Road Segmentation from Aerial Images

Arda Düzceker, Jonas Hein, Juan Lopez and Marilou Beyeler
Department of Computer Science, ETH Zurich, Switzerland
Email: {ardad, heinj, juanlo, mabeyele}@student.ethz.ch

***Abstract*—Automated semantic segmentation of satellite images is an important research field concerning numerous applications, such as automated map generation and cartographical statistics. In this paper, we tackle the problem of detection of roads in satellite images. We present an ensemble of three original fully convolutional encoder-decoder architectures utilizing some well-known image classification networks as their encoder backbones. Also, we discuss the improvements that we achieve by training with publicly available but imperfectly annotated data as well as many data augmentation techniques. Our best model which is an ensemble gives 0.925 F1-Score (weighted average of public and private score) in the Kaggle competition which is ranked 1st in this year's competition.**

I. Introduction

In computer vision, semantic image segmentation is the classification task, assigning each pixel of an image to a category. Cities have never been growing so fast, and following this trend, their road networks are expanding dramatically. To stay up to date with this fast growth, existing road maps need to be constantly adjusted. For years, this has been done manually. On the other hand, the volume of aerial imagery data captured with airborne or spaceborne platforms is growing, making manual interpretation prohibitive [1] increasing the need for an autonomous road segmentation system. In this work, we try to come up with an adequate machine learning model that eliminates the need of manual labor by classifying each pixel of an RGB high-resolution aerial satellite image as either road or background. This is a challenging problem as roads might, for example, be hidden by trees and cars, have different materials or structures as do land roads compared to city roads. As for many other computer vision research areas, the usage of deep learning techniques has been shown to be extremely powerful for this task, since they can discover such high level, abstract relations while reducing the human effort by removing the necessity of designing task-specific features.

II. Related Work

State-of-the-art semantic segmentation builds on the use of so-called fully convolutional neural networks (FCN), as introduced by J. Long et al. [2]. Their main contribution was that of substituting the last block of neural networks (NN), composed of fully connected layers, by their convolutional counterparts; this way feature maps are obtained instead of scores. Outputs produced by CNNs undergo an upsampling process with a map of the complete scene as result. Architectures following this scheme are commonly known as encoder-decoder. Despite their remarkable performance and elegant design, these models suffer from some deficiencies that have been covered in previous works.

Semantic segmentation requires a comprehensive understanding of scenes; to that end, models such as [3] include an additional postprocessing step in order to refine final predictions. Likewise, inclusion of dilated convolutions translates to an exponential enlargement of the receptive field, allowing the model to gather more context/contextual information [4].

The main drawback of downsampling images during segmentation is the loss of spatial information like exact edge shapes. Fusing low- and high-level features from the encoder significantly helps to cope with this issue [5], [6].

This work closely follows [7], where some of the previously mentioned advancements are embedded, and innovations such as the inclusion of a Spatial Pyramid Pooling block (ASPP) between encoder and decoder enhance performance at different segmentation benchmarks.

III. Models

A. Baseline Models

We compare our model to two baselines: First, an approach based on graph-cut, and second, a deep neural network based on encoder-decoder architecture.

1) Graph-Cut Baseline: Our first baseline implements the Graph-Cut based image segmentation algorithm, proposed by Boykov *et al.*[8]. The implementation is adapted from one of the exercises of Mathematical Foundations of Computer Graphics and Vision [9]. However, instead of interactive user input, the image labels are used to calculate the class a priori probabilities. Overall, the minimum cut of the created graph defines the globally optimal color space segmentation.

2) U-Net Baseline: Our second baseline is based on U-Net, as originally proposed by Ronneberger *et al.*[5] and the implementation is based on [10]. The encoder part uses five blocks, each consisting of two 2D convolutions followed by a max-pooling layer. The first encoder block uses 32 filters for each convolution. For every following encoder block, the number of filters is doubled. Similarly, the decoder part of the network has five blocks each consisting of a 2D deconvolution followed by two 2D convolutions. The first decoder block has 512 filters for its convolutions and the number of filters is halved in every consecutive block, leaving the last decoder block with 32 filters. In total, the network consists of about 31M parameters.

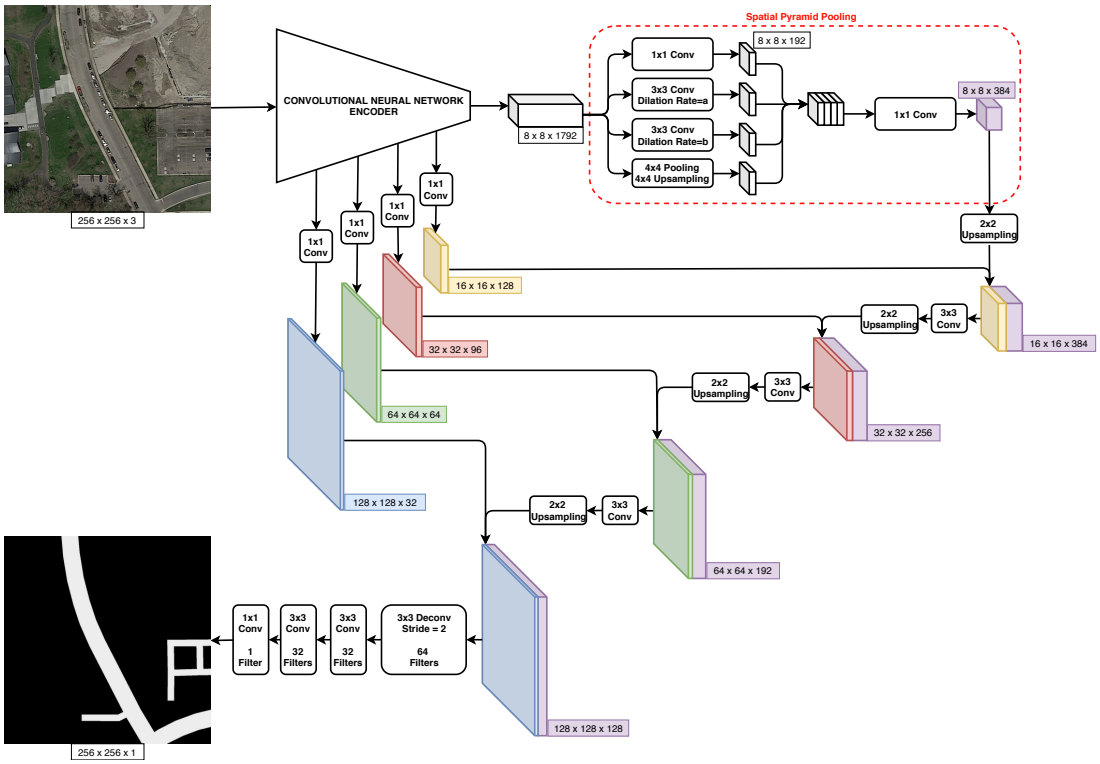


Figure 1. Architecture of our Generic Convolutional Neural Network Encoder with Spatial Pyramid Pooling and Upsampling Decoder. The number of spatial pyramid blocks and the dilation rates a and b depend on the model and are described in the corresponding sections.

B. Our Models

The models that we designed are again based on the encoder-decoder architecture. For such architectures, the encoder component is a crucial part of the network in order to discover high level, abstract relations in the image. There are publicly available, well-known and proved to be successful convolutional neural networks aiming at image classification which can easily be integrated to our models as the encoder backbone. Furthermore, these networks can be initialized with weights pretrained on the ImageNet dataset [11] with no extra effort thanks to Keras library [12]. Although ImageNet classification task and input images are substantially different from our segmentation goal and our input images, using pretrained networks are proved to generally deliver faster and better convergence by helping especially the initial layers to start from a meaningful state [13]. Keeping this in mind, we decided to utilize some of these networks.

1) MobilenetV2 Encoder with Spatial Pyramid Pooling and Upsampling Decoder: Our first model is shown in Figure 1 uses MobilenetV2 [14] as the encoder and employs the Spatial Pyramid Pooling component presented in [7]. The authors discuss that such component enables the interpretation of the features present in the final layer of the encoder in a spatially consistent manner. This is achieved by performing one identity mapping, several dilated convolutions (dilation rates a=2, b=3) and average pooling. The outputs of these parallel operations are then concatenated and fed into decoder network. The decoder network uses bilinear upsampling instead of deconvolution layers which are used in U-Net baseline in order to decrease the number of parameters and prevent overfitting. In decoder network, we also concatenate the output of the corresponding (size-wise) encoder layer to help the decoder to recover spatial dimension in a pixel accurate way similar to U-Net. The model has approximately 14M parameters.

2) Xception Encoder with Spatial Pyramid Pooling and Upsampling Decoder: For our second model, we replaced MobilenetV2 with Xception network presented in [15], which is a larger encoder model with ∼21M parameters compared to ∼5M. Also, our training trials indicated us that feeding not only the output of the encoder backbone network, but also the one before the output (16x16 level) produces promising results. We believe that for spatial pyramid pooling technique to work well, one should have optimal size for the spatial dimension. Therefore, in our second model, we use two spatial pyramid pooling blocks, i.e. yellow tensor in Figure 1 is also produced by a spatial pyramid pooling block (with dilation rates a=3, b=5 and parallel-filters=128, last-filters=256) instead of a simple 1x1 convolution. With approximately 34M parameters this model is significantly larger than the MobilenetV2-based model

presented in Section III-B1.

3) Xception Encoder with Many Spatial Pyramid Pooling Blocks and Upsampling Decoder: We keep the trend of increasing the number of spatial pyramid pooling blocks in our last model. Here, all encoder levels are fed into their respective spatial pyramid pooling blocks, i.e. yellow, red, green and blue tensors in Figure 1 are produced by spatial pyramid blocks instead of 1x1 convolutions (dilation rates, filters and kernel sizes for different levels can be examined in the source code). Also, the number of filters in almost all of the convolutional layers are reduced to keep the size of the network reasonable. Even so, our final model is the largest one with $\sim$38M parameters. One should note that, the models designed in Section III-B2 and Section III-B3 are very data-hungry models and they can easily suffer from overfitting when trained with only the competition data. Instead, they are purely aimed at to be trained utilizing our additional data (see Section IV-B) while still applying strong dropouts.

C. Ensemble and Equivariant Prediction

Our final submission is obtained from an ensemble of the top-3 models that we discussed above. In addition, we exploited the fact that aerial images are equivariant with respect to rotations and flips. While the models should learn this property implicitly during training when random flips and rotations are introduced, our experiments have shown that some road segments are only detected in some orientations of the input image. Therefore, we decided to feed each model with 8 unique orientations of each input image (4 different 90° rotations times 2 options for flipping). Since this step increases the runtime by a factor of 8, this is only done during inference for test images. With this approach, we end up with 24 predictions per image and the per-pixel predictions are then found by majority-voting.

IV. Input Data and Preprocessing

The data set consists of 100 high-resolution, labeled satellite images of mostly suburban areas. The original training images (400×400) are resized to 256×256 after performing desired augmentations. The size of the test images is reduced by a similar factor from 608×608 to 384×384, exploiting the fact that our models are fully convolutional architectures.

A. Augmentation

Since the competition dataset is quite small, several data augmentation techniques are applied during runtime in order to make the models more robust and prevent overfitting. First, we alter the hue, contrast and brightness of the input images by a small, random factor with the goal of making the model more robust against the differences in the inputs' color space. Second, we randomly flip the image and apply 90° rotations, exploiting the equivariance of the data with respect to these transformations. Also, we employ random-size and random-place crops to achieve scale and translation invariance. Moreover, we add random rotations in the range of $[-12, 12]^\circ$. These rotations are introduced because our experiments showed that horizontal and vertical roads were detected much better than the diagonal roads. Note that the rotation range is limited in order to avoid too many black pixels appearing in the corners which is a side-effect of the rotations. Then, a large patch can be cropped from the center of the image without any black pixels in it.

B. Additional Data

The literature suggests that deep convolutional neural networks are to-go method. However, one of the biggest disadvantages of deep learning methods is that they often require lots of data to generalize well. We found that the generalization of our models is not sufficient when we use the competition data only. However, the amount of training data can be significantly increased with publicly available but less accurate, noisy data gathered from sources like Google Maps or Open Street Map. Given that the majority of gathered extra data contains reasonably accurate labels, one can expect the model to learn robustly against the noise present in the publicly available data as [1] suggests. Following the findings of this paper, we decided to gather additional training data using Google Maps API. Since many of the training images show the typical American grid-style road layout, lots of low-rise buildings, suburban residencies and American sports facilities, we gathered 1000 training images each from the largest cities in the US, namely Los Angeles, Chicago, Miami, Dallas and Philadelphia. We excluded New York due to its very unique, diverse cityscape.

The images are randomly sampled within a manually defined bounding box for each city. The zoom level (corresponding to camera altitude or scale) is slightly altered by a random factor. Images with less than 3% road pixels are discarded. In order to validate the findings of [1] and keep the human effort minimal while increasing the model's performance significantly, we utilized the gathered data as it is. Thus, our extra training data contains noisy labels such as misaligned road annotations, false building annotations, as well as reasonably accurate ones (see Figure 2).

Figure 2. Examples of misaligned road annotations (left), false building annotations (middle), and reasonably accurate labels (right) in the additional data set. The corresponding labels are overlaid as red.

Model	Validation RMSE	Private Score	Public Score	Overall Score	Δ Overall Score
All background		0.83063	0.84743	0.83886	
Baseline Graph-Cut	-	0.46547	0.46939	0.46739	
Baseline U-Net	0.0683	0.80626	0.82718	0.81651	
+ Additional Data	0.0479	0.90801	0.91369	0.91079	+0.09428
MobilenetV2 Based Model (only competition data and simple augmentations)	0.0563	0.86120	0.86452	0.86283	
+ Augmentation with Random Rotations in $[-12, 12]^\circ$	0.0505	0.88514	0.89484	0.88989	+0.02707
+ Additional Data	0.0456	0.91532	0.92565	0.92038	+0.03049
Xception Based Model (additional + competition data and all augmentations)	0.0418	0.91376	0.91659	0.91515	
+ 2nd Spatial Pyramid Block	0.0397	0.91976	0.91965	0.91971	+0.00456
+ Many Spatial Pyramid Blocks	0.0411	0.91571	0.92282	0.91919	+0.00405
Ensemble	-	0.92168	**0.92775**	0.92465	
+ 8 Averaging of Equivariant Predictions	-	**0.92184**	0.92766	**0.92469**	+0.00004

Table I
RESULTS OVERVIEW. ALL TEST SCORES ARE REPORTED AS MEAN F1-SCORES ACQUIRED FROM THE KAGGLE COMPETITION. THE THREE MODELS HIGHLIGHTED IN BLUE MAKE UP THE ENSEMBLE.

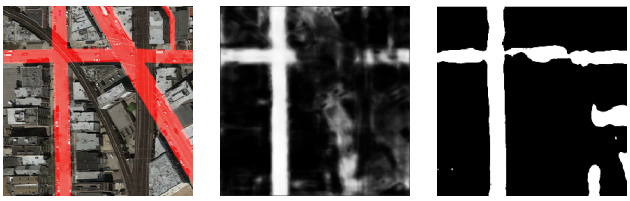


Figure 3. Exemplary prediction of the baseline U-Net model trained with binary cross-entropy loss (middle) and dice loss (right). The left image shows the input with the true labels overlayed in red.

V. TRAINING

All models were trained using dice loss (corresponds to maximizing F1-score), since our initial experiments showed clear advantages of this loss over the others. For instance, using cross-entropy loss results in rather blurry predictions and more pixels labeled as road. In contrast, the predictions based on dice loss are significantly sharper. Exemplary predictions of the baseline U-Net with these losses are displayed in Figure 3.

We keep the training procedure, the Adam optimizer and the dice loss same for all of our best three models. We use the ImageNet pretrained weights for our encoder backbones and freeze these weights initially. First, we train the rest of the network for 15 epochs using only the additional data with a learning rate of 1e-2. Then, we unfreeze all the weights and train the whole network again using only the additional data with a learning rate of 1e-3. Finally, we fine-tune the whole network using only the competition with a low learning of 1e-5. We use early stopping which monitors the validation loss during the latter steps. We use 15% of the utilized dataset as the validation set for the respective step of the training procedure. Furthermore, we utilize frequently placed spatial dropout layers with the dropout rate of 0.25 for MobilenetV2 based model and 0.5 for the Xception based ones to tackle the overfitting issue.

VI. RESULTS AND DISCUSSION

Table-I summarizes the results of our experiments.

U-Net baseline performs quite well, even getting close to our best-performing model when additionally being fed with the whole dataset (competition + additional). On the other hand, Graph-Cut based models performance was significantly poorer than that of the remaining models; for example, grey houses were often incorrectly labelled, considerably increasing the number of false positives. This can be explained by Graph-Cuts shrinking bias, making it hard to segment thin elongated objects [16] .

Models based on Xception and MobileNetV2 achieved the highest scores in our experiments. The addition of several Spatial Pyramid Blocks to the model using Xception as backbone seems to further help extract relevant features up to a point. Note that while the scores obtained by Xception and MobileNetV2 based models are close, the number of parameters of each model greatly differ. Therefore, MobileNetV2 may easily be the preferable choice when one wants to use one the models individually. However, we achieved the overall best score with an ensemble of our three individually best performing models. Also, our results clearly show that the usage of additional data, as well as diverse augmentation techniques, play a key role in the performance. Finally, we observed that equivariant prediction provided only a small improvement, suggesting that our final model was able to robustly learn features independently of their orientation.

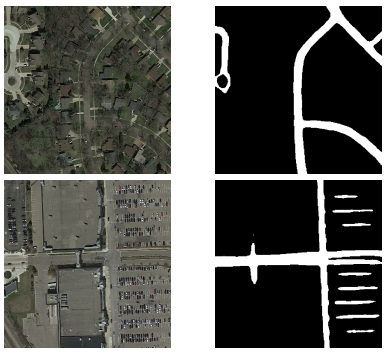


Figure 4. Example Qualitative Results on Test Set

REFERENCES

[1] P. Kaiser, J. D. 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. 55, no. 11, pp. 6054–6068, 2017.

[2] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[3] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[4] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "Enet: A deep neural network architecture for real-time semantic segmentation," *arXiv preprint arXiv:1606.02147*, 2016.

[5] 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.

[6] V. Badrinarayanan, A. Handa, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling," *arXiv preprint arXiv:1505.07293*, 2015.

[7] 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.

[8] Y. Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in nd images," in *Proceedings eighth IEEE international conference on computer vision. ICCV 2001*, vol. 1. IEEE, 2001, pp. 105–112.

[9] "Mathematical foundations of computer graphics and vision." [Online]. Available: https://cgl.ethz.ch/teaching/mathfound19/home.php

[10] Y. Raymond, "Image segmentation with tf.keras," 2018. [Online]. Available: https://github.com/tensorflow/models/blob/master/samples/outreach/blogs/segmentation_blogpost/image_segmentation.ipynb

[11] 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.

[12] "Keras applications." [Online]. Available: https://keras.io/applications/

[13] P. Yakubovskiy, "Segmentation models," 2019. [Online]. Available: https://github.com/qubvel/segmentation_models

[14] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.

[15] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[16] S. Vicente, V. Kolmogorov, and C. Rother, "Graph cut based image segmentation with connectivity priors," *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

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):

Computational Intelligence Lab: Road Segmentation

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

Name(s):	**First name(s):**
Hein	Jonas
Beyeler	Marilou
López Fernández	Juan
Duzceker	Arda

With my signature I confirm that

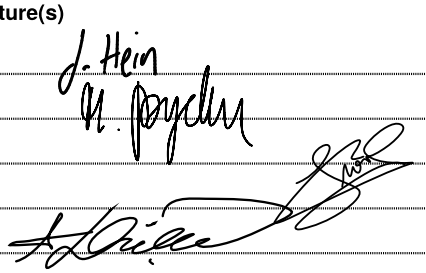
- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

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

Place, date

Zurich, 04.07.2019

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.