between road and non-road objects in aerial images, and is useful to many different applications such as city planning, car navigation, autonomous driving, etc. Recently, the development of convolutional neural networks (CNNs) [1], [2] and their successful application to fully convolutional neural networks (FCNs) [3] have transformed the field of semantic segmentation.

In 2015, *Ronneberger et al.* [4] proposed the U-Net: a novel FCN architecture using an encoder-decoder paradigm. In encoder-decoder designs, a contracting path comprised of stacked convolutions and pooling (encoder) captures the context of an image into a high-level semantic representation. As fine spatial information is often lost in the encoded representation, an expansive path comprised of convolutions and upsampling (decoder) is applied to recover spatial details. Commonly, the decoder also retrieves low-level features from the encoder using ‘skip connections’ to assist with the process [4], [5], [6].

In the context of road segmentation, however, a single aerial image often comprises of many local sub-contexts (highway, suburb, parking lot, etc.) which can become blurred in the high-level semantic representation. Although the fusion with low-level features via skip connections can help alleviate the issue, aggregations are often shallow [7] and usually involve concatenating [4] or summing [6] low-level features with the high-level features, providing limited semantic enhancement. Consequently, *Ding et al.* [8] propose the LAnet, which enhances an FCN with a *Patch-Attention Module (PAM)* and *Attention Embedding Module*

(*AEM*) to address the weaknesses in high-level semantic representations and representation fusion, respectively.

In this work we aim to build on the success of the LAnet, and further improve the performance of this approach in the context of road segmentation. Hence, we propose a modified architecture to the existing approach which is more suitable to our task. Furthermore, we propose a novel method for the post-processing of our predictions which exploits the spatial structure of road pixels in aerial images.

The specific problem we aim to tackle in this work is as follows. Given an aerial image of roads, we want to assign a label in $[0, 1]$ to each pixel in the image, which indicates the probability that the given pixel contains a road. Ultimately, however, we evaluate our approach on a per-patch basis, i.e. we average the predicted labels for each 16×16 patch and assign the label 1 (road) to a patch if the average probability is larger than 0.25 and the label 0 (background) otherwise.

Our report is structured as follows. In Section II, we explain the baselines we compare our model with, the architecture of the LAnet, and our proposed approach. Furthermore, we clarify the dataset, exact training and inference methodology, and explain the details of our novel post-processing method. In the following Section III, we compare our results to the previously described baselines, perform an ablation study, and qualitatively analyze the performance of our approach by visualizing predictions, before discussing the results in Section IV. Finally, we briefly conclude our work in Section V.

II. MODELS AND METHODS

A. Baselines

We establish three different baselines for aerial image segmentation with which we compare our proposed approach. This allows to highlight the improvement of our approach compared to other commonly employed methods.

1) *CNN Patch Classifier*: Our first baseline is a shallow CNN which classifies 16×16 aerial road image patches as road (1) or background (0) independent of the image context. It is composed of three convolutional layers, each followed by a max-pooling layer, followed by a dense network to a single output unit using the logistic activation function to output a value in $[0, 1]$. It includes batch normalization and dropout layers to ensure better generalization of the network and more stable training.

2) *U-Net*: As mentioned in Section I, the U-Net [4] is a symmetric encoder-decoder FCN which uses skip connections, implemented as concatenations, to fuse low-level feature maps from the encoder with corresponding sparse feature maps from the decoder. In our implementation, we employ bilinear upsampling layers rather than tranpose convolutions in the decoder, as there is evidence in the literature that the latter may produce high frequency checkerboard-like artifacts [9].

3) *Pix2Pix*: Another powerful approach for image segmentation is by means of conditional generative adversarial networks (cGAN). cGANs differ from traditional generative adversarial networks (GAN) in that they learn a conditional generative model by conditioning both the generator and discriminator on some auxiliary information. In the context of image-to-image translation, for instance, one can condition on the images of the training data. Isola et al. [10] propose a cGAN for image-to-image translation, dubbed *Pix2Pix*, which has achieved promising results even with regard to image segmentation. It comprises of a generator which adopts a U-Net-based architecture and a discriminator called *PatchGAN* which is a small convolutional network. Rather than directly classifying each pixel of an image as real (originating from the true data distribution) or fake (synthesized using the generator), the PatchGAN discriminator classifies entire patches of size $N \times N$ instead. In our case the generator, conditioned on the input aerial images, synthesizes road segmentation maps. Aside from the standard cGAN minimax objective

$$G^* = \arg \min_G \max_D \left(\mathbb{E}_{x,y} [\log D(x,y)] + \mathbb{E}_{x,z} [\log (1 - D(x, G(x,z)))] \right) \quad (1)$$

that the Pix2Pix cGAN aims to optimize, in our case the generator also attempts to minimize the L1 reconstruction loss \mathcal{L}_{L1} with respect to the ground truth image [10], i.e., segmentation map. We adjust the original implementation by removing some layers from the encoder and decoder of the U-Net generator, as we observe better performance on the given data.

B. Neural Network Architecture

As discussed in Section I, *Ding et al.* [8] enhance an FCN with PAMs and an AEM in their LANet architecture. We properly introduce the PAM and AEM, before describing our proposed modified architecture.

1) *Patch Attention Module*: The PAM enhances feature maps with local patch-level semantic information to address the inadequacy of regular encoded representations in capturing local sub-contexts. For a single $h_p \times w_p$ patch p , let z_c denote its descriptor for the c th channel:

$$z_c = \frac{1}{h_p \cdot w_p} \sum_{i=1}^{h_p} \sum_{j=1}^{w_p} x_{c,i,j}$$

where $x_{c,i,j}$ is the pixel value for the c th channel at patch coordinates (i, j) . From the feature map $\mathbf{X} \in \mathbb{R}^{C \times H \times W}$, we can generate a map of patch descriptors $\mathbf{Z} \in \mathbb{R}^{C \times H' \times W'}$ where $H' = \frac{H}{h_p}$ and $W' = \frac{W}{w_p}$, and h_p, w_p are set according to the output stride of the input feature map. The PAM uses bottleneck convolutions [11] to generate an attention map $\mathbf{A} \in \mathbb{R}^{C \times H \times W}$:

$$\mathbf{A} = F_U \{ \sigma [H_i \delta (H_r \mathbf{Z})] \}$$

where σ and δ represent sigmoid and ReLU functions, respectively; H_r represents a 1×1 dimension-reducing convolution to $\frac{C}{r}$ channels for some reduction ratio r ; H_i represents a 1×1 dimension-restoring convolution back to C channels; F_U is the bilinear upsampling operation to the output dimensions. Completely, the PAM uses a residual mapping [12] to enhance the feature map with

$$\mathbf{X} = \mathbf{X} + (\mathbf{X} \odot \mathbf{A})$$

where \odot represents element-wise multiplication, i.e., the Hadamard product.

2) *Attention Embedding Module*: The AEM attempts to embed local attention from high-level semantic features into the low-level spatial feature map more effectively than simple concatenation or summing. Let $\mathbf{X}_h \in \mathbb{R}^{C_h \times H_h \times W_h}$ and $\mathbf{X}_l \in \mathbb{R}^{C_l \times H_l \times W_l}$ represent the high level and low level feature maps, respectively. We generate a patch descriptor map $\mathbf{Z}_h \in \mathbb{R}^{C_h \times H' \times W'}$ from \mathbf{X}_h , then similarly generate an attention map $\mathbf{A}_l \in \mathbb{R}^{C_l \times H_l \times W_l}$ for low level features:

$$\mathbf{A}_l = F_U \{ \sigma [H_l \delta (H_r \mathbf{Z}_h)] \}$$

where H_l is a 1×1 dimension-changing convolution to C_l channels. We embed the lower level features with

$$\mathbf{X}_l = \mathbf{X}_l + (\mathbf{X}_l \odot \mathbf{A}_l)$$

3) *Complete Architecture*: *Ding et al.* [8] incorporate the PAM and AEM into an FCN with a ResNet [12] encoder backbone. Instead, our proposed architecture (cf. Figure 1) replaces the ResNet with our own shallower encoder using fewer convolutional and pooling layers. Notably, the encoder has an output stride of 16: each pixel in the high-level (output) feature map corresponds to a 16×16 patch in the input image, i.e., the unit of prediction when evaluating the model score. As in the LANet, the model extracts \mathbf{X}_l and \mathbf{X}_h from the first and output layer of the encoder, enhances both feature maps using their respective PAMs, then embeds \mathbf{X}_h into \mathbf{X}_l using an AEM. The final prediction mask $\hat{\mathbf{Y}} \in \mathbb{R}^{1 \times H \times W}$ is generated as

$$\hat{\mathbf{Y}} = \sigma [F_U (H_1 \mathbf{X}_h) + F_U (H_1 \mathbf{X}_l)]$$

i.e., a *predict and fuse* step, where H_1 is a 1×1 dimension-reducing convolution to a single channel for prediction.

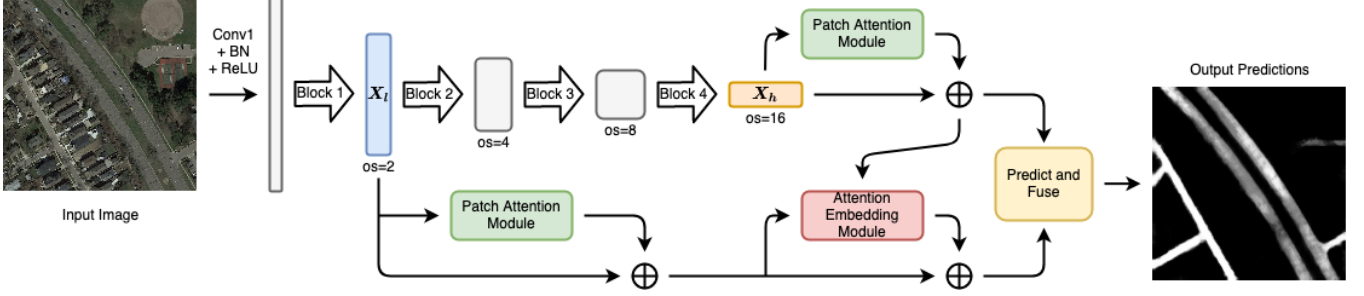


Figure 1: Our proposed FCN architecture with PAMs and an AEM. Blocks 1, 2, 3, and 4 each comprise of a 2×2 max pooling layer followed by two sequential (3×3 Convolution + Batch normalization + ReLU) units. (os: output stride)

C. Dataset

We train our model on a set of 100 400×400 RGB aerial images from Google Maps with corresponding binary masks $\in \{0, 1\}$ generated from probabilistic masks $\in [0, 1]$ using a 0.5 threshold. We test our model on a set of 94 608×608 RGB aerial images. We normalize the training images to zero mean unit variance, and normalize the test images using the means and standard deviations of the training set to ensure value equivalency during inference.

D. Training

We train and validate our model using a random 80-20 training-validation data split. To avoid overfitting to the training set, we employ pre-training and data augmentation.

1) *Pre-Training*: We pre-train our model on the DeepGlobe [13] dataset, which contains 6226 1024×1024 RGB aerial images with corresponding binary masks. To maintain consistency with the training data, we generate 2 random 400×400 crops from each image for a total of 12452 pre-training samples, similarly normalized by the means and standard deviations of the training set.

2) *Data Augmentation*: During training, we perform random data augmentation using image shifting by a factor $a_s \in [-0.0625, 0.0625]$, scaling by a factor $a_z \in [1, 1.1]$, rotation by a factor $a_r \in [-45, 45]$, brightness augmentation by a factor $a_b \in [1, 1.2]$, contrast augmentation by a factor $a_c \in [1, 1.2]$, horizontal flip, and vertical flip. Each augmentation is applied with a probability $p = 0.5$.

3) *Implementation*: For our best model, we use a linear combination of Binary Cross Entropy (BCE) and Dice coefficient [14] loss, and use Adam [15] as our optimizer. We use a learning rate $\eta = 5 \times 10^{-4}$, batch size of 4, and \mathcal{L}_2 -regularization term of 10^{-8} .

E. Test-time augmentation

At inference, we apply test-time augmentation (TTA) to our model as follows. First, we generate a set \mathcal{T} of the following transformations: rotation of 90° , 180° , 270° , horizontal flip, and vertical flip. Then, we generate all possible combinations of transformations in \mathcal{T} , i.e., the power set

$\mathcal{P}(\mathcal{T})$. Since $|\mathcal{T}| = 5$, this results in a total of $2^5 = 32$ transformations. We generate a prediction for each of the 32 transformed images and then restore the prediction to match the original image by applying the reverse of the transformations. Finally, we average the probability of each prediction to generate final output predictions for the given test image.

F. Post-processing

Inspired by [16], we propose a novel post-processing layer which exploits the tendency for road segments to be continuous and of consistent width. Firstly, we apply a dense Conditional Random Field (CRF) model [17] to our prediction probabilities; For image road segmentation, CRF models have been shown to produce clearer output masks by detecting dependencies between the input and output domains [18]. Next, we apply the algorithm presented in [19] on the output of the CRF model to produce a thin road network 'skeleton': we denote the set of pixel coordinates belonging to the skeleton as \mathcal{S} . We use \mathcal{S} to generate a prediction probability adjustment map $\mathbf{M} \in [-0.5, 0.5]^{H \times W}$:

$$m_{i,j} = \max \left\{ -0.5, \quad 0.5 - 0.1 * \min_{s \in \mathcal{S}} L_1[(i,j), s] \right\}$$

The adjustment value $m_{i,j}$ linearly decreases based on the L_1 distance between pixel coordinates (i,j) and its nearest skeleton pixel coordinates. Hence, \mathbf{M} encourages consistent-width road predictions around the identified skeleton, and diminishes noisy predictions further away from the skeleton. We add \mathbf{M} to the original prediction probabilities, i.e., $\hat{\mathbf{Y}} = \hat{\mathbf{Y}} + \mathbf{M}$, before binary thresholding at 0.9 to produce predictions in $\{0, 1\}$. Finally, the layer applies morphological opening by erosion and dilation to further de-noise the final mask.

Figure 2 demonstrates the key effects of the layer. In the top test image, it identifies a skeletal structure and generates a map which enables some parts of the roads between the parking lots to exceed the binary threshold. In the bottom test image, the layer closes small gaps between road segments and enforces a more consistent width.

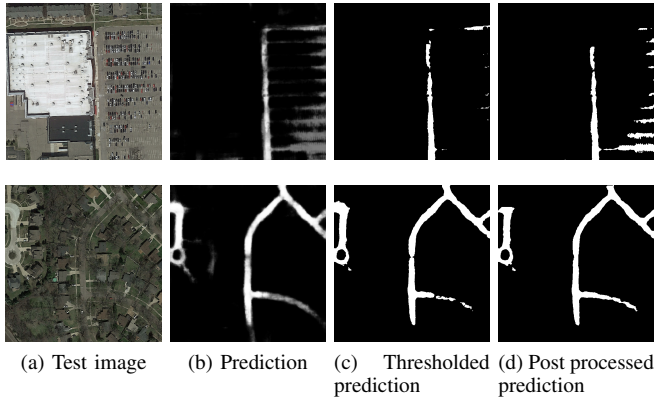


Figure 2: Comparison between prediction probabilities (b), binary thresholded predictions without post-processing (c), and predictions after our post-processing layer (d).

III. RESULTS

Table I depicts the performance of the three baselines and our approach, as well as an ablation study. The (mean) F1 score corresponds to the public score on Kaggle computed using a fraction of the test data set. Visualisations of the predictions produced by the various models are given in Figure 3 in the appendix. Note that methods which do not use our post-processing layer instead binary threshold the prediction probabilities at 0.5 to generate the final binary prediction. Evidently, the CNN patch-based classifier has by far the worst performance, while the U-Net and Pix2pix cGAN constitute competitive baselines, especially with data augmentation and TTA. Our FCN with the PAM and AEM, coupled with all of our proposed methods, achieves the highest score of 0.92147.

Method	F1 Score
Baseline 1: CNN Patch Classifier	0.80267
Baseline 2: U-Net	0.88195
Baseline 3: Pix2Pix	0.86251
U-Net + AUG + TTA	0.90130
Pix2Pix + AUG + TTA	0.89518
Ours (FCN + PAM + AEM)	0.88200
Ours + Pre-training	0.88966
Ours + Pre-training + AUG	0.91528
Ours + Pre-training + AUG + TTA	0.92043
Ours + Pre-training + AUG + TTA + PP	0.92147

Table I: Comparison between different methods.

AUG: Training data augmentation, **TTA**: Test-time augmentation, **PP**: Post-processing

Table II compares the results between the vanilla LANet from [8] and our approach. We also investigate the impact that the PAM and AEM have on the performance of our model. Note that all the networks in Table II use pre-trained weights and employ data augmentation and test-

Method	F1 Score
LANet [8]: ResNet FCN + PAM + AEM	0.91086
Our FCN	0.91433
Our FCN + PAM	0.91492
Our FCN + AEM	0.91477
Our FCN + PAM + AEM	0.92043

Table II: Comparison between the LANet [8] and our FCN architecture & an ablation study of our FCN with regard to the PAM and AEM modules.

time augmentation. The results show that even without the PAM and AEM modules, our proposed FCN has similar performance to the LANet on the given dataset. Although the PAM and AEM individually do not substantially improve the accuracy, a larger improvement is observed when they are combined.

IV. DISCUSSION

It is clear from the results that simple approaches such as the CNN patch classifier are easily outperformed by more sophisticated methods (U-Net and Pix2Pix). This is because the CNN patch classifier is limited by the fact that it classifies patches without considering the context around them. The Pix2Pix, which incorporates a U-Net as its generator, does not achieve results as good as those of the U-Net alone. Since the Pix2Pix approach employs a cGAN, which is in general notoriously difficult to train and tune, it is possible that we may not have been able to "unlock" the full potential of Pix2Pix.

Furthermore, a possible explanation as to why our approach outperforms the ResNet-based LANet is that our shallower network has less parameters, making it not only easier to train, but also better suited when dealing with a small training data set [20]: in our case, we only have 100 training images. Even when data augmentation is applied, the training data still constitutes a relatively small set compared to many publicly available data sets that neural networks for image segmentation are typically trained on (cf. [13], [21], [22]). Deeper networks such as ResNet [12] are likely more prone to overfitting on our small data set, but with further addition of regularization methods, comparable results to our small FCN could be attainable.

V. CONCLUSION

In this work, we have developed an approach that leverages the main ideas put forward by Ding et al. [8]: We have designed our own small FCN architecture in conjunction with LANet's patch-attention and attention embedding modules. Furthermore, we bolstered the performance of our network by applying a novel post-processing method, data augmentation and TTA. This combination of various pre- and post-processing methods has lead to significant performance enhancements.

REFERENCES

- [1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds.
- [2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015.
- [3] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *CoRR*, vol. abs/1411.4038, 2014. [Online]. Available: <http://arxiv.org/abs/1411.4038>
- [4] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015.
- [5] G. Lin, A. Milan, C. Shen, and I. Reid, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1925–1934.
- [6] Z. Zhang, X. Zhang, C. Peng, X. Xue, and J. Sun, "Exfuse: Enhancing feature fusion for semantic segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 269–284.
- [7] F. Yu, D. Wang, E. Shelhamer, and T. Darrell, "Deep layer aggregation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2403–2412.
- [8] L. Ding, H. Tang, and L. Bruzzone, "Lanet: Local attention embedding to improve the semantic segmentation of remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 1, pp. 426–435, 2020.
- [9] A. Odena, V. Dumoulin, and C. Olah, "Deconvolution and checkerboard artifacts," *Distill*, 2016. [Online]. Available: <http://distill.pub/2016/deconv-checkerboard>
- [10] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," 2018.
- [11] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.
- [12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [13] I. Demir, K. Koperski, D. Lindenbaum, G. Pang, J. Huang, S. Basu, F. Hughes, D. Tuia, and R. Raskar, "Deepglobe 2018: A challenge to parse the earth through satellite images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 172–181.
- [14] 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, 2017, pp. 240–248.
- [15] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [16] W. Chen, "Road segmentation based on deep learning with post-processing probability layer," *IOP Conference Series: Materials Science and Engineering*, vol. 719, p. 012076, 01 2020.
- [17] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," *CoRR*, vol. abs/1210.5644, 2012. [Online]. Available: <http://arxiv.org/abs/1210.5644>
- [18] A. Dhawan, P. Bodani, and V. Garg, "Post processing of image segmentation using conditional random fields," in *2019 6th International Conference on Computing for Sustainable Global Development (INDIACom)*, 2019, pp. 729–734.
- [19] T. Y. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns," *Communications of the ACM*, vol. 27, no. 3, pp. 236–239, 1984.
- [20] L. Brigato and L. Iocchi, "A close look at deep learning with small data," 2020.
- [21] V. Mnih, "Machine learning for aerial image labeling," Ph.D. dissertation, University of Toronto, 2013.
- [22] A. V. Etten, D. Lindenbaum, and T. M. Bacastow, "Spacenet: A remote sensing dataset and challenge series," 2019.

APPENDIX

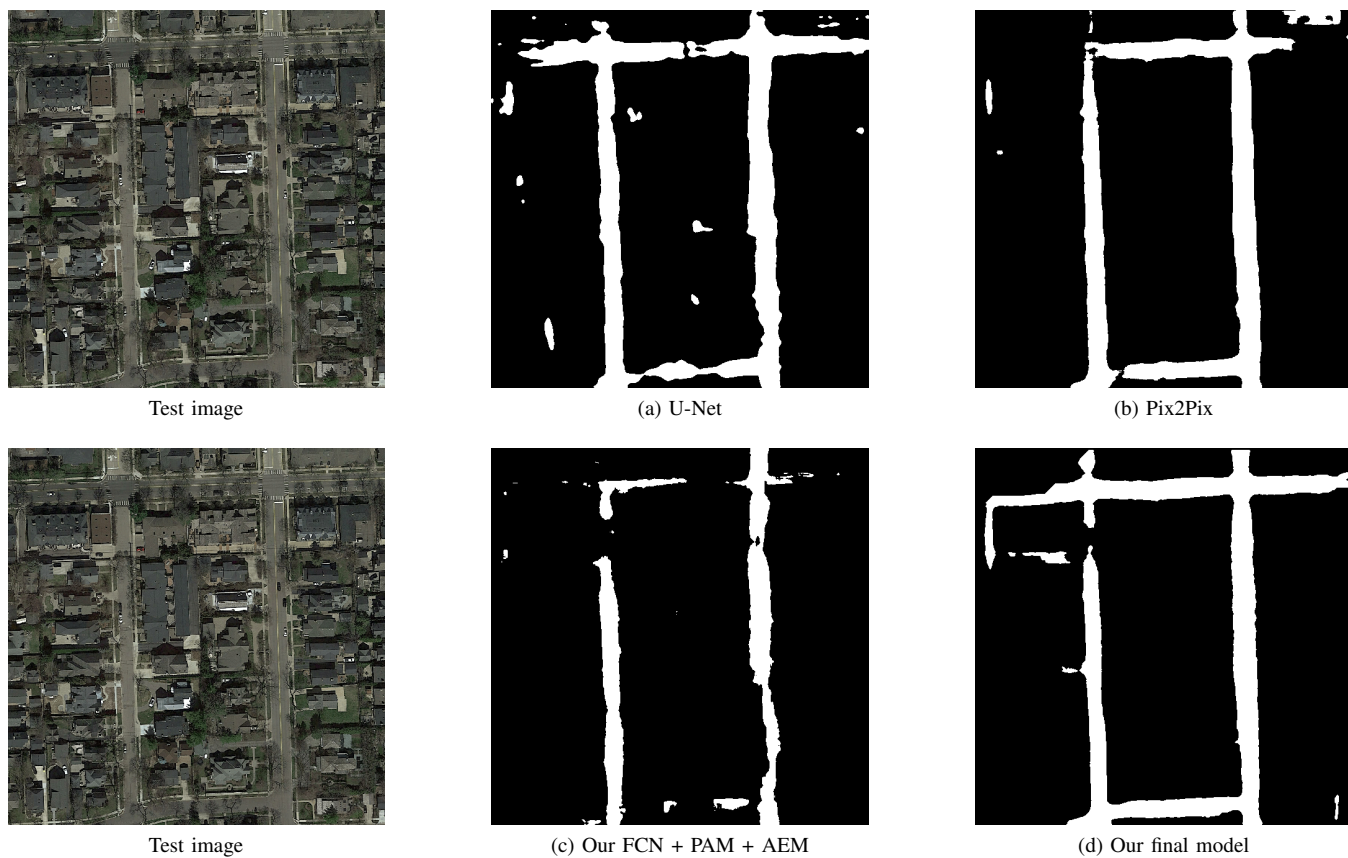


Figure 3: Comparison between the produced masks of the baselines (a) and (b). (c) corresponds to our proposed network with no pre-/post-processing or pre-training and (d) is our proposed network with pre-training, data augmentation, TTA and post-processing.



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):

Road Segmentation in Aerial Images

Authored by (in block letters):

For papers written by groups the names of all authors are required.

Name(s):

Zhou

Walesch

Jokic

First name(s):

Thomas

Yan

Stefan

With my signature I confirm that

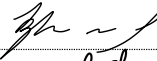
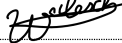

- I have committed none of the forms of plagiarism described in the '[Citation etiquette](#)' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

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

Place, date

Zürich, 22.07.2021

Signature(s)

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