# Project Report: Squeeze and Excitation with Dilated Convolutions for Road Segmentation in Aerial Images

**ETH Zurich, Computational Intelligence Lab,**
**Project 3, Group: Optimus Prime, Spring Semester 2021**

Martin Bucher[1], Gabriela Krasnopolska[1], Valentin Weiss[1], and Matthias König[1]

[1]Department of Computer Science, ETH Zürich

***Abstract*—The extraction of road information solely from aerial images has been an emerging research topic in the past decade, thanks to the advent of deep learning and more precisely, convolutional neural networks. In this report, we present our results of project 3 as part of the Computational Intelligence Lab course at ETH Zurich during spring semester 2021. We present *ExtLinkNet*, a new deep convolutional architecture that allows us to explore the usage of *Squeeze-and-Excitation* blocks as well as dilated convolutions for the task of road segmentation. We empirically evaluate our model and compare it to four state-of-the-art baselines, achieving a patch accuracy of 0.93335 on the final Kaggle challenge.**

## I. Introduction

Image segmentation has gained a lot of attraction in the past decade due to the massive performance boost brought by deep learning (DL) and specifically convolutional neural networks (CNNs) as a subclass of it [1]–[5]. DL provides state-of-the-art performance for many different tasks in the field of computer vision, such as image classification, object detection, image segmentation, and tracking. Semantic image segmentation is one particular type of image segmentation and constitutes the task of predicting pixel-wise labels for a given input image. Semantic image segmentation has many applications in the fields of medical imaging, object detection, face recognition, and remote sensing, wherein the latter the automatic extraction of road information (e.g., the information if a given pixel belongs to a road or not) has important use cases for vehicle navigation, autonomous driving, geographical information systems, and urban planning, to name just a few.

Various methods for road segmentation have been proposed using CNNs as a base architecture, and the encoder-decoder structure has shown some of the strongest results in recent years [6]–[8]. U-Net was first introduced to tackle image segmentation on biomedical data [9] and has since then been repeatedly extended for usage on road segmentation [6], [10], [11]. Another architecture that has produced state-of-the-art results is LinkNet [7]. Different extensions and building blocks have been added or changed such as a middle block using dilated convolutions [12], non-local operations [13] and regional attention networks, as well as asymmetric filters [8], [14], [15].

We tackled the task of road segmentation within the context of project 3 of the Computational Intelligence Lab course in its 2021 edition. Given a set of RGB aerial images of urban areas, the task was to predict for a given patch of dimension 16x16 pixels whether that patch contains a road area (1) or not (0). A patch is defined to contain road area if more than 25% of the pixels in it contain road. We solved this task by making pixelwise binary predictions and mapping to 16x16 patches subsequently.

## II. Method

### A. Dataset

The initial dataset provided for project 3 consists of 100 RGB images of size 400x400 pixels together with ground-truth binary segmentation masks of the same dimension for the training set and 96 RGB images of size 608x608 pixels without any ground-truth masks. The images in both sets have roughly the same scale. Hence, as the images from the test set are larger, they contain more information while being at the same scale semantically. Because most of the considered models — trained solely on this small training dataset — showed strong indications of overfitting, we used 2000 additional RGB images retrieved via the Google Maps Static API [16], inspired by last year's project from [17]. The additional dataset contains images from Los Angeles, Chicago, Houston, Phoenix, Philadelphia, San Francisco, and Boston, as those cities look similar to the provided training and test data. As the test set contains a remarkably high amount of parking lot images (over 50%), we decided to further augment the training set with 82 images showing parking lots in Chicago, Atlanta, and Houston. We subsequently refer to the original dataset as `CILonly` and to the larger, combined dataset as `CIL+Gmaps`.

### B. Model Architecture

Our proposed model architecture is based on D-LinkNet [12] and is illustrated in figure 1. It exploits an encoder-decoder structure as many other state-of-the-art road segmentation networks. By processing the image at different resolutions, it enables the computation of more abstract

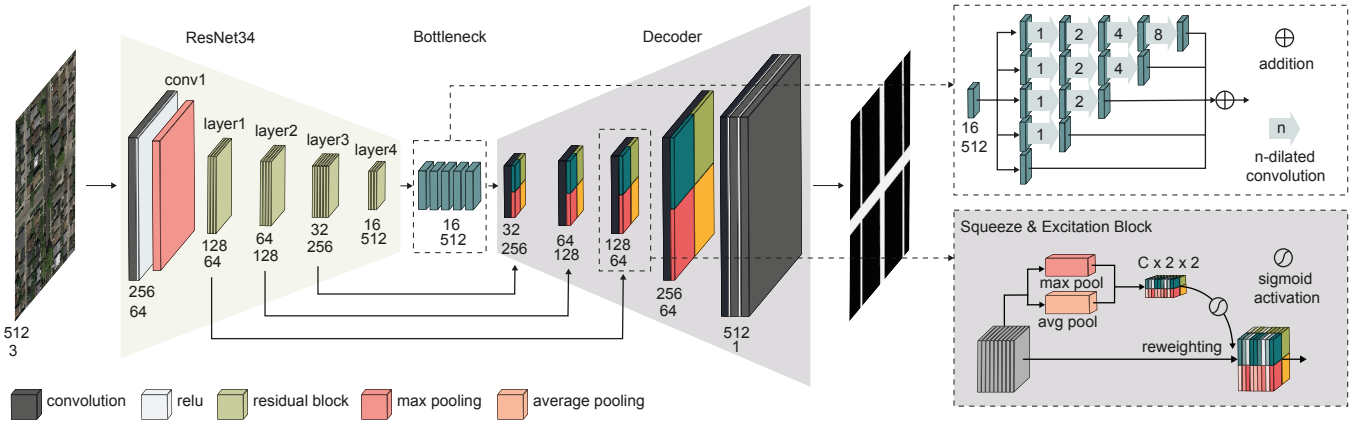



Figure 1: Overview of *ExtLinkNet34* with the image sizes and number of channels throughout the network.

feature maps. We modified the D-LinkNet decoder by inserting Squeeze-and-Excitation (SE) blocks [14] to each stage, as they improved the performance of high-resolution road segmentation tasks [8]. Specifically, we assumed this to be useful for images with parking lots since these are harder to predict and only make up part of the image. Our model extends D-LinkNet, therefore we refer to it as *ExtLinkNet34* and *ExtLinkNet50*, depending on the used encoder.

*1) Encoder:* The encoder is based on either ResNet34 or ResNet50 [1], resulting in 4 stages, one for each ResNet layer. At each layer, we downsample and further transform the input by using multiple convolutions, as proposed in the original ResNet architecture. We use skip connections, meaning that the output of each encoder stage is added up to the output of every decoder stage. Our encoder was pretrained on ImageNet [2], availing the properties of transfer learning.

*2) Bottleneck:* The bottleneck consists of dilated convolutions to grasp more information from pixels that are further apart from each other. It concatenates dilated convolutions with iteratively higher dilation [1,2,4,8,(16)]. Finally, the outputs of each convolution are added together with the input to the bottleneck and given to the decoder.

*3) Decoder:* The decoder is again based on the D-LinkNet architecture using deconvolutions to upsample the feature maps at each stage. Our architecture differs from D-LinkNet by adding a *Squeeze-and-Excitation* (SE) block at each layer. The SE block was first introduced in [14] and applied to the task of high-resolution road segmentation in [8]. The proposed block derives weights from different channels and regions of the feature map. More specifically, given a feature map, the SE block is computed by first splitting the map into 2x2 or 4x4 regions and then downsampling the tensor using max and average pooling. Finally, the SE block is obtained by element-wise addition of both max and average pooling blocks. After applying the sigmoid function on the block, the elements are multiplied with their corresponding region and channel in the feature map, allowing to give different channels and spatial regions more or less weight.

### C. Preprocessing

All images are resized to either 512x512 pixels or 384x384, which are the input sizes for all given models. The compared baselines use a resolution of 384x384, while our proposed architecture uses 512x512. We tried out several different dimensions but finally opted for 512x512, as it is repeatedly divisible by 2 and accommodates downsampling and splitting the feature map into evenly sized regions in the SE blocks. The RGB values are normalized to lie in the range of 0 to 1. We applied random data augmentation on the fly for both `CILonly` and `CIL+Gmaps` on all models we evaluated. For the augmentation, we randomly flip the image together with its ground-truth mask, both horizontally and vertically with a probability of 0.5, and randomly rotate it by either 0, 90, 180, or 270 degrees with a probability of 0.25 for each rotation. We further experimented with RGB color standardization, both global standardization and instance standardization [18]. As experiments on our first models showed difficulties labeling roads covered by trees, we also experimented with data augmentation that inserts trees into the training images.

### D. Implementation and Training

All training data is randomly shuffled and used during training with a batch size of 3. Networks were trained with a different number of epochs since for different architecture sizes, varying numbers of iterations were needed until the model started to converge and overfit on the training data. The best model during training was picked by the lowest validation loss. As there was no specific validation set provided, we set the split ratio to 10% for both the original (small) `CILonly` dataset and the larger `CIL+Gmaps` dataset consisting of 10 and $0.1 * (2000 + 100 + 82) = 218$ images for the validation phase respectively.

All models were implemented in Python using PyTorch and trained on the Leonhard cluster at ETH Zurich with

| **Baseline Models** | Dataset | TTA | RGB Stand. | # of Params | **Pixelw. Acc.** | **Patchw. Acc.** | **F1 Score** | **Kaggle Score** |
|---|---|---|---|---|---|---|---|---|
| U-Net | `CILonly` | ✓ | ✓ | 31,037,633 | 0.95044 | 0.94171 | 0.73186 | 0.80201 |
| U-Net | `CIL+Gmaps` | ✓ | ✓ | 31,037,633 | **0.95258** | **0.94381** | **0.82194** | **0.91777** |
| ResUnet | `CILonly` | ✓ | ✓ | 13,043,009 | 0.94458 | 0.93750 | 0.80383 | 0.84967 |
| ResUnet | `CIL+Gmaps` | ✓ | ✓ | 13,043,009 | 0.94346 | 0.93356 | 0.67107 | 0.90461 |
| RCNN-Unet | `CIL+Gmaps` | ✓ | ✓ | 7,700,161 | 0.93490 | 0.92359 | 0.62010 | 0.87295 |

Table I: Baselines trained on `CILonly` and `CIL+Gmaps` with *Instance Standardization*. Pixelwise accuracy, patchwise accuracy, and F1 score were computed on the validation set. The Kaggle score is the 16x16 patchwise accuracy computed for the public 49% of the test set.

varying types of GPUs (in most cases NVIDIA GeForce GTX 1080). The models were trained for a maximum of 40 hours using ADAM as an optimizer with a learning rate of 1e-4, which was multiplied by a factor of 0.1 at epoch 45 and again by 0.1 at epoch 80.

### *E. Testing and Postprocessing*

The data from the test set has dimension 608x608 pixels and hence a different size than the 400x400 training data. Naively changing the scale of the test images leads to poor performance as the zoom of the test data then differs from the one of the training data. Thus, we apply the following strategy: First, each test image is augmented by rotating it by 0, 90, 180, and 270 degrees and applying a horizontal flip, resulting in a total of 8 images for a single test image. Then, five different crops of dimension 400x400 are extracted from each image at the top left, top right, bottom left, bottom right, and center. Predictions are made on all of the crops obtaining $8 * 5 = 40$ masks of size 400x400. For each of the 8 images, we place the 5 crops to their original location, compute the mean and obtain 8 segmentation masks of dimension 608x608. After rotating and flipping them back to their original orientation, we obtain a final single prediction mask. We refer to this approach as Test Time Augmentation (`TTA`).

Additionally, we also tried a postprocessing technique using the Simple Linear Iterative Clustering (SLIC) superpixel segmentation algorithm, introduced in [19] and used by [20] for the postprocessing of satellite image segmentation. The idea is too smooth the segmentation masks according to the average classification in each superpixel as nearby pixels from the same superpixel are likely to belong together according to their RGB values.

## III. Experiments

The evaluation of our different models is done on three different metrics: Pixelwise accuracy, 16x16 patchwise accuracy, and F1 score, all computed on the validation set. The reported Kaggle score is the 16x16 patchwise accuracy on the public 49% of the test set.

### *A. Baselines*

To compare the performance of our approach to existing model architectures, we implemented three state-of-the-art baselines for this report:

1) U-Net: The given U-Net based example architecture for project 3 from the Google Colab notebook [21] provided for this years course.
2) ResUnet [10], which is a model based the original U-Net, but with additional residual units inspired by the ResNet architecture [1].
3) RCNN-Unet, which incorporates a RCNN unit into the vanilla U-Net framework.

U-Net and ResUnet were trained with a Binary Cross-Entropy (BCE) loss, RCNN-Unet with a Dice loss. Additionally, we compared our model to D-LinkNet34, as we have based its design on the D-LinkNet architecture.

### *B. Results and Discussion*

*1) Baselines:* As a first step in our experiments, we evaluated the given baseline U-Net and ResUnet architectures on both `CILonly` and `CIL+Gmaps` together with RGB *Instance Standardization*, and *TTA* on the test set, as those properties showed the best performance across all experiments on the three baseline models. The results are presented in Table I. Both U-Net and ResUnet already achieve stable results on the validation set trained with `CILonly`. However, using `CIL+Gmaps` leads to a significant improvement on the Kaggle score and the validation scores for U-Net. This motivates the training of subsequent models only on the larger `CIL+Gmaps` dataset, as they achieve better generalization. A possible explanation of the performance improvement on the test set is that there are proportionally more parking lots in the test set than in the original training set and more of them within the `CIL+Gmaps`. Additionally, a larger dataset reduces the risk of overfitting, which is especially important for larger and more complex models.

| **Loss** | **Pixelw. Acc.** | **Patchw. Acc.** | **F1-Score** | **Kaggle** |
|---|---|---|---|---|
| BCE | 0.95208 | 0.94493 | 0.73963 | 0.92260 |
| Dice | 0.95403 | 0.95013 | **0.77450** | 0.92501 |
| DiceBCE | 0.95370 | 0.94950 | 0.75321 | **0.93061** |
| Tanimoto | 0.95364 | 0.94900 | 0.76567 | 0.92088 |

Table II: Metrics obtained when training *ExtLinkNet50* with 2x2 spatial SE block regions using different losses.

*2) Losses:* In order to find the most suitable loss for our new architecture, we performed a comparison of three losses commonly applied in road segmentation: BCE loss, Dice

| Model | TTA | RGB Stand. | Loss | Backbone | SE Blocks | # of Params | **Pixelw. Acc.** | **Patchw. Acc.** | **F1-Score** | **Kaggle** |
|---|---|---|---|---|---|---|---|---|---|---|
| D-Linknet34 | ✓ | ✗ | DiceBCE | Resnet34 | 0 | 31,273,281 | 0.95614 | 0.95225 | 0.76552 | 0.93205 |
| D-Linknet34 | ✓ | ✓ | DiceBCE | Resnet34 | 0 | 31,273,281 | 0.95563 | 0.95164 | 0.76496 | 0.93165 |
| ExtLinknet34 | ✓ | ✓ | DiceBCE | Resnet34 | 2 | 31,273,281 | 0.95477 | 0.94984 | 0.76472 | 0.92865 |
| ExtLinknet34 | ✓ | ✓ | DiceBCE | Resnet34 | 4 | 31,273,281 | 0.95492 | 0.95026 | 0.76026 | 0.92724 |
| ExtLinknet50 | ✓ | ✓ | DiceBCE | Resnet50 | 2 | 220,467,233 | 0.95370 | 0.94950 | 0.75321 | 0.93061 |
| ExtLinknet34 | ✓ | ✗ | DiceBCE | Resnet34 | 2 | 31,273,281 | 0.95502 | 0.95077 | 0.75376 | 0.93169 |
| ExtLinknet34 | ✓ | ✗ | DiceBCE | Resnet34 | 4 | 31,273,281 | 0.95501 | 0.95029 | 0.76417 | 0.93100 |
| ExtLinknet50 | ✓ | ✗ | DiceBCE | Resnet50 | 2 | 220,467,233 | 0.95426 | 0.95003 | 0.76370 | **0.93262** |

Table III: Model Architectures all trained on `CIL+Gmaps`. Metrics are the same as in Table I.

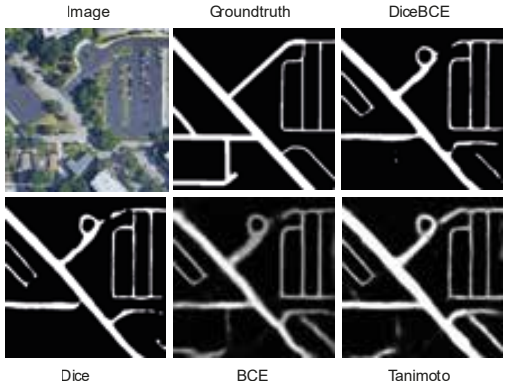



Figure 2: Visual comparison of the predicted masks when training *ExtLinkNet50* with 2x2 spatial SE block regions with different losses.

loss, and DiceBCE as a simple sum of both BCE and Dice loss. Thus, the losses are given by

$$L_{BCE} = -\frac{1}{N}\sum_{i=1}^{N}(\log p^{(i)} + (1-t^{(i)})(1-\log p^{(i)})) \tag{1}$$

$$L_{Dice} = 1 - \frac{2|T \cap P|}{|T| + |P|} \tag{2}$$

$$L_{DiceBCE} = L_{Dice} + L_{BCE} \tag{3}$$

where T is the ground truth, P the predictions, t and p the ground truth and predictions of the i-th pixel respectively.

Additionally, we evaluated the usage of the Tanimoto loss presented in [22]. Table II shows the metrics obtained when training an *ExtLinkNet50* with 2x2 spatial SE block regions on `CIL+Gmaps`. A visual comparison of the predictions can be found in figure 2. Both BCE and Tanimoto lead to blurry masks, which is not the case for the two Dice loss-based variants. Dice loss also achieves the best F1 score. This might be due to the fact that Dice loss performs better in presence of class imbalance [23], and the percentage of roads is significantly lower than non-road (no weighting was performed) in all images. However, using the BCE loss during training leads to notably faster convergence. The best Kaggle results are achieved when using DiceBCE, which is why we use it for all following models.

*3) ExtLinkNet Comparison:* We evaluated different versions of our proposed model architecture and compared them to the framework they build upon, namely D-LinkNet. The results are shown in table III. Interestingly, D-LinkNet34 and all of our *ExtLinkNet* variants perform slightly better without standardization than with standardization, which in contrast, improved the performance of all three baseline models. All evaluated *ExtLinkNet* models show similarly good performance and achieve better results than the three baselines U-Net, ResUnet, and RCNN-Unet, apart from the F1 score on the validation set. On the public Kaggle score *ExtLinkNet50* with 2x2 spatial SE block regions and without *Instance Standardization* achieves the overall best result. We were able to improve the Kaggle score to 0.93335 by applying a majority voting on the predictions of the three *ExtLinkNet* variants trained without *Instance Standardization*. However, all our ResNet34 based models are outperformed by the base model D-LinkNet34, and adding more spatial SE block regions worsens the performance. We suspect that this is due to the significantly lower resolution of our input images (400x400 pixels) compared to for instance DBRANet which specifically applies the concept of SE blocks to high-resolution images (1024x1024 pixels).

*4) Postprocessing:* We evaluated SLIC as a method for postprocessing the predicted segmentation masks. We determined the best parameters for a given model using grid search on the training set. However, this method did not improve the Kaggle score and produces rather fuzzy masks instead of smoothing them out.

## IV. Conclusion

In this report, we presented a new model architecture named *ExtLinkNet*. With D-LinkNet as the base, we added SE blocks in the decoder part as used in DBRANet with the aim to give different channels and spatial regions different weights. As opposed to DBRANet we used dilated convolutions at the bottleneck part of the network. Our proposed model architecture showed comparable results with respect to the considered baselines and D-LinkNet34. We further introduced a prediction pipeline using Test Time Augmentation (TTA) and compared four different losses on the same network architecture and briefly discussed their differences. In summary, although the gain on performance by using SE blocks is not very evident, our proposed architecture improves on the public Kaggle score by a small margin compared to D-LinkNet34.

## REFERENCES


[1] 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.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[3] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 801–818.

[4] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 3431–3440, iSSN: 1063-6919.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, May 2017. [Online]. Available: https://doi.org/10.1145/3065386

[6] Z. Zhang, Q. Liu, and Y. Wang, "Road Extraction by Deep Residual U-Net," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 5, pp. 749–753, May 2018, conference Name: IEEE Geoscience and Remote Sensing Letters.

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

[8] S.-B. Chen, Y.-X. Ji, J. Tang, B. Luo, W.-Q. Wang, and K. Lv, "DBRANet: Road Extraction by Dual-Branch Encoder and Regional Attention Decoder," *IEEE Geoscience and Remote Sensing Letters*, pp. 1–5, 2021, conference Name: IEEE Geoscience and Remote Sensing Letters.

[9] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[10] F. I. Diakogiannis, F. Waldner, P. Caccetta, and C. Wu, "Resunet-a: A deep learning framework for semantic segmentation of remotely sensed data," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 162, pp. 94–114, 2020.

[11] N. Y. Q. Abderrahim, S. Abderrahim, and A. Rida, "Road Segmentation using U-Net architecture," in *2020 IEEE International conference of Moroccan Geomatics (Morgeo)*, May 2020, pp. 1–4.

[12] L. Zhou, C. Zhang, and M. Wu, "D-LinkNet: LinkNet With Pretrained Encoder and Dilated Convolution for High Resolution Satellite Imagery Road Extraction," 2018, pp. 182–186. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2018_workshops/w4/html/Zhou_D-LinkNet_LinkNet_With_CVPR_2018_paper.html

[13] Y. Wang, J. Seo, and T. Jeon, "NL-LinkNet: Toward Lighter But More Accurate Road Extraction With Nonlocal Operations," *IEEE Geoscience and Remote Sensing Letters*, pp. 1–5, 2021, conference Name: IEEE Geoscience and Remote Sensing Letters.

[14] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation Networks," 2018, pp. 7132–7141. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2018/html/Hu_Squeeze-and-Excitation_Networks_CVPR_2018_paper

[15] X. Ding, Y. Guo, G. Ding, and J. Han, "ACNet: Strengthening the Kernel Skeletons for Powerful CNN via Asymmetric Convolution Blocks," 2019, pp. 1911–1920. [Online]. Available: https://openaccess.thecvf.com/content_ICCV_2019/html/Ding_ACNet_Strengthening_the_Kernel_Skeletons_for_Powerful_CNN_via_Asymmetric_ICCV_2019_paper.html

[16] G. LLC. Google maps static api. [Online]. Available: https://developers.google.com/maps/documentation/maps-static/overview

[17] D. Chiappalupi, E. Iannucci, G. Lain, and S. G. Piazzetta, "U-net based network ensemble for satellite road segmentation," 2020, project 3, Computational Intelligence Lab 2020, group: PochiMaPochi.

[18] D. Ulyanov, A. Vedaldi, and V. S. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *CoRR*, vol. abs/1607.08022, 2016. [Online]. Available: http://arxiv.org/abs/1607.08022

[19] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slic superpixels," Tech. Rep., 2010.

[20] M. Längkvist, A. Kiselev, M. Alirezaie, and A. Loutfi, "Classification and segmentation of satellite orthoimagery using convolutional neural networks," *Remote Sensing*, vol. 8, no. 4, p. 329, 2016.

[21] E. Z. Data Analytics Laboratory. Project 3 - road segmentation. [Online]. Available: https://colab.research.google.com/github/dalab/lecture_cil_public/blob/master/exercises/2021/Project_3.ipynb

[22] F. I. Diakogiannis, F. Waldner, P. Caccetta, and C. Wu, "Resunet-a: A deep learning framework for semantic segmentation of remotely sensed data," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 162, pp. 94–114, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0924271620300149

[23] S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

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):

Project Report: Squeeze and Excitation with Dilated Convolutionsfor Road Segmentation in Aerial Images

**Authored by** (in block letters):

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

| **Name(s):** | **First name(s):** |
|---|---|
| König | Matthias |
| Krasnopolska | Gabriela |
| Bucher | Martin |
| Weiss | Valentin |

With my signature I confirm that

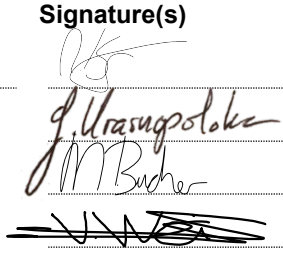
- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

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

**Place, date**

Zürich, 31.07.21

**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.*