Road Segmentation: patch-wise vs pixel-wise models

Felix Sarnthein, Rares Constantin, Juraj Micko and Aashish Kumar Singh
Group: GeeseSquad
Department of Computer Science, ETH Zurich, Switzerland

***Abstract*—The task of image segmentation has been widely explored, and a range of algorithms find a wide domain of usage nowadays. Satellites produce a massive amount of high-quality images across all landscapes and potentially enable machine map generation. However, human labeled data is usually expensive to produce and labels might be noisy. Road Segmentation is the problem of dividing each image into regions that contain roads and regions that do not. We explore patch-wise and pixel-wise approaches to road segmentation in a small dataset with noisy labels.**

I. Introduction

This work aims to carry out the task of road segmentation via deep neural networks – exploring potential architectures and analyses of the results. Given a set of 400x400 RGB aerial images, we are asked to classify each 16x16 pixel patch as either containing road or background. A patch is defined as containing road if more than 25% of its pixels are marked as road in a pixel-wise segmentation mask. Therefore, the task can be seen as a pixel-wise classification or as a patch-wise regression which aims at predicting the ratio of road pixels in a patch. We explore patch-wise and pixel-wise approaches of this dual task definition, identify difficulties of the dataset and ultimately propose a strategy to solve them.

Section 2 of this paper discusses some related works published that aim at the task of road segmentation in aerial images. Section 3 describes the techniques involved in additional training data acquisition and data pre-processing. It further presents details of the model architectures implemented as a part of the solution. Section 4 discusses the results of our experiments. Section 5 discusses the experiments and results and further sheds light on potential future work.

II. Related Work

In this section, we are briefly going over some of the previous deep learning techniques for the task of road segmentation in aerial images.

One of the standard deep learning methods for tackling semantic segmentation problems is using a fully convolutional neural network (FCNN). This method involves learning the images' features in a compressed latent space, followed by an upsampling phase to get the predictions in the input space. Some of the papers that adopt the FCNN approach for road segmentation in satellite images are the ones by Henry et. al [1] and Buslaev et al. [2].

An extension of FCNN for semantic segmentation was proposed by Ronneberger et al. [3]. This approach revolves around an architecture called U-Net, which consists of "a contracting path to capture context and a symmetric expanding path that enables precise localization". Utilizing transpose convolutions in the decoder and skip connections to get contextual information from low-level layers, the U-Net network outperforms previous architectures by generating sharper segmentation masks. Even though the U-Net was initially designed for biomedical images, researchers such as Hou et al. [4] have managed to adapt the architecture for the task of road segmentation.

One advantage of Convolutional Networks is the weight-sharing which heavily regularizes the model, making it easier to train. However, this comes at the cost of highly local features: On the one hand, neighboring pixels will always have similar features, making it more difficult to produce rigid boundaries like in segmentation. On the other hand, the context covers only parts of the image, making it more difficult to model global structures like roads. The recently proposed Vision Transformer (ViT) [5] is a novel image model without convolutions. It was extended to work with semantic segmentation for example in the Segmenter acrchitecture [6] and STEGO [7].

III. Proposed Method

We identify four difficulties with the task at hand. Most prominently, the 144 images constitute a very *small dataset* which means that even for a simple neural network, the problem is heavily over-parameterized. Secondly, the dataset is *imbalanced*, with only about 14% of the pixels belonging to the road class. Furthermore, the data is *noisy* as the definition of roads is not very precise, and human annotation is not always coherent. Finally, although we are confronted with a segmentation task, the final evaluation is on patch-wise averages. This gives rise to a *dual task* definition: either a binary classification of pixels or a logistic regression of patch-wise aggregates. In this section, we evaluate five different components specifically designed to tackle the difficulties of the task: Baselines, Losses, Architectures, Augmentations, and Additional Dataset.

A. Baselines

As baselines, we consider two Convolutional Neuronal Networks (CNN) inspired by the original UNet architec-

ture [3] trained to minimize a binary cross-entropy (BCE) loss. Both consist of blocks with two padded 3x3 convolutions, ReLU activation functions, and batch normalization. An encoder consists of five such blocks. After every block, the resolution is halved with a 2x2 max-pooling operation while the number of channels doubles. Hence, an initial 3x400x400 input image is encoded layer-by-layer into a 1024x25x25 activation map.

A decoder is applied to this 25x25 activation map in the U-Net to produce a 400x400 output map. The decoder consists of the same blocks as the encoder but uses learnable transposed convolutions to upsample the resolution. Intermediate activation maps of the encoder are passed directly to the decoder, building so-called skip-connections.

We evaluate two baselines, one for each interpretation of the task description: a standard encoder-decoder UNet for pixel-wise classification and an encoder-only CNN for patch-wise logistic regression. The encoder-only architecture applies a 1x1 convolution to the intermediate activation map, which yields the 25x25 patch-wise output map.

B. Losses

Due to the class imbalance of 18% on patches (14% in pixels), a classifier always predicting background will already achieve 82% accuracy. For evaluation of such a task, the weighted F1 score is a more robust metric with respect to label imbalance since it combines the precision and recall of both the negative and positive classes by aggregating them according to their class weight. The standard BCE loss computes a pixel- or patch-wise error and aggregates it over all 625 patches and 8 images in a batch. Since the background class is over-represented in most batches, one would expect the gradient of the BCE loss to be skewed towards the negative class. An optimal solution would then trade-off precision for recall, which would result in a less optimal f1w score.

To counteract this behavior, we implement a balanced BCE (BBCE) loss, which rebalances the weight of the negative and positive samples. In particular, we compute the class imbalance for every patch as if the classes were balanced:

$$bbce = \frac{1}{2N^+N^-}\left(\sum_{i^+ \in N^+} N^- ce_{i^+} + \sum_{i^- \in N^-} N^+ ce_{i^-}\right)$$

Another problem of the dataset is label noise, which adds wrong optimization pressure due to incoherent labels: something might be classified as a road but is not, or vice-versa. The Focal loss [8] addresses this issue for binary classification by down-weighing samples far away from the decision boundary using a discount factor γ. We extend the focal loss to the logistic regression setting for patch-wise predictions as shown in Figure 1.

C. Architectures

Using pre-trained models is a different approach to handling small datasets with noisy and imbalanced data. The

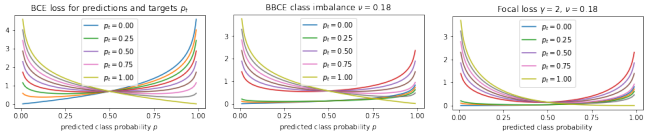


Figure 1: Comparing BCE, BBCE, and Focal loss for patch-wise logistic regression. The BBCE applies less optimization pressure to samples with $p_t < 0.25$. The Focal loss down-weighs predictions far away from p_t. Best viewed on screen.

gold standard for pre-trained CNNs are Residual Networks (ResNets) [9] pre-trained on ImageNet [10]. Like the U-Net encoder, ResNets consists of five blocks, but each block has a residual connection and possibly more convolutions. The timm library [11] provides implementations and Imagenet pre-trained ResNets, which we modify for our needs. In particular, we use a ResNet-D Variant [12] which features a deep stem, i.e., a block of 3x3 convolutions instead one 7x7 convolution is applied to the input image, and change the initial stride from 2 to 1. We motivate this by the fact that – as opposed to object detection – in segmentation tasks, knowledge about high-resolution features is important. As an additional benefit, the modified ResNet-D encoder now produces 25x25 feature maps, which we directly use for patch-wise predictions. For pixel-wise predictions, we implement a ResNet decoder as an inverted ResNet. Aiming to follow the encoder structure as closely as possible, we insert transposed convolutions between the residual blocks and add skip-connections from layers of the same stride similar to the baseline U-Nets. We compare the sizes of all evaluated architectures in Table I.

In vision transformers, an image is usually embedded into patches of 16x16 pixels and transformed through a multi-layer attention mechanism, making it very suitable for our patch-wise regression task. This theoretically allows for global context and pattern-specific activation maps. Although ViTs are not size-invariant by default, we find that the self-supervised training technique DINO [13] also works for a size of 400. We evaluate the small version of the ImageNet pre-trained DINOViT for patch-wise logistic regression called dinovits, similar to STEGO [7].

size	patch-wise	pixel-wise
unet	$18.8 \cdot 10^6$	$31.0 \cdot 10^6$
resnet18d	$11.2 \cdot 10^6$	$16,3 \cdot 10^6$
resnet50d	$23,5 \cdot 10^6$	-
dinovits	$21,6 \cdot 10^6$	-

Table I: Size of evaluated architectures in comparison.

D. Augmentations

To avoid overfitting due to the small dataset size and improve the model's robustness, we have adapted a training strategy based on augmentations, similar to the one presented by Ronneberger et al. [3]. Thus, we chose to preprocess images by augmenting them dynamically during training using the Albumentations library [14]. **??** illustrates the transformations an image undergoes during our augmentation process. Each transformation has an independent probability of being applied (e.g., 0.5) and falls into one of the three different categories of transformations we defined: color mutations, affine transformations, and distortions.

Color mutations – The role of the color mutations is to expand the color distributions of the training images so that they cover the properties of the images used for validation and testing. Hence, we randomly apply different filters and modify certain color attributes of the training images, such as: hue, saturation, brightness, and contrast change, as well as Gaussian noise addition. All of these mutations' ranges were carefully selected so that the images' results still look natural.

Affine transformations – By applying affine transformations to the input images, the model learns to be invariant to them, increasing overall performance. We use horizontal and vertical flips, transposes, and 90 ± 10 degrees rotations. Not utilizing all degree values when rotating the images can be justified by the big empty spots that appear in every corner in the case of the excluded degree values.

Distortions – To further improve the robustness of the model with respect to small deformations, we apply different types of subtle distortions to the images, such as: grid distortion, optical distortion, and elastic transform. To avoid generating images with unreal features, a maximum of one distortion is used at a time when augmenting an image. We apply a center crop followed by a resizing to the original size to remove potential artifacts at the border.

E. Additional Dataset

While augmentations increase the variability in input images, they do not introduce more variability in the structure of roads. To further improve generalization, we create a much larger dataset by gathering additional satellite images containing roads using the Google Maps API [15].

Utilizing the Maps Static API service, we selected areas from multiple US cities, which, empirically, have a similar appearance to the provided dataset. Precisely, we chose similar areas of the same cities as a team that previously participated in this competition in 2020 [16]. The numbers of images for each city were chosen proportionally to their corresponding area, resulting in 12,000 image-label pairs in total, covering a total area of $7,335\,\mathrm{km}^2$. The exact distribution of images in this dataset is as follows: Los Angeles (3,271 images), Chicago (4,067 images), Houston (1,735 images), Phoenix (2,470 images), Philadelphia (235 images), San Francisco (112 images) and Boston (110 images). Satellite images were downloaded with a constant zoom level and by specifying i.i.d. generated coordinates of centers within each region. To generate the image masks, we set up a custom map style in the Google Cloud Console, configuring the visibility of map features (i.e., roads, landscape, pathways, road widths, etc.) on a black background.

Normalization Given that the different cities represent different data distributions, we apply per-city data normalization as a pre-processing step. For every city, we once compute the channel-wise statistics of all pixels marked as roads. When loading the images, we subtract the channel-wise mean and divide it by the channel-wise standard deviation. This means that the distributions of road pixels are now close to the standard distribution, while the overall statistics are more irregular. We refer to Figure 3 and Figure 4 in the Appendix for more details.

IV. Experiments and Results

We compare all four different strategies to our two Baselines. In all cases, we train on a training split of 129 images using AdamW [17] with default parameters and a batch size of 8. We evaluate on a validation split of 15 carefully selected images, measuring the weighted F1 accuracy (F1W) score. We train for 100 epochs and apply early-stopping on the highest validation F1W score for patch-wise predictions. In the special case of augmentations, we train for 250 epochs. When training on the Google Maps data, we validate on all 144 original training images.

A. Losses

We observe that the balanced losses are much more unstable to train than the standard BCE. While initially maintaining higher recall and lower precision, they converge towards the baseline values but with lower overall performance. This is also reflected in the evaluation in Table II.

	patch-wise		pixel-wise	
experiment	**F1W**	**epoch**	**F1W**	**epoch**
BCE	**0.88**	97	**0.883**	94
BBCE	0.869	67	0.865	99
Focal	0.669	47	0.84	98

Table II: Different losses compared to the two baselines.

B. Architectures

In Table III, we observe that the ResNet18-D architectures perform similar to the baselines when trained from scratch, although they have fewer parameters. Furthermore, we observe the ImageNet pre-trained models increase the performance by 1-2%. We evaluate the feature quality of pre-trained models by training the decoder only and observe that those features achieve competitive performance, especially for pixel-wise predictions where the decoder is more complex. Unfortunately we were only able to evaluate the

superior performance of ViT features. Training ViTs seems to require a complex optimization procedure that we could not reproduce.

	patch-wise		pixel-wise	
experiment	**F1W**	**epoch**	**F1W**	**epoch**
U-Net	0.88	97	0.883	94
ResNet18d	0.882	29	0.884	93
ResNet18d*	0.727	99	0.878	67
ResNet18d**	0.892	53	**0.893**	18
ResNet50d	0.879	98	-	-
ResNet50d*	0.741	100	-	-
ResNet50d**	**0.894**	87	-	-
DinoViTs*	0.790	100	-	-

Table III: Different architectures compared to the two baselines. *Imagenet pre-trained backbone: training decoder only **Imagenet pre-trained backbone: fine-tuning end-to-end

C. Augmentations

In Table IV we observe that augmentations increase the generalization abilities by 1-2% and that the group of affine augmentations contributes the most.

	patch-wise		pixel-wise	
augmentations	**F1W**	**epoch**	**F1W**	**epoch**
without	0.886	225	0.884	124
color&noise	0.882	412	0.871	136
affine	**0.903**	271	**0.901**	357
distortions	0.897	228	0.887	126
all	0.902	481	**0.901**	355

Table IV: Different augmentations compared to the baseline.

D. Additional Dataset

In Table V observe that training on the large Google Maps dataset increases performance by 2%, of which 1% is due to the sophisticated normalization strategy.

	patch-wise		pixel-wise	
training set	**F1W**	**epoch**	**F1W**	**epoch**
original	0.88	97	0.883	94
gmaps	0.889	11	**0.894**	8
gmaps*	**0.899**	7	**0.894**	8

Table V: Models' performance when using the additional dataset for training compared to the one that uses the original dataset. *Every dataset is normalized for road pixels to match the standard normal distribution.

E. Combining Strategies

After evaluating the different strategies, select a few combinations to, as shown in Table VI. The model used for the final Kaggle submission is a voting ensemble of all these models. Using this method, we achieved a final score of **0.93524** on the platform.

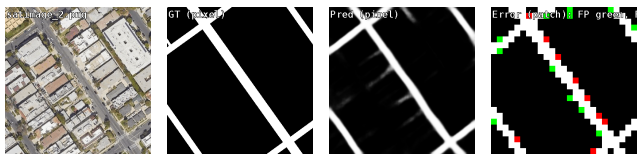
	patch-wise		pixel-wise	
model	**F1W**	**epoch**	**F1W**	**epoch**
baseline	0.88	97	0.883	94
U-Net	0.921	87+3	0.917	60 + 27
ResNet18d	**0.927**	79 + 5	0.918	60 + 4
ResNet50d	-	-	**0.922**	80 + 12

Table VI: The scores of models combining the best strategies. All models were trained on the Google Maps images and fine-tuned on the original dataset.

V. Discussion and Future Work

In our quest to solve the task of training a segmentation model on a *small* and *imbalanced* dataset with *noisy* labels for *patch-wise predictions*, we find that most of our components help to conquer these problems. Most importantly, to solve the dual task, we observe that patch-wise models are competitive with pixel-wise models, but the former perform slightly better in an optimal setting. While specialized losses did not achieve to overcome imbalanced and noisy labels, pre-trained models proved to do so. Using augmentations, an additional dataset with pre-processing and fine-tuning reduced the burden of a small dataset. Finally, creating a voting ensemble of best-performing strategies resulted in a satisfactory Kaggle score.

Our results show the superior feature quality of vision transformers, which suggests looking into these models for patch-wise predictions. For future work, we consider using self-supervised learning to avoid noisy labels. Furthermore, an even larger dataset could be created to get more variability in the structure of roads. Finally, we consider developing a more sophisticated method to produce patch-wise activations from pixel-embeddings instead of simple averaging. For example, an attention mechanism could bridge the gap between the pixel- and patch-wise prediction tasks.



(a) Source image (b) Label (c) Prediction (d) Error type

Figure 2: Qualitative evaluation of a U-Net model, demonstrating the potential for improvement by ViTs: false positives (green), false negatives (red).

VI. Conclusion

Producing accurate segmentation masks for roads in satellite images proved challenging, especially with access to few training examples. Using a multitude of techniques, such as generating a larger dataset for training, creating augmentations, and using transfer learning, we surpassed the performance of the selected baseline models, thus achieving satisfactory results.

REFERENCES

[1] C. Henry, S. M. Azimi, and N. Merkle, "Road segmentation in sar satellite images with deep fully convolutional neural networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 12, pp. 1867–1871, 2018.

[2] A. Buslaev, S. Seferbekov, V. Iglovikov, and A. Shvets, "Fully convolutional network for automatic road extraction from satellite imagery," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

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

[4] Y. Hou, Z. Liu, T. Zhang, and Y. Li, "C-unet: Complement unet for remote sensing road extraction," *Sensors*, vol. 21, no. 6, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/6/2153

[5] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *CoRR*, vol. abs/2010.11929, 2020. [Online]. Available: https://arxiv.org/abs/2010.11929

[6] R. Strudel, R. Garcia, I. Laptev, and C. Schmid, "Segmenter: Transformer for semantic segmentation," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 7242–7252.

[7] M. Hamilton, Z. Zhang, B. Hariharan, N. Snavely, and W. T. Freeman, "Unsupervised semantic segmentation by distilling feature correspondences," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=SaKO6z6Hl0c

[8] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, pp. 318–327, 8 2017. [Online]. Available: https://arxiv.org/abs/1708.02002v2

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[10] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *CoRR*, vol. abs/1409.0575, 2014. [Online]. Available: http://arxiv.org/abs/1409.0575

[11] R. Wightman, "Pytorch image models," https://github.com/rwightman/pytorch-image-models, 2019.

[12] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, "Bag of tricks for image classification with convolutional neural networks," *CoRR*, vol. abs/1812.01187, 2018. [Online]. Available: http://arxiv.org/abs/1812.01187

[13] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," *CoRR*, vol. abs/2104.14294, 2021. [Online]. Available: https://arxiv.org/abs/2104.14294

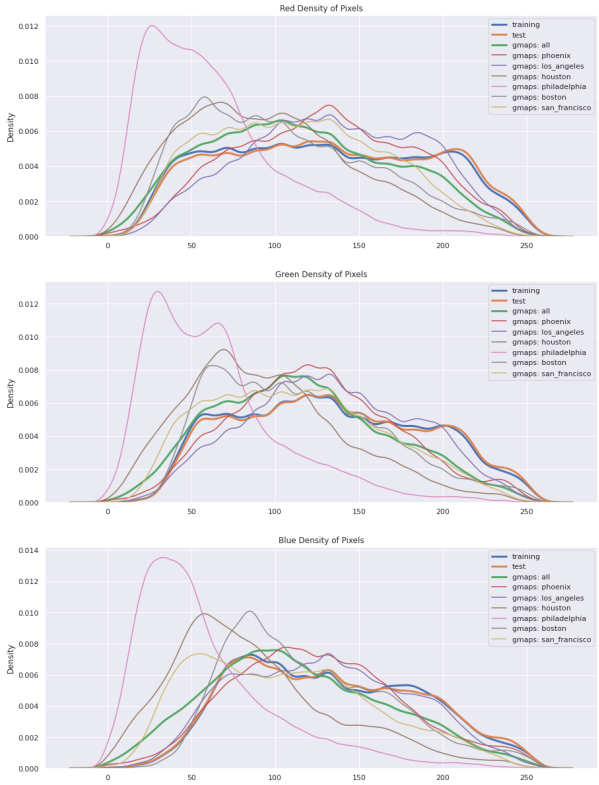
[14] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: Fast and flexible image augmentations," *Information*, vol. 11, no. 2, 2020. [Online]. Available: https://www.mdpi.com/2078-2489/11/2/125

[15] G. Svennerberg, *Beginning Google Maps API 3*, 2nd ed. USA: Apress, 2010.

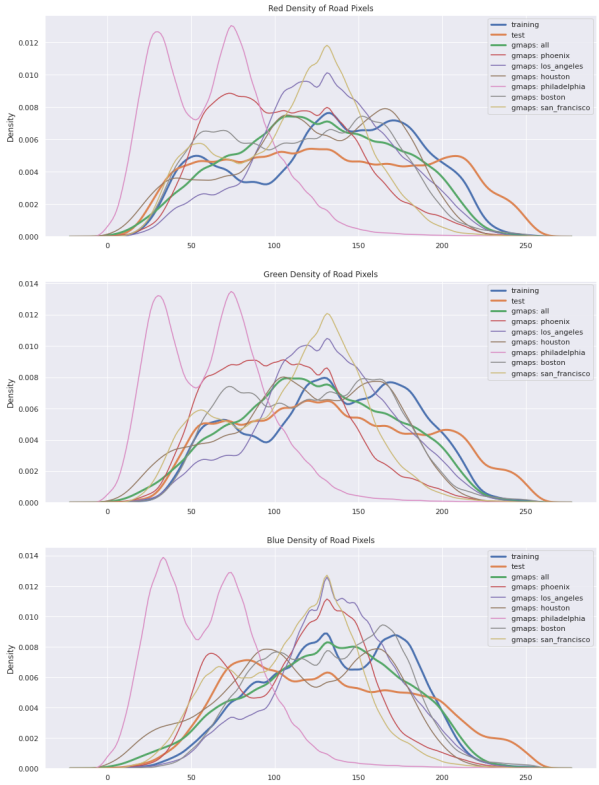
[16] D. Chiappalupi, E. Iannucci, S. Piazzetta, and G. Lain, "danich8/road-segmentation-eth-cil-2020: Computational intelligence lab project, eth zürich - spring semester 2020." [Online]. Available: https://github.com/daniCh8/road-segmentation-eth-cil-2020

[17] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2017. [Online]. Available: https://arxiv.org/abs/1711.05101

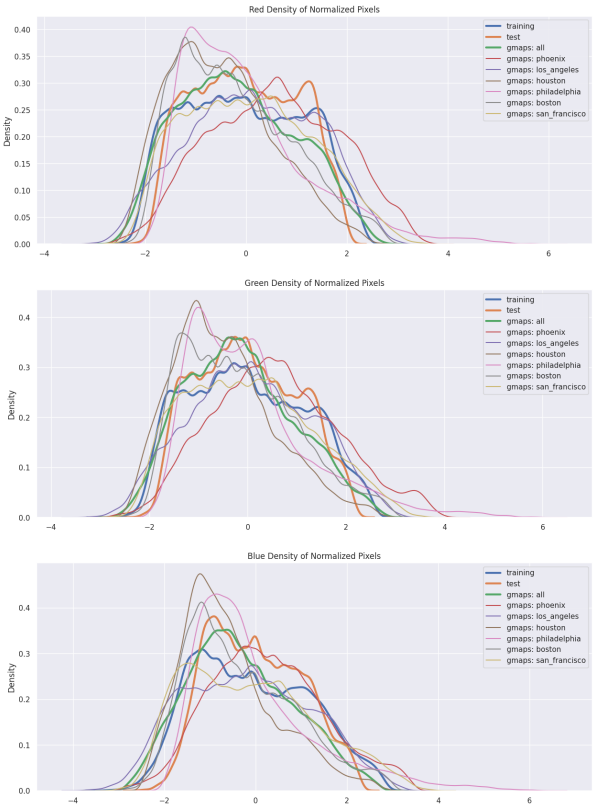
VII. Appendix



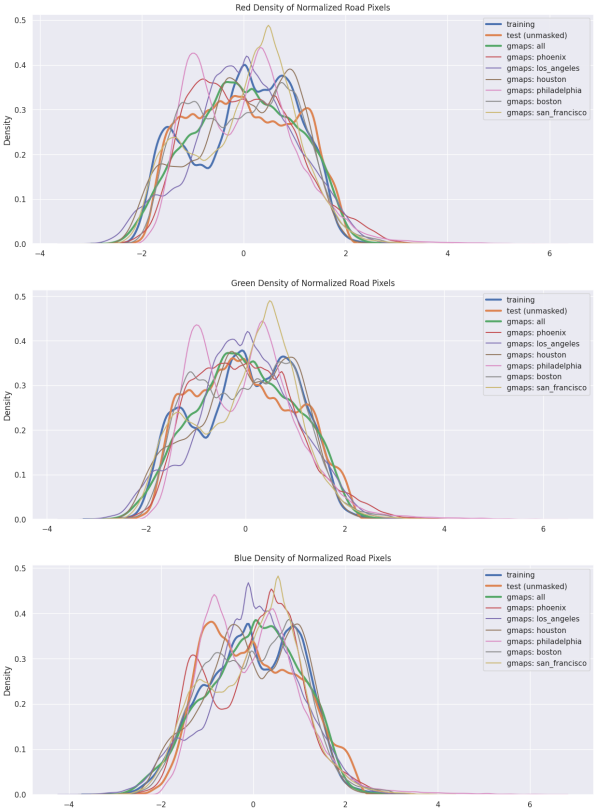
(a) densities all pixels



(b) densities road pixels



(c) normalized densities all pixels



(d) normalized densities road pixels

Figure 3: Channel-wise densities of road vs all pixels before and after normalization. Note that normalization according to road statistics causes non-road pixels to be irregularly distributed and potentially easier to separate.

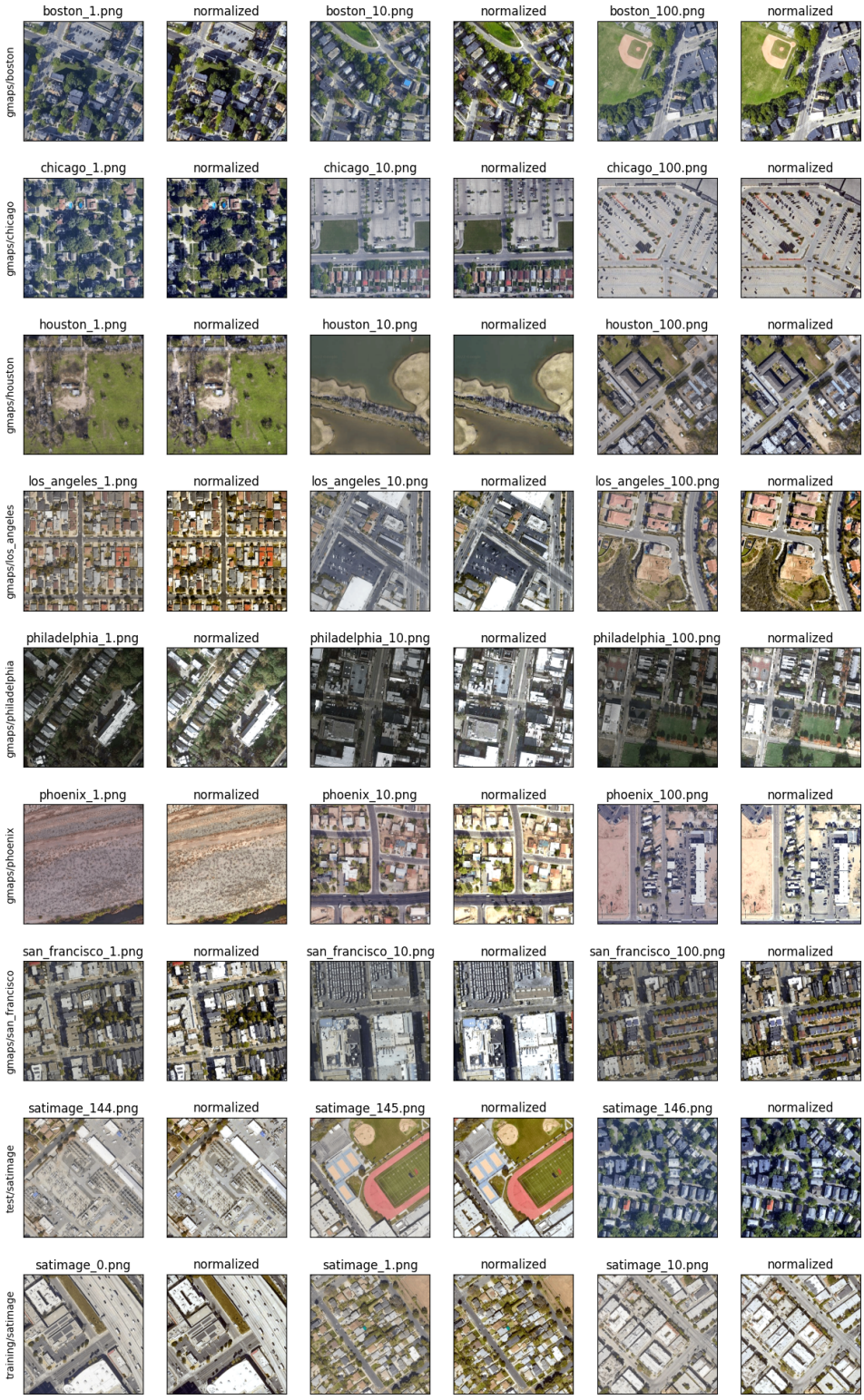


Figure 4: Examples from all cities before and after normalization. Note that road pixels have more similar colors across cities.

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 2022 project: Road Segmentation

Authored by (in block letters):

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

Name(s):	**First name(s):**
Singh	Aashish Kumar
Micko	Juraj
Sarnthein	Felix
Constantin	Rares

With my signature I confirm that

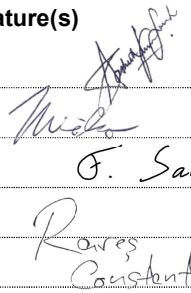
- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

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

Place, date

Zurich, 31.7.2022

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.