Computational Intelligence Laboratory 2019 - Road Segmentation Report

Group: Seggy Roady Projecty
Martin Blapp, Laurin Paech, Nihat Isik, Qais El Okaili
Department of Computer Science, ETH Zurich, Switzerland

***Abstract*—The extraction of information from image and video is a challenging task in computer vision and machine learning. A concrete problem is locating and labeling objects in an image, for example during the conversion of satellite images to road maps. In theory, this allows map-services to enhance and update their images automatically by detecting new roads from updated satellite images as well as to notify of discrepancies between machine-generated and human-generated predictions. Factors such as large variances in road designs, lighting conditions, and occlusions make roads surprisingly challenging to categorize correctly. This project explores the use of various fully convolutional networks for labeling roads in RGB satellite images and concludes with the use of a modified U-Net to give reliable results for even a small data set of training images.**

I. Introduction

Road segmentation of satellite or aerial images is an important task for autonomous driving, infrastructure monitoring, lane-wise traffic management, and urban planning. The low quantity of labeled images and the changing nature of the underlying data makes this a hard problem. Recent work has brought innovative ideas to this field and produced different complex models relying fundamentally on Convolutional Neural Networks [1] [2].

Given satellite images together with the labelled road positions, our task is to train a model capable of pixel-wise predicting roads on satellite images. This is a nontrivial task due to occlusion, other structures looking similar to roads, and the high variance in the visual appearance. We present a novel idea, which extends and improves a common image segmentation model called U-Net [3] significantly, and results in convincing visual road segmentation even with limited training data.

As our novel idea uses an initial encoder-decoder that produces the first predictions followed by secondary encoder-decoder that refines the details. Further, we make use of a technique called "intermediate supervision" to alleviate training issues. We discovered that this approach in combination with dilated convolutional layers helped us getting more accurate road predictions, and helped predicting more continuous road segments.

II. Models and Methods

A. U-Net Implementation

We chose to investigate extensions on a commonly used fully convolutional model, called U-Net[3]. Such a U-Net can be trained end-to-end, and can provide convincing results in biomedical image segmentation even with few training samples. A U-Net has an encoding-decoding structure with a symmetric information flow between similar layers in the encoding and decoding stages. This allows capturing information at every scale, while still enabling precise localization. For our initial codebase, we use a U-Net implementation from GitHub in Keras https://github.com/zhixuhao/unet, which we heavily modified.

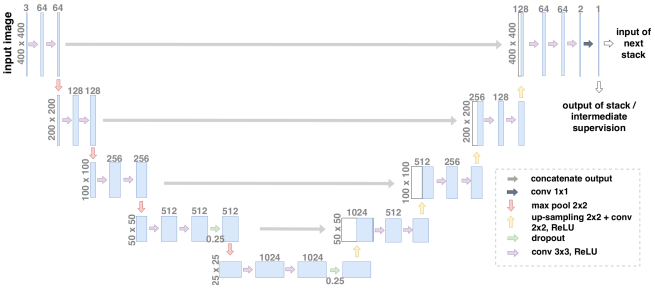

Figure 1: Stacked U-Net Architecture used in our implementation.

Our U-Net consists of four encoding stages, which each have two convolutions with filter size 3x3 and ReLU activation, and finally a max pooling layer of size 2x2. We start with 64 filters, and then double the number of filters at each encoding stage, up to a maximum of 1024 filters. The output of the convolutions of each encoding stage is concatenated to the input of the matching decoding stage. After the four encoding stages we apply dropout of 0.25. Moreover we have two additional convolutions without max pooling, and dropout of 0.25.

Secondly, we have four decoding stages. In each stage, we first up-sample the input by a factor of 2 and apply a convolution filter of size 2x2. Subsequently, we concatenate the output of the matching encoding stage to the output of the convolution and apply two convolutions with filter size 3. After the four decoding stages a 3x3 convolution is added to reduce the number of channels to 2. Finally, a 1x1 convolution with sigmoid activation function which reduces the channel number to 1 acts as the output.

B. Stacked U-Net with Intermediate Supervision

Our novel approach is inspired by state-of-the-art work in human pose estimation, where several stacked encoder-

decoders help to reassess higher order spatial relationships of predictions of previous encoder-decoders. Consistent with the naming in the aforementioned paper, we dubbed this model a "Stacked U-Net". A Stacked U-Net combines two or more U-Nets, where the output of one U-Net flows into the input of the next one. In order to feasibly train such a stacked U-Net with two or more stacks, we use intermediate supervision [4]. The output of each U-Net in the stack is part of the loss function, which allows for repeated bidirectional inference [4]. In the results section, it will be shown that the Stacked U-Net approach helps to get cleaner predictions, and allows infilling of clearly missing parts of predicted roads.

C. Data Augmentation

To address the difficulty of only 100 training images, we relied heavily on perturbations. Similar to [3] we used various types of elastic deformations of the input. Examples are rotations, image flipping, stretching in one or both axis and shearing of the images. Those augmentations were randomly chosen from a range for each image individually, and at each epoch differently. In order to still have valid 400 by 400 pixel input images, we cut off unneeded regions or used mirroring to infill the missing regions.

Figure 2: top left: training image; others: automatically applied perturbations to this image.

For predictions, we additionally employ an ensemble method. Each training image to be predicted is first rotated around 0, 90, 180 and 270 degrees respectively. Additionally, each flip of these rotations is taken. This results in 8 predictions per image, which we combine by using the average of them.

D. Dilated Convolutions

The main limitation of the approach so far is the lack of context of neighboring pixels. It has trouble recognizing bigger objects and is fooled by any local obstruction, such as trees or bridges. Many errors happened on tricky parts of the pictures, such as roads below a railway bridge could not be classified based on the nearby pixels alone. That is the reason why we added dilated convolution filters[5] before each pooling layer (see Figure 2). Dilated convolutions are similar to normal convolutions, but the filter is applied to points that are not contiguous on the image. Each one is separated from its neighbors by a fixed distance called *dilation rate*, and they form a grid of evenly spaced points. With 2 layers and a dilation rate of 2, we get some information from neurons 4 pixels away instead of 2 [6].

E. Input Size Invariance

Another difficulty provided the increased size of the test images of 608 by 608 pixels. We tried different approaches to address this problem. In theory, convolutions should be size invariant if the input is bigger, but we clearly had worse results by just using the 608 by 608 pixel images as input to a network trained with 400 by 400 pixel images. Alternatively, we tried an Overlapping-tile strategy, as implemented in [3]. We found simply resizing the test images to 400 by 400 pixels during predictions provided similarly good predictions while being more efficient. Thus the results presented in this paper will use the resizing of the input to provide the required input size invariance.

F. Training Details

If not noted specifically in the results, training was done using a provided set of 100 aerial images from Google Maps, as well as ground-truth images where each pixel is labeled as either road or background. We trained the model using a pixel-wise binary cross-entropy loss between predictions of probabilities of a pixel being part of a road and the pixel-wise binary ground truth labels. For validation, we used an additional 100 Images, that we generate from the original dataset from which the training images originated. During training, the model with the highest validation accuracy was saved. The training was done with a batch size of 2 using Adam optimizer with a learning rate of 0.001 and default beta values[7]. For results using a leaky ReLU activation function, we used an alpha parameter value of 0.001. Our best performing model was trained on 1000 epochs, which took approximately 9 hours.

III. Results

A. Baseline Comparisons

We will subsequently compare our model to several baseline implementations. A description of these baselines implementations can be found in the Appendix. As baselines

we use a SegNet architecture [8], a basic U-Net implementation and a simple Encoder-Decoder. For all those baseline implementations, we used data augmentation given the limited training data. All baseline implementations were clearly outperformed by our proposed model as listed in Table I.

B. Stacked U-Net

Data Augmentation proved critical, where the 360 rotations, shear, flip and zoom of the training data were crucial for decent results. Specifically this increased the detection accuracy of diagonal roads since our data set contains mostly horizontal and vertical ones.

Stacking two U-Nets improved the visual quality of the results significantly, and the accuracy of the score slightly. As can be shown in Figure 5. the second U-Net can significantly improve on the visual quality of output of the first U-Net. Using an additional third U-Net did not improve the result as can be seen with the performance of U-Net-IX.

The next best source of improvement was dilated convolutional layers. Figure 3 presents the outputs of U-Net-III, on the left when using dilated convolutional layers and on the right without using dilations. It is clear that dilated convolutions helped in the first two images, since the streets are all well connected compared to the non-dilated images on the left. However, this does not always give good results as depicted in the images in the third row, because it can make close streets merge together and thus increase the error in those regions.

Model	Stk	Rot	Aug	Dil	Ens	Lk	**Score**
Encoder-Decoder	-	✓	-	-	-	-	0.8449
Segnet	-	✓	✓	-	✓	-	0.8701
U-Net-I	1	-	-	-	-	-	0.8545
U-Net-II	1	✓	-	-	-	-	0,8689
U-Net-III	1	✓	✓	-	-	-	0.8805
U-Net-IV	2	-	-	-	-	-	0.8494
U-Net-V	2	✓	✓	-	-	-	0.8823
U-Net-VI	2	✓	✓	✓	✓	✓	0,8891
U-Net-VII	2	✓	✓	-	✓	-	0.8895
U-Net-VIII	2	✓	✓	✓	✓	-	**0.8977**
U-Net-IX	3	✓	✓	✓	✓	-	0.8954

Table I: **Stk**: Nr. of Stacks; **Rot**: 360 rotations; **Aug**: Augmentations shift, shear, resize and flip; **Dil**: Dilation Layers; **Ens**: Ensemble Prediction; **Lk**: Leaky ReLu used, otherwise ReLu. **Score** is calculated as a mean of the public score and private score of the Kaggle submissions.

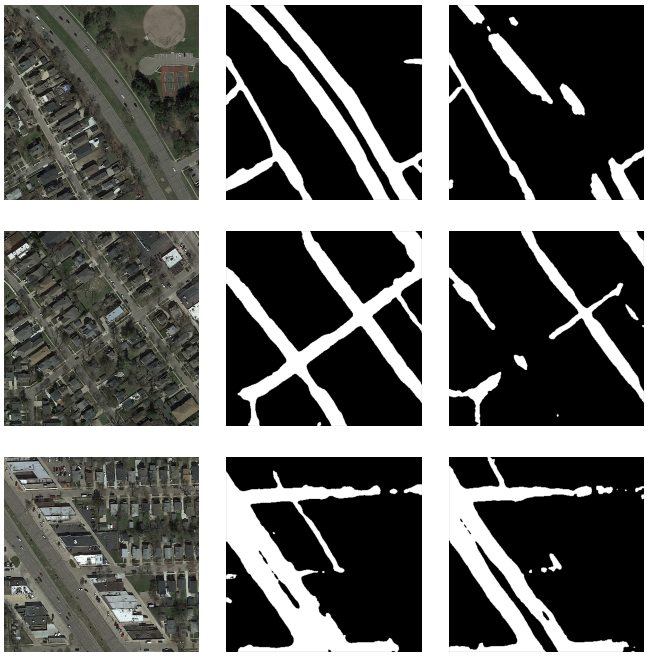

Figure 3: Comparison of test images number 92, 168 and 200 from top to bottom. The final predicted test images in the middle and in the left were predicted using U-Net-III, once with dilated convolutions (middle images) after the pooling layers in the encoding phase and once without dilation (right images).

C. Ensemble Predictions

We can see the improvements introduced by *ensemble* by looking at our models U-Net-V and U-Net-VII in table I. These two models differ only in the ensemble step. In fact, the *ensembled* predictions are more accurate and have less noise around the edges.

D. Additional Training Data

To evaluate our model which was trained using only 100 Images, we also tested training the model with 600 images, which improved our score slightly. This model was subsequently used in the final submission. The provided training images are the upper-left corner of the images 1-109 of the chicago dataset[2]. We generated additional training images by taking patches of remaining images in the dataset, some of them handpicked as they represented rare cases. Overall, training with the additional 500 training images gave us our best Kaggle result (our selected Kaggle score).

IV. Discussion

The Stacked U-Net improves the total details of the segmentation in comparison to the vanilla U-Net (cf. Fig. 5). Therefore, the roads are more continuous, streets are clearly detected and smoothed. Furthermore, it produced good results with only 100 training images.

Although our method achieves good outcomes, they are not uniformly positive (cf. Fig. 5). In spite of the good

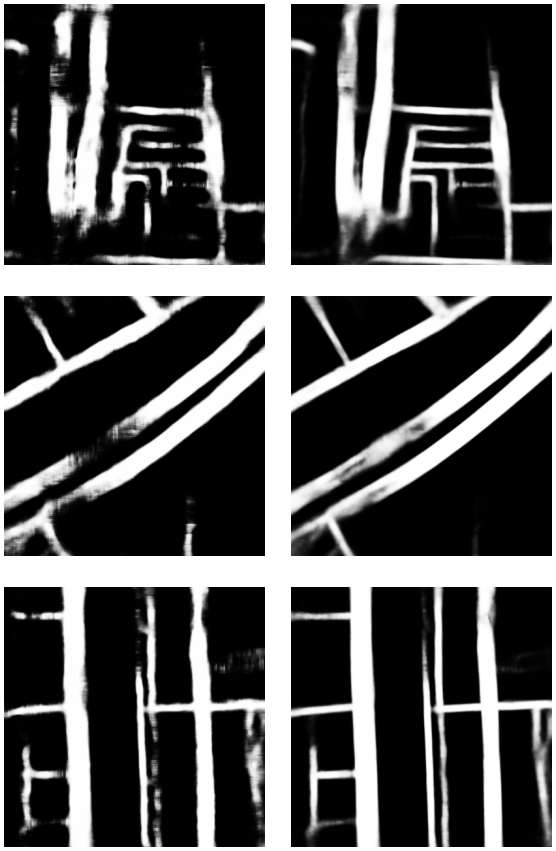

Figure 4: Examples of improved quality of the Stacked U-Net. On the left we have the output of the first U-Net, and on the right the matching output of the second U-Net.

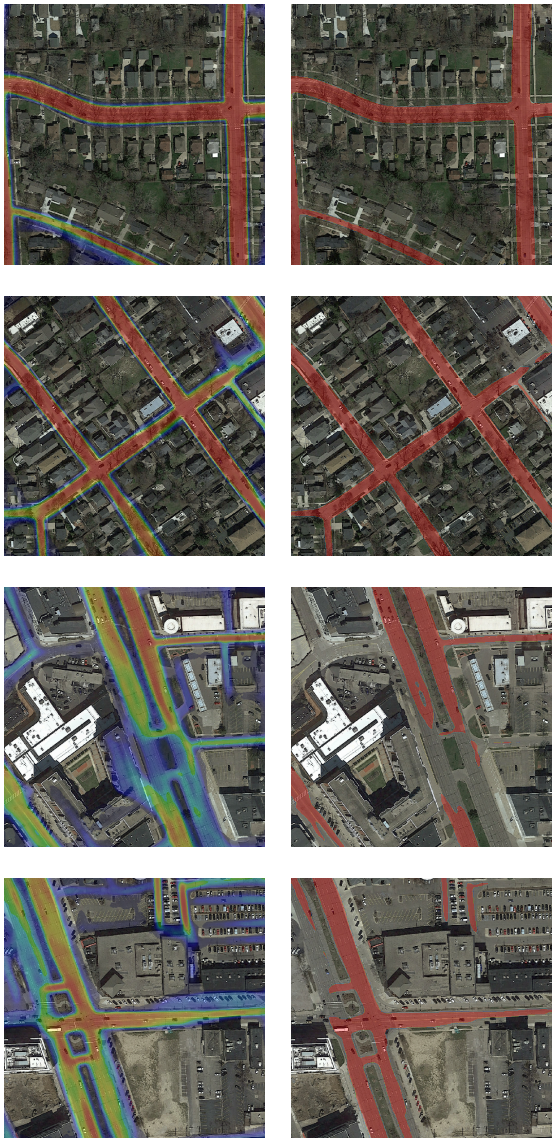

Figure 5: Example results of U-Net-VIII trained on only 100 Images. On the left we have the output probabilities as overlay over the input images. On the right the predicted roads using a probability cutoff of 0.6.

results considering our limited training data, in certain cases, it struggled with generalization challenges such as partial or full occlusion of the streets, shadow creating different illumination and overall rare cases. For example, test data includes images of landing strips, that could be mistaken for streets. Furthermore, the training set suffered from occasionally wrong labeled data and especially parking areas were considerably ambiguous. To mitigate such problems and to observe the overall limitation of our approach, we generated more training data from other data sources.

While we tried to explore further methods of adding additional preprocessed training inputs, such as discrete wavelet transformation or Laplace transformation, we did not have enough time to come to a final verdict. The results, however, looked promising.

V. Summary

As we showed, using a stacked version of U-Net with intermediate supervision leads to qualitatively convincing results with very little data, but requires extensive data augmentation. By using dilated convolutions within the U-Net we can additionally improve the continuity of predicted roads and get information from distant pixels. The definition of a road can be tricky but it seems that road segmentation, in general, is almost a solved problem and the model is sometimes even better than humans.

References

[1] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[2] P. Kaiser, J. D. Wegner, A. Lucchi, M. Jaggi, T. Hofmann, and K. Schindler, "Learning aerial image segmentation from

online maps," *CoRR*, vol. abs/1707.06879, 2017. [Online]. Available: http://arxiv.org/abs/1707.06879

[3] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *CoRR*, vol. abs/1505.04597, 2015. [Online]. Available: http://arxiv.org/abs/1505.04597

[4] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," *CoRR*, vol. abs/1603.06937, 2016. [Online]. Available: http://arxiv.org/abs/1603.06937

[5] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[6] F. Huszr, "Dilated convolutions and kronecker factored convolutions." 2016. [Online]. Available: https://www.inference.vc/dilated-convolutions-and-kronecker-factorisation/

[7] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[8] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.