

Computational Intelligence Lab

Road Segmentation – Project Report

Christian Bohn
cbohn@student.ethz.ch

Olivier Bitter
bittero@student.ethz.ch

Cyril Schroeder
cyrils@student.ethz.ch

Roknoddin Azizibarzoki
azizibar@student.ethz.ch

ABSTRACT

In this report, we present a deep learning model for pixelwise semantic segmentation of satellite images into "roads" and "background". To this end, we used a Convolutional Neural Network (CNN) inspired by the U-Net architecture, which is a state-of-the-art approach to this sort of image segmentation problem.[1] Due to the very small size of the provided dataset, we first focused our efforts on the preprocessing and augmentation of the training set, which yielded noticeably improved results. Further keys to success were the implementation of regularization methods and model adaptations, such as spatial dropout, adapted class weights in the loss function, data whitening based on zero-phase component analysis, as well as dilated convolutions. Finally, we averaged the predictions of the nine best models to produce our final results, which our best entry for the Kaggle competition is based on. The public and private F_1 -scores of our best submission were 90.52% and 89.61%, respectively corresponding to ranks 13 and 6 out of 30 participating teams.

1 INTRODUCTION

Semantic segmentation is the task of clustering parts of images together which belong to the same object class.[2] Segmentation problems are among the more challenging problems in the field of computer vision, and have applications in diverse disciplines such as biomedical imaging analysis, autonomous driving or cartography.[3–5] A state-of-the-art solution for this sort of problem is to use convolutional neural networks (CNN), such as AlexNet, VGGNet, or ResNet.[6–8] These deep learning models are inspired by biological vision, as connectivities between nodes in the model graphs correspond to the organization of visual cortices in animals.[9] Using small and simple patterns, CNNs are able to learn hierarchically complex patterns.[10]

In our case, the problem of interest consisted of segmenting satellite images into the categories "road" and "non-road" in 16 by 16 patches. We were provided with a dataset of 100 images, along with their groundtruth data, depicting the location of roads in each of the images. To solve the posed problem, we implemented an improved variant of the U-Net architecture.[1] This model architecture provides state-of-the-art segmentation performance as it is able to recognize image features of significantly varying sizes and complexities. The U-Net implementation on its own already performed quite well, but it quickly became apparent that the size of the training data would severely limit the achievable performance. Therefore, we chose to implement extensive data augmentation, which improved results significantly. Furthermore, we implemented data whitening, spatial dropout, an adapted loss function, as well as dilated convolutions, as is further elaborated in chapters 2 and 3.

2 MODEL ARCHITECTURE AND IMPLEMENTATION

Our deep learning model was inspired by the U-Net architecture, a state-of-the-art model for semantic image segmentation.[1]

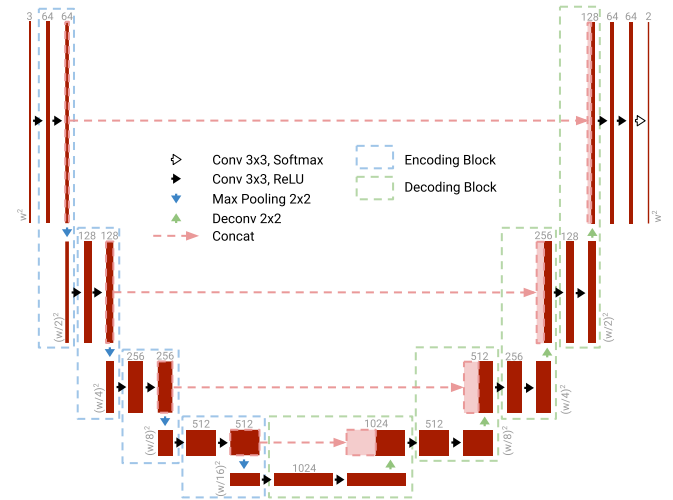


Figure 1: Overview representation of our U-Net architecture.

Like many other segmentation architectures, a U-Net model consists of an encoding stage which produces very high-dimensional, complex descriptions for image regions, followed by a decoding stage, where these representations are transformed back to the original image shape. The first stage consists of four blocks, each reducing the sizes of the spatial dimensions by half, and doubling the number of feature maps. Each of these encoding blocks consists of two convolutional layers and one max-pooling layer. Subsequently, the data tensor is fed into a decoding stage with transpose convolutional layers, where deconvolutions are used to upsample back to the original input size, while halving the number of feature maps in each block.

In comparison to other image segmentation architectures, a U-Net model yields a relatively large receptive field due to the effective doubling of the size of the receptive field with each pooling operation. Furthermore, the concatenations along the channel dimension via skip-connections from the encoding to the decoding stage enable the model to still take high-frequency details in the input image into account. Especially for a complex vision problem, like the one presented here, these are critically important requirements in order for the model architecture to produce high-quality results.

Our adaptation represents a structural improvement over the original U-Net architecture: As long as the height and width of the input images are divisible by 16, our implementation produces an output of the same spatial size as the input without the need to for any explicit padding of the input. This is achieved by configuring each convolutional layer in the model such that its output has the exact same size as its input (or exactly twice the height and width for the deconvolutions).

In order for our architecture to be extensible beyond just binary classification, we chose to produce two logits for each pixel as output of the model graph. After the application of a softmax function, these represent the probabilities of a pixel being labeled as "road" and "non-road", respectively. This is not strictly needed for binary classification, but provides a useful interface for any multi-class segmentation problem to be handled by our architecture in future applications.

Since the training data exhibits a significant class imbalance of about 4 to 1 in the number of "non-road" pixels to the number of "road" pixels, we added class weights to our loss function in order to penalize false negatives more heavily. The used weighted per-pixel cross-entropy loss $L_{\text{pixelwise}}$ is defined as follows:

$$L_{\text{pixelwise}} = -w_r \cdot y_r \cdot \log(p_r) - y_n \cdot \log(p_n) \quad (1)$$

where w_r is the weight with which we want to penalize false negatives, y_r and y_n are indicator variables for the true label of the pixel ("road" or "non-road"), and p_r and p_n are the predicted class label probabilities as yielded by a softmax activation of the logit output of our network.

To produce the final, scalar loss, we compute the mean over the losses of all pixels in a training batch.

To improve robustness, we applied regularization methods such as the random dropout of entire feature maps in the data tensors, also known as spatial dropout, and early stopping by tracking the accuracy on the validation set. Also, we initially implemented batch normalization after each convolutional layer, but did not include it in later models due to its disappointing performance.

In order to increase the size of the receptive field beyond what the standard *U-Net* implementation already provides, we applied dilated convolutions at different dilation rates.

We implemented our model architecture using TensorFlow 1.14, and performed all computations on an NVidia GeForce GTX 1080 GPU.[11]

3 DATA PREPARATION

The provided initial training data contained 100 satellite images and their groundtruths labeled with pixels as "road" and "non-road". All data sets, as well as their processed versions as described below which were used for this project, can be found in the submitted repository.[12]

3.1 Data preprocessing

The first step of our data preprocessing was the manual inspection of the provided groundtruths, revealing evident human errors and inconsistencies. We left the latter, such as parking lots being labeled as "roads" or "non-roads" in a seemingly arbitrary fashion across different samples, as is. However, we manually corrected the former, such as "holes" in the labeling where two roads intersect. Subsequently, of the original 100 images, we curated a validation set consisting of 20 images so that enough rare or unique features would be left in the remaining training set. Additionally, to account for the stark underrepresentation of images with light roads in the training set, whose recognition proved challenging for our models, we selected five images containing such roads from the training set and added them to it twice more, yielding a final training set of 90 images.

The ultimate step of the preprocessing pipeline consisted of

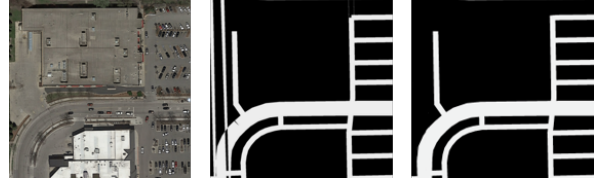


Figure 2: Representative example of manual groundtruth correction. From left to right: Original image, its original groundtruth data, and the manually corrected groundtruth.

whitening the data via zero-phase component analysis (ZCA).[13] The matrix of flattened, normalized, whitened images X_{ZCA} is obtained as follows:

$$X_{ZCA} = U \cdot \begin{bmatrix} \frac{1}{\sqrt{\sigma_1 + \epsilon}} & & & \\ & \frac{1}{\sqrt{\sigma_2 + \epsilon}} & & \\ & & \ddots & \\ & & & \frac{1}{\sqrt{\sigma_n + \epsilon}} \end{bmatrix} \cdot U^T \cdot X \quad (2)$$

where X holds the flattened original images as rows. U denotes the matrix of the left-singular vectors and $[\sigma_1, \dots, \sigma_n]$ denote the singular values of the SVD of the covariance matrix of X . The hyperparameter ϵ is the so-called whitening coefficient.

Applying ZCA to a set of image vectors transforms it into a white noise vector set. Thus, across the entire set, the pixels of all images are uncorrelated and have variance 1 before renormalization. This transformation has the double advantage of removing the strong correlations between neighboring pixels, thereby increasing the training difficulty, and ensures models do not mistakenly recognize erroneous correlations across different images.



Figure 3: Representative example of image whitening via ZCA. From left to right: Original image and whitened versions thereof with whitening coefficient $\epsilon = 0.1$ and 0.001 respectively.

3.2 Data augmentation

As the initial dataset alone was of insufficient size for a robust, high-quality solution to the segmentation task, we had to add an aggressive data augmentation pipeline for generating exponentially many different training samples. Using the *imgaug* library, we generated a set of images with random augmentations on the fly for every training batch.[14]

For each image, a random subset of transformations was selected, and for these transformations, parameter values were randomly sampled from a previously specified range. Image augmentations were performed in two phases. Firstly, we applied affine spatial transformations, which were applied to both image and groundtruth, followed by pixel-value transformations of only the image.



Figure 4: Illustration of different types of image transformations applied for the online augmentation of the training set. From left to right: upper row: original image, flipping, shearing, rotating, translating, zooming; lower row: scaling, CN, saturation and hue change, AGN, channel dropout, pixel dropout.

We chose *horizontal flipping*, *vertical flipping*, *shearing*, *rotating*, *translating*, and *zooming* as affine spatial transformations, some of which were applied in a mutually exclusive manner to limit the generation of unrealistic fringe artifacts. The *shearing*, *rotating*, and *translating* transformations introduced zero-pixels at the image borders, which were filled by symmetrically mirroring the original image. As image-wide pixel-value transformations, we selected *scaling*, *contrast normalization* (CN), *saturation and hue change*, *additive Gaussian noise* (AGN). Furthermore, we chose *pixel dropout* and *channel dropout* as pixelwise value transformations.

We specified the following ranges for the different affine spatial transformation parameters:

- *Shear*: $(-10, 10)$ degrees of shear in negative/positive direction.
- *Rotation*: $(0, 359)$ degrees of rotation.
- *Translation*: $(-10, 10)$ percent translation in negative/positive direction.
- *Zoom*: $(5, 20)$ percent zoom relative to original size.

Ranges for the different pixel-value transformation parameters were the following:

- *Scaling*: $(1.0, 1.5)$: Factor by which pixel intensities are scaled.
- *CN*: $(0.8, 1.5)$: Factor by which contrast is normalized.
- *Saturation and hue change*: $(-10, 10)$: Value added to hue and saturation (independently selected for all channels).
- *AGN*: $(0, 0.05 \cdot 255)$: Intensity of random Gaussian noise added to each pixel.
- *Pixel dropout* $(0, 0.03)$: Probability of masking out pixel values, performed both for single and all channels.

4 RESULTS AND DISCUSSION

Firstly, we implemented three simple semantic segmentation models for baseline accuracy comparison. We used the logistic regression classifier and the CNN with two convolutional + pooling layers and softmax loss as provided in the frame of the exercises for the CIL course. The accuracy on the validation set was 0.648 for the former model, and 0.688 for the latter model. The third implemented baseline model classified each 16 by 16 image patch using three small dense layers: For each patch, it took the $16 \cdot 16 = 256$ pixels with their three color channels as input, flattened them, computed two hidden layers of 256 and 16 activation units respectively, and produced a label of either "road" or "non-road" as output. This model yielded an accuracy of 0.793 on the validation set. All of the accuracies reached by these baseline models were surpassed by a generous margin by our *U-Net* implementations, as described below.

IA	SD	ACW	ZCA	DC	Val. Acc.
-	-	-	-	-	0.893
✓	-	-	-	-	0.922
✓	✓	-	-	-	0.946
✓	✓	✓	-	-	0.938
✓	✓	✓	✓	-	0.952
✓	✓	✓	✓	✓	0.948

Table 1: Comparison of the validation accuracy results obtained using the original *U-Net* architecture with different implementation modifications. Legend: IA: training set image augmentation; SD: spatial dropout; ACW: adapted class weights; ZCA: image whitening using zero-phase component analysis; DC: dilated convolutions; Val. Acc.: validation set accuracy. "✓" indicates the implementation of a given modification in a model, "-" its absence.

As first step for our main model implementation, we experimented with the depth of the original *U-Net* model architecture, *i.e.* we changed the number of encoding and decoding blocks in the neural network. We observed that a shallower model with fewer layers produced noisy results, which is presumably due to the smaller size of the model's receptive field. We undertook no further efforts implementing deeper models, as they would have had too many parameters to be trained efficiently. Hence, we used the same number of encoding and decoding blocks as the original *U-Net* implementation, which corresponds to a model with approximately 31 million parameters.

The subsequent description follows the chronological order of implementation of different modifications to our *U-net* model, each of which providing evident, optically determined improvements to the predictions, as well as generally ameliorated validation accuracies.

Having determined the optimal model depth, we first significantly improved our predictions by implementing the aggressive data augmentation pipeline described in section 3.2. After implementing spatial dropout layers instead of ordinary dropout layers, we observed further compelling improvement in prediction accuracy, as can be seen in Table 1. This improvement over ordinary dropout is deemed to be due to the leaving out of entire feature maps, which limits overfitting.

Upon realizing the obvious class imbalance between "road"- and "non-road"-pixels which would result in an $\approx 80\%$ accuracy for an "all non-road"-prediction, we forced our model to label more pixels as "roads" by penalizing false negatives, as shown in equation 1. Empirically, we found that values for the "road" class weight w_r in the range between 3.5 and 5.0 perform best, which roughly corresponds to the ratio of "non-road"- to "road"-pixels in the groundtruth.

At this point, we observed that our predictions contained small patches of misclassified pixels. To remove such patches and improve the quality of our results, we postprocessed our predictions by applying a morphological opening followed by a morphological closing operation on the model output.[15]

We could further achieve a noticeable increase in prediction quality by whitening our data using zero-phase component analysis,

as described in subsection 3.1. By completely decorrelating the pixels, we removed the strong correlations between neighboring pixels, making our models less hypersensitive to local features and thus more robust. Furthermore, this whitening ensured that our models did not mistakenly recognize false correlations across different images.

As the last major model modification, we implemented dilated convolutional layers, replacing ordinary convolution layers. By further increasing the receptive field size of our models, we observed a very significant improvement in optical quality of our predictions. Up to this point, our models were able to correctly predict "road" patches, which were however not yet contiguous. Following the implementation of dilated convolutions, such patches became connected and the resulting predictions for "roads" finally started resembling realistic high-quality road maps. Having evaluated model performance with different dilation rate parameters, we recognized that a dilation rate of 2 yielded the best results, while a rate of 3 tended to produce more false positives and introduced some artifacts. After adding dilated convolutions, we found the above-mentioned morphological postprocessing to no longer be necessary. Using the software and hardware described in section 2, the training times of our final models were between 6 and 14 hours with early stopping, which we considered practicable.

The final step in our approach to solve the given problem was to implement pixelwise averaging across the predictions of our 9 best-performing models, in order to leverage their different respective strengths.

One should note that while the validation accuracies reported in Table 1 do not follow a strictly increasing trend for additional implementation modifications, the improvements in prediction quality were optically very evident. This discrepancy between validation accuracy and observed prediction quality on the test set is likely due to the small size of the validation set of only 20 images as well as to the sometimes lacking quality and inconsistency of the provided groundtruth data.

Since our model produces pixelwise predictions for input images, we had to apply averaging and thresholding steps to produce the submission files for the Kaggle competition: For each 16 by 16 image patch, we computed the mean value of the pixelwise predictions and assigned a 1-label to it if the mean was greater than or equal to a threshold of 0.7. Empirically, we found that this choice for the threshold value yielded the best results. The public and private F_1 -scores of our best submission were 90.52% and 89.61%, respectively corresponding to ranks 13 and 6 out of 30 participating teams, which we considered highly satisfactory given the very small and imperfect provided training data. We believe that our model's accuracy could still be considerably improved by utilizing a significantly larger dataset as could be found online. However, we feel that looking up and using such additional data would not have been in the spirit of this assignment and competition.

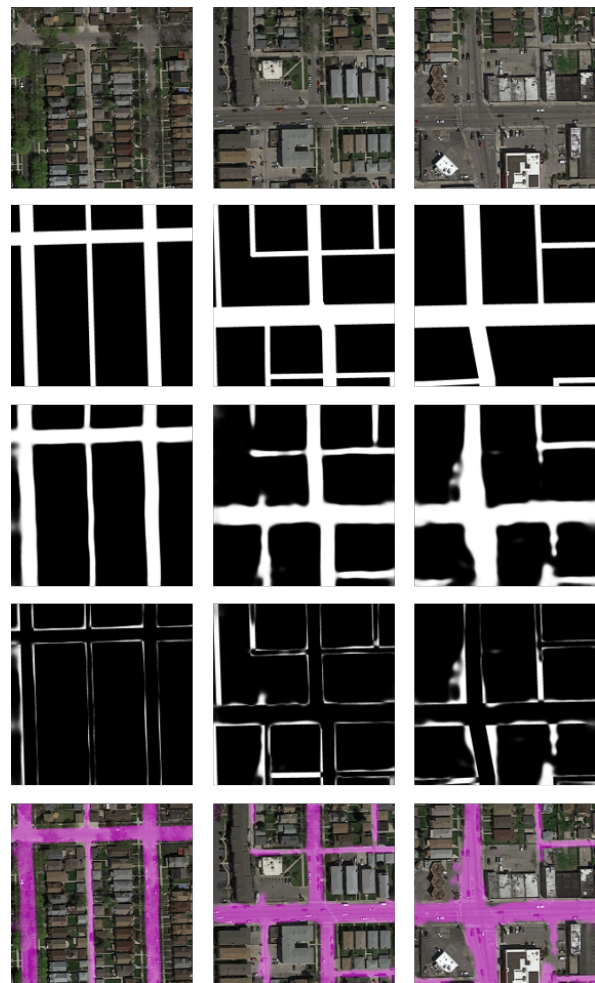


Figure 5: Representative image selection illustrating our results. 1st row: original images from validation set; 2nd row: groundtruth masks; 3rd row: final model predictions; 4th row: differences between groundtruth masks and predictions; 5th row: overlays of predictions onto original images. One can see that our model's predictions largely overlap with the ground truth and that most misclassifications happen in ambiguous image regions that would be difficult to classify even in human inspection. For example, in the third image, our model predicted the regions around the large crossway on the left to be "roads", which a human examiner probably would have done as well, and thus in a sense surpassed the provided groundtruth.

REFERENCES

- [1] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.
- [2] M. Thoma. A survey of semantic segmentation. *CoRR*, abs/1602.06541, 2016.
- [3] S. Acton and N. Ray. Biomedical image analysis: Segmentation. *Synthesis Lectures on Image, Video, and Multimedia Processing*, 4(1):1–108, 2009.
- [4] Ç. Kaymak and A. Uçar. A Brief Survey and an Application of Semantic Image Segmentation for Autonomous Driving. *arXiv e-prints*, page arXiv:1808.08413, Aug 2018.
- [5] N. das Graças Medeiros, E. da Silva, D. Rodrigues, and J. Nogueira. Image segmentation using morphological watershed applied to cartography. In A. Sanfeliu, J. Martínez Trinidad, and J. Carrasco Ochoa, editors, *Progress in Pattern Recognition, Image Analysis and Applications*, pages 156–162, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.
- [6] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

- [7] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv e-prints*, page arXiv:1409.1556, Sep 2014.
- [8] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. *arXiv e-prints*, page arXiv:1512.03385, Dec 2015.
- [9] N. Laskar, L. Sanchez Giraldo, and O. Schwartz. Correspondence of Deep Neural Networks and the Brain for Visual Textures. *arXiv e-prints*, page arXiv:1806.02888, Jun 2018.
- [10] R. Yamashita, M. Nishio, R. Do, and K. Togashi. Convolutional neural networks: an overview and application in radiology. *Insights into Imaging*, 9(4):611–629, Aug 2018.
- [11] P. Barham E. Brevdo Z. Chen C. Citro G. Corrado A. Davis J. Dean M. Devin S. Ghemawat I. Goodfellow A. Harp G. Irving M. Isard Y. Jia R. Józefowicz L. Kaiser M. Kudlur J. Levenberg D. Mané R. Monga S. Moore D. Gordon Murray C. Olah M. Schuster J. Shlens B. Steiner I. Sutskever K. Talwar P. Tucker V. Vanhoucke V. Vasudevan F. Viégas O. Vinyals P. Warden M. Wattenberg M. Wicke Y. Yu X. Zheng M. Abadi, A. Agarwal. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *CoRR*, abs/1603.04467, 2016.
- [12] <https://gitlab.ethz.ch/cbohn/cil-project>.
- [13] K. Pal and K. Sudeep. Preprocessing for image classification by convolutional neural networks. In *2016 IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)*, 2016.
- [14] A. Jung. imgaug. <https://github.com/aleju/imgaug>, 2018. [Online; accessed 02-Jul-2019].
- [15] K. Anuar, A. Jambek, and N. Sulaiman. A study of image processing using morphological opening and closing processes. *International Journal of Control Theory and Applications*, 9:15–21, 01 2016.



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 - PROJECT REPORT

Authored by (in block letters):

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

Name(s):

Bohn

Schroeder

Azizibarzoki

Bitter

First name(s):

Christian

Cyril

Roknoddin

Olivier

With my signature I confirm that

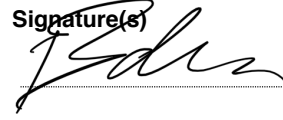
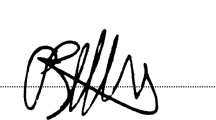
- I have committed none of the forms of plagiarism described in the '[Citation etiquette](#)' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

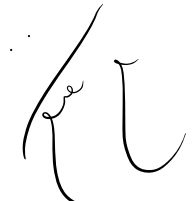
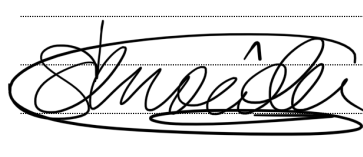
I am aware that the work may be screened electronically for plagiarism.

Place, date

July 5th, 2019

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.