CutTheRoad – Aerial Image Road Segmentation

Utkarsh Bajpai, Remo Geissbühler, Thomas Lang, Robin Wiethüchter (Team fubar)[§]
Department of Computer Science, ETH Zurich, Switzerland

***Abstract*—Semantic segmentation is a classic problem in computer vision. In this paper, we investigate road segmentation, i.e. the task of detecting roads in satellite imagery. We use techniques of supervised deep learning which have become the de-facto standard in semantic segmentation. The aim of this paper is to compare different models (trained on different loss functions), to explore effects of data augmentation on the performance. Moreover, we introduce our novel solution. It is based on a Signed Distance Function representation of the data, which should implicitly take topological information into account. The Signed Distance representation encodes the distance to the closest pixel of the other class and is negative for pixels within the positive class. The novel solution shows comparable performance to some classic baselines, such as the CNN, U-net and its variations.**

I. Introduction

Road segmentation is a class of problem that involves segmentation of aerial images of a city. Each pixel of an aerial image is classified into two classes – 'road' or 'not road'. As such, it is a special case of semantic segmentation, where regions of images are to be distinguished and labelled. More precisely, we aim at learning a mapping $\mathbb{R}^{n\times m\times 3} \rightarrow \{0,1\}^{n\times m}$, where 1 denotes the road class and 0 denotes background. Road segmentation is increasingly useful in real life. Countless applications that use GPS navigation such as Google Maps, Uber etc. use aerial images of cities which have to be segmented to detect roads for proper navigation. Along with navigation, road segmentation is also useful in disaster relief where we can detect from aerial images which roads are blocked and where help is required.

In this paper, we present different approaches for image segmentation based on Neural Network architectures. We also compared the effect of different loss functions such as cross entropy loss, Lovasz loss, dice loss etc. We then present our novel approach which learns a Signed Distance Function (SDF) instead of the binary ground truth.

II. Related Work

Image segmentation, and consequently detection of roads in aerial images, has been done via various different techniques. Recently, most approaches tend to use a convolutional architecture. They range from simple Fully Convolutional Networks [1] to more complex models which are often based on the same ideas as FCNs.

[§]All authors contributed equally to this work.

A prominent example is the U-net [2], which initially was designed for biomedical image segmentation. It is of a similar structure like the DeconvNet, but intermediate outputs of the encoding convolutional layers are concatenated to the inputs of the decoding upsampling layers. These skip connections help to recover the fine details in the upsampling process.

While various different kinds of models have been used and sometimes also have been improved, i.e. 'U-Net++' [3] as a better version of U-Net, some of the focus was laid upon the prediction of shapes rather than just predicting in a pixel-wise manner. For example in [4] the authors tried to infuse the general knowledge of the shapes of roads and buildings to the predictions by using a Conditional Random Field model. Another approach can be found in [5] where an adapted U-Net model is trained such that it predicts a signed distance function, thus a shape, instead of a simple pixel-wise segmentation. Just as the original U-Net, it is designed for use in biomedical work, though.

III. Data Augmentation

The original data set to be used for training consists of only 100 images, 400×400 pixels each. This is not much per se, thus it is certainly a good idea to increase the amount of data by using a set of different data augmentation methods.

The first and most basic approach is to simply flip and rotate each image in 90 degree steps, multiplying the data space by a factor of 8. Additionally, any image can also be randomly rotated. We used angles between 10 and 80 degrees to prevent getting images that are too close to the original images. The rotated image is then cropped such that all pixels are valid data and resized back to the original size.

Finally, to adapt the data closer to the test data, we also included desaturated versions of the images. To make the model even more robust, one could also use blurred or grayscaled versions of the images, or add some noise to the data. As these techniques only introduced significant computational overhead without really increasing the performance, we decided to not apply them.

Even though the above described techniques increase the data space by a factor of up to 64 times, we further expanded the data space by introducing new images. The CITY-OSM data set [6] offers aerial images of different cities, especially of Chicago, from which the original test data is presumably taken. By rescaling and cutting the aerial images of Chicago to the same size and proportions we could get an additional

1828 images for training purposes. Of course, the previously described augmentation techniques were also applied to that new set of data, ultimately creating a solid amount of data to choose from.

For this paper we prepared five different data sets, each using a different subset of the methods described above.

1) **Original** – The original data set, as provided on Kaggle.
2) **Extended** – The original data set, together with the additional images of the CITY-OSM data set.
3) **Small Augmentation** – The extended data set with the flip and 90° rotations applied.
4) **Medium Augmentation** – Small augmentation with desaturation and one random rotation applied.
5) **Large Augmentation** – Medium augmentation with three random rotations instead.

IV. Losses

1) **Cross Entropy Loss** – Cross entropy loss is a distribution-based loss which measures the dissimilarity between two distributions. In our case, the data distribution is given by the training set, so the entropy ($H(q)$) is constant. Cross entropy (CE) is derived from Kullback-Leibler (KL) divergence. Thus, minimizing CE is equivalent to minimizing KL divergence.
2) **Dice Loss** – Dice Loss is a region-based loss functions aim to minimize the mismatch or maximize the overlap regions between ground truth and predicted segmentation. It directly optimizes the Dice coefficient which is the most commonly used segmentation evaluation metric.
3) **Focal Loss** – Focal loss adapts the standard CE to deal with extreme foreground-background class imbalance, where the loss assigned to well-classified examples is reduced.
4) **Lovasz Loss** – Lovasz loss function index[7] specifically designed to optimize the Jaccard [8]. Lovasz Softmax loss is used for multi class image segmentation. Jaccard set function has been shown to be submodular [9] and can be computed in polynomial time.

V. Baselines

A. Simple Patch-Wise Convolution

A very simple and basic approach to solve the problem is to simply run one convolution step with an adequate amount of trainable kernels with size 16×16, followed by another convolution layer with 1 kernel of size 1×1 and a sigmoid activation to map the output to a label prediction.

B. U-net model

The U-net in the original paper [2] was used as one of the baselines. It had multiple 2D Convolution layers with 32, 64, 128, 256, 512 filters. It uses 2 maxpool and ReLU function as its activation. For upsampling, the original U-net model uses filters of size 256, 128, 64 and 32 respectively in Conv2DTranspose layer.

C. MobileNetV2 U-net

For better results we use U-net based architectures. First, we use the U-net provided by tensorflow-examples using MobileNetV2 as encoder.

For more flexibility we use an implementation of U-net as proposed in the original paper [2].

D. Extended U-net model

We also extended the original U-net model by adding more layers. We added conv2D layer with 16 filters in the beginning of the model and made it deeper in the end by adding a layer with 1024 filters. For upsampling, relevant layers of filter size 16 and 1024 were also added in Conv2DTranspose. Figure1 shows the complete architecture of the model. This was done in the hope that more granularity in the U-net model would help in recognizing the streets with more accuracy.

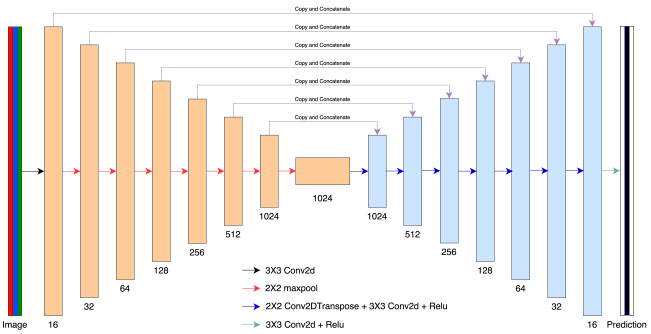


Figure 1: Deeper U-net architecture

E. Ensemble Method

Primarily in pursuit of a better generalisation score on Kaggle, we combined several different models and averaged their outputs, as suggested e.g. in [10]. Although this method seemed to produce visually better results, the Kaggle score did not improve using the ensemble learning.

VI. Novel Model

A. Motivation

One of the main challenges of this task is to use the prior information of what valid streets look like and teach the model accordingly. In particular, it is important that the predicted street has (at least) similar topological properties to the actual truth. For example, a connected part in the ground truth represents an uninterrupted street and should therefore be predicted as such.

Our previous attempts to adapt topological loss from [11] or to simply increase the weight on the edges in the loss function proved unfruitful, so we came up with a different approach.

Our novel solution is inspired by [5], which is based on a method from [12]. [5] uses a Signed Distance Function as ground truth to improve the shape of the predictions. First, let us formally define this transformation of the data.

B. Definitions

Definition 1. Let $y \in \{0,1\}^{n\times m}$ be a binary mask. We define the boundary mask of y as the output of the common Sobel edge detection [13], i.e. $\partial y := \mathrm{Sobel}(y)$. In particular, $(\partial y)_{ij} = 1$ iff (i,j) is on an edge between the positive and negative regions in y.

Definition 2. Consider a binary mask $y \in \{0,1\}^{n\times m}$. We define the *Signed Distance Function* of y component-wise as follows:

$$\mathrm{sdf}(y)_{ij} := \begin{cases} -d(y_{ij}, \partial y) & \text{if } y_{ij} = 1 \\ d(y_{ij}, \partial y) & \text{otherwise,} \end{cases} \tag{1}$$

where

$$d(y_{ij}, z) := \min_{k,l:\ z_{kl}=1} \|(i,j) - (k,l)\|. \tag{2}$$

In fact, we first extract the edges of the road to then construct the signed distance function as in [5]. It follows from the definitions above that the pixels $\{(i,j) \mid y_{ij} = 1\}$ correspond to the non-positive ones in the SDF, namely $\{(i,j) \mid \mathrm{sdf}(y)_{ij} \leq 0\}$.

C. Benefits

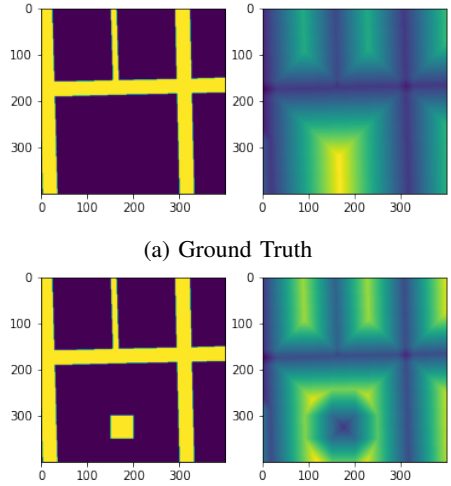
The advantage of using SDF data instead of the ground truth is that it implicitly encodes topological information. To illustrate this, consider the examples in Figs. 2a and 2b. Although a relatively small part is misclassified, which would only produce a relatively small usual loss, the SDF for this perturbation changes significantly.

D. Method

The preprocessing step that computes an SDF representation from given Ground Truth uses a Mutli-Source BFS algorithm. We represent the image as a graph and then start searching from all the edge pixels, i.e. from all (k,l) for which $(\partial y)_{kl} = 1$. Although is not the fastest solution ($O(nm)$), it only consumes $O(nm)$ memory, whereas precomputing the pairwise distance matrix to minimise over in (2) would require an impractical $O(n^2m^2)$ of memory.

To learn the SDF representation, we use the popular U-net introduced in [2]. Unfortunately, predicting the whole SDF (using a linear activation in the last layer) proved unsuccessful. The range of values is too large for the network to confidently learn it. In order to alleviate that issue, we instead used several different final activations – $\tanh$, softsign and 'fanh', a flatter version of tanh ($x \mapsto \tanh(\alpha x)$), which should retain more of the SDF structure. All of these activations scale the values into the interval $[-1,1]$.

With these activations, our model showed similar results as the baselines. With more fine-tuning, we are convinced that this method could outperform a simple U-net. As to which activation is preferable, cross-validation shows no clear winner. Quantitative results can be found in below.



(a) Ground Truth

(b) Erroneous Prediction

Figure 2: (a): The SDF for a ground truth image in the training dataset. (b): The SDF for a (fictional) wrong prediction. As we can see, the wrong part also influences the pixels around it and therefore produces a larger error. Indeed, the shown example has an accuracy of over 98.4% but the SDF has a MSE of more than 695. Best viewed in color.

E. Post-Processing

For the post-processing, we used a Fully-Connected Conditional Random Field (CRF) by [14] with 10 inference steps. Its idea is to minimize the Gibbs distribution induced by the energy

$$E(y) = \sum_{(i,j)} \psi_u(y_{ij}) + \frac{1}{2} \sum_{(i,j)\neq(k,l)} \psi_p(y_{ij}, y_{kl}), \tag{3}$$

where y is a mask of assigned classes, ψ_u denotes the unitary potential and ψ_p is the pairwise potential that accounts for inter-pixel interactions. For details, we refer the reader to [14]. In order to further assist connecting roads, we used a custom compatibility function that punished pixels classified as background but close to roads more. An example of a connection being added can be found in Fig. 3. The parameters for the kernel in [14] we used were:

$$\theta_\gamma = 2,\ \theta_\alpha = 40,\ \theta_\beta = 11,\ \mu = \begin{pmatrix} 0 & 50 \\ 10 & 0 \end{pmatrix}$$

VII. Comparative Study - Models and Losses

We started out by training our models under similar conditions (number of epochs, early-stopping) on the given 100 training images by using 5-fold cross-validation with 80 images to train and 20 to validate, due to the limited amount of training data. The results are shown in Table I.

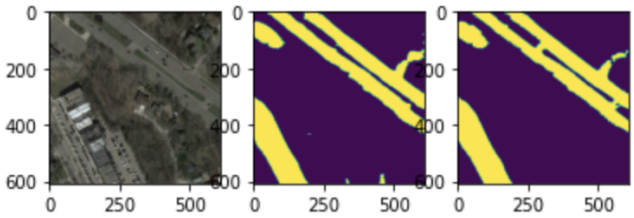


Figure 3: An example of the CRF connecting roads. The images shown are (from left to right): The test image, the raw output from the model, the output after post-processing.

The Kaggle-accuracy on the model resulting from the first fold of the cross validation is used as a test for the model. For our novel approach, using $\tanh$ or fanh (flat $\tanh$ with $\alpha = 0.1$) as well as using the U-net or deep U-net (dU-net) as underlying architecture shows no major differences in validation accuracy, mean-IoU and F1. While the Kaggle-test-accuracy is similar for all models, we do see significantly better results for our novel model on validation data.

	Kaggle	Validation		
	acc	acc	mIoU	F1
SDF-tanh U-net	0.8513	0.9814	0.9455	0.9547
SDF-fanh U-net	0.8495	0.9776	0.9351	0.9459
SDF-tanh dU-net	0.8351	0.9816	0.9463	0.9562
SDF-fanh dU-net	0.8568	0.9731	0.923	0.9355
dU-net cross entropy	0.8457	0.8497	0.427	0.4281
dU-net focal	0.8469	0.9255	0.4021	0.6855
dU-net dice	0.8126	0.934	0.8061	0.8292
dU-net Lovasz	0.8646	0.9069	0.6215	0.6996

Table I: 5-Fold Cross-validation on the original training data. Validation-scores are the average over the 5 folds. The Kaggle-Score has been created by using the resulting model of the first fold.

For better results, we add the Chicago Dataset as training data. We now use a fixed split of original images into training (70 images) and validation (30 images). The results in Table II clearly shows that our novel solution is an improvement over the Simple Patch Convolution baseline and the MobileNetV2 encoder U-net from tf-examples. The Kaggle-test-accuracy is similar among the other (deep) U-nets, but our novel SDF model again outperforms the other models on the validation on accuracy, mean-IoU and F1.

Results of different levels of augmentation on the deep U-net are shown in Table III. While there is a slight improvement in accuracy when using any kind of augmentation, the differences between the levels of augmentation seem to be much smaller. Since using the large augmentation (and even the medium one) significantly increase the training time, it is safe to conclude that it is not really worth using it. Nevertheless, when using a mixture of the augmentations where the original data is fully augmented as in the large augmentation and the Chicago data as in the small one, annotated as *special_aug*, we could get a small improvement, while still keeping the runtime at a reasonable level. Furthermore, when using CRF post-processing on the output of this model we achieve our highest score as can be seen in Table III.

	Kaggle	Validation		
	acc	acc	mIoU	F1
SDF-tanh dU-net	0.8645	0.9657	0.9049	0.9178
SDF-fanh dU-net	0.8688	0.9677	0.9068	0.9183
U-net cross entropy	0.8642	0.9634	0.4025	0.8965
U-net focal	0.8746	0.9597	0.3941	0.7507
U-net dice	0.8708	0.9478	0.8612	0.8805
U-net Lovasz	0.8474	0.7882	0.3941	n/a
dU-net cross entropy	0.8644	0.9614	0.4507	0.8783
dU-net focal	0.8655	0.959	0.3941	0.7397
dU-net dice	0.8823	0.9502	0.8631	0.8825
dU-net Lovasz	0.8633	0.8647	0.5859	0.5428
Simple Patch Convolution	0.7912	0.7674	0.3747	0.3749
MobileNetV2 tfe-U-net	0.8057			

Table II: Training on 70% of the original training data and the Chicago dataset. Validation on the remaining 30% of the original training data.

	Kaggle
dU-net dice small aug	0.88346
dU-net dice medium aug	0.87670
*dU-net dice special aug	0.88532
ensemble	0.88054
CRF post-processing of *	0.89508

Table III: Results of the deep U-net with different levels of augmentation as described in Section III. Also shown is the Ensemble method from Section V-E.

VIII. Summary, Future Work

To conclude, our novel SDF model combined with U-Net gives a decent accuracy. Compared to other models, it especially shows promising results when being trained with a small amount of data. Having a much higher mean-IoU score than the other models hints that SDF might actually be successful at its goal of predict shapes and their borders especially well. However, training SDF on large amounts of data is computationally challenging because of the resource heavy calculation of the distances.

In the future, the intuitive statement that predictions violating the ground truth's topology are punished more in SDF Loss could be made precise. Moreover, it could be an interesting approach to use the SDF representation in a loss function instead of a preprocessing step.

An interesting approach that we tried was to use a GAN-like [15] architecture: Instead of using a traditional loss function, we used a second network, the *Discriminator* to distinguish between actual ground truth images and bad network predictions. It seems that the discriminator was learning 'too rapidly' and that thus the 'generator' was unable to extract proper gradients for training. We mention this here because it still seems like an interesting approach.

REFERENCES

[1] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," 2014.

[2] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015.

[3] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," 2018.

[4] J. A. Montoya-Zegarra, J. D. Wegner, L. Ladický, and K. Schindler, "Semantic segmentation of aerial images in urban areas with class-specific higher-order cliques," ser. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, U. Stilla and C. Heipke, Eds., vol. II-3/W4. Göttingen: Copernicus, 2015, pp. 127 – 133, joint ISPRS workshops on Photogrammetric Image Analysis (PIA 2015) and High Resolution Earth Imaging for Geospatial Information (HRIGI 2015); Conference Location: Munich, Germany; Conference Date: March 25-27, 2015.

[5] S. M. M. R. al Arif, K. Knapp, and G. Slabaugh, *SPNet: Shape Prediction Using a Fully Convolutional Neural Network*, 09 2018, pp. 430–439.

[6] K. Pascal, W. J. Dirk, L. Aurelien, J. Martin, H. Thomas, and S. Konrad, "Learning Aerial Image Segmentation From Online Maps," Jul. 2017. [Online]. Available: https://doi.org/10.1109/TGRS.2017.2719738

[7] M. Berman, A. R. Triki, and M. B. Blaschko, "The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[8] A. Rakhlin, A. Davydow, and S. Nikolenko, "Land cover classification from satellite imagery with u-net and lovász-softmax loss," 06 2018, pp. 257–2574.

[9] J. Yu and M. Blaschko, "The lovász hinge: A novel convex surrogate for submodular losses," 2015.

[10] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

[11] X. Hu, L. Fuxin, D. Samaras, and C. Chen, "Topology-preserving deep image segmentation," 2019.

[12] A. Tsai, A. Yezzi, W. Wells, C. Tempany, D. Tucker, A. Fan, W. E. Grimson, and A. Willsky, "A shape-based approach to the segmentation of medical imagery using level sets," *IEEE Transactions on Medical Imaging*, vol. 22, no. 2, pp. 137–154, 2003.

[13] A. K. Jain, *Fundamentals of digital image processing*. Prentice-Hall, Inc., 1989.

[14] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," *CoRR*, vol. abs/1210.5644, 2012. [Online]. Available: http://arxiv.org/abs/1210.5644

[15] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014.