Few-Shot Road Segmentation

*Nicolas Zucchet, *Tristan Cinquin, *Constantin Le Clei, *Rafael Bischof
**Group: Bandoleros, Department of Computer Science, ETH Zurich, Switzerland*
{*nzucchet, tcinquin, cleclei, rabischof*}*@student.ethz.ch*

***Abstract*—Manually labeled data for Road Segmentation can be costly and hard to come by. It is therefore crucial to make the best use of the available samples. We seek to achieve this by training a U-Net in a few-shot learning fashion with the rationale that this allows us to efficiently use datasets from different parts of the world while disposing of only a limited number of samples from our task of interest.**

Our contribution is two-fold. We create the DiverCity dataset, that contains several tasks, each one corresponding to a specific region of the world. It allows us to train the meta-learning algorithm REPTILE.

We then conduct several experiments that show the superiority of meta-learning over transfer-learning in several use-cases, especially when data is rare.

I. Introduction

With 2666 active satellites orbiting around Earth [1], among which 278 are specialized on optical imagery for civil or commercial purposes, a staggering amount of footage is taken every day. Aerial road segmentation systems can exploit this abundance of data to map and monitor the 64 million kilometres of roads [2].

In an era marking the emergence of various autonomous systems like self-driving cars and drones, exploiting satellite imagery can provide valuable information for orientation and path-finding. In case of recent changes to the road networks, maintenance work, bypasses or other hazards, a self-driving car could, for instance, consult up-to-date satellite images to identify alternative routes. Publicly available datasets [3, 4] provide numerous high quality labelled images and make it easier to compare different segmentation systems.

In the following, we seek to segment images from a specific region of the world while disposing of only a very small number of similar labeled images. To do so, we create a diverse dataset and use a training algorithm from the few-shot learning approach – considered as one of the remaining challenges in image segmentation [5]. We identify two major issues when coping with segmentation of roads:

- Geography and culture: streets follow a different pattern in North American suburbs than they do in the labyrinthine old town of Rome. Learning a globally valid model of roads is therefore a highly complex task;
- Labeling policy: different datasets may adhere to varying definitions of "roads". In our case, this is especially true for parking lots: do they (and which parts of them) count as roads?

A model trained with few-shot learning does not have to generalize to all possible shapes and colors of roads, nor does it have to cope with possible contradictions in the dataset. It will much rather gain a general understanding of the road network, while keeping enough flexibility to quickly adapt to a given task.

II. Few-shot learning

When data, here precise road maps for area with potentially specific roads or landscape, is hard to acquire it is crucial to get the most out of each sample. Several approaches exist to tackle this problem [6]; we use the meta-learning one. It consists in finding a model which can quickly adapt to a new task when provided with few training examples and training steps.

We define a task $\mathcal{T}_i$ by its probability distribution $\mathcal{D}_i$ and we note $\mathcal{T}$ the uniform distribution over all tasks. In the road segmentation context, a task will consist in detecting roads in a specific region on images sharing the same characteristics (zoom, resolution...).

Formally, given a parametric model f_θ, a training algorithm $\mathcal{A}$ that takes an initialization weight and some samples (x, y) and outputs the learned weight and a loss function $\mathcal{L}$, we want to find parameter θ such that

$$\theta \in \operatorname{argmin} \mathbb{E}_{\mathcal{T}_i \sim \mathcal{T}} \, \mathbb{E}_{(x,y),(x',y') \sim \mathcal{D}_i} \, \mathcal{L}(f_{\theta'}(x), y) \qquad (1)$$

where $\theta' = \mathcal{A}(x', y'; \theta)$.

To find an approximate solution of (1), we use the REPTILE algorithm [7], a first-order version of MAML [8]. For each *meta-iteration*, we randomly sample one task. Then, we train the network on data from this task using some variant of gradient descent, starting from W and finishing at W'. We will refer to the number of training epochs on a single task as the number of *inner epochs*. The last step of the meta-iteration is the update of the current estimate W of the weights:

$$W \leftarrow W + \alpha(W' - W)$$

with α a parameter that linearly decreases from its initial value to 0 during meta-training.

III. Model and Methods

A. Datasets

1) CIL dataset: The provided dataset consists of 100 training images and 94 testing images. We will hereafter be referencing it by the name *CIL dataset*.

Figure 1: Geographical distribution of the tasks from our DiverCity dataset.

2) DiverCity dataset: To match real-world conditions, we create a dataset of road segmentation obtained using Google Maps. We choose 40 geographical regions (Figure 1) across the globe where descent quality satellite imagery is available and which greatly differs in terms of landscape, road and weather conditions. For each region, we sample 100 images distributed along some arbitrarily chosen grid. Some examples can be found on Figure 2. Following few-shot learning denomination, we will refer to a specific region as one task.

This data generation process is extremely adaptable: one can easily get images from a large set of regions. However, the labels produced are not as accurate as if they had been obtained manually and not always available in remote areas.

B. U-Net

We use a U-Net architecture [9]. Thanks to its low number of parameters and its fully-convolutional nature – thus insensitive to input size – this model is one of the most versatile and easy to use. It has been reported to produce good segmentation in various contexts [10–12]. Many recent architectures are encoder-decoder based (e.g. LinkNet [13], SegNet [14], SDN [15]), as U-Net, which leads us to believe that the same methods can be applied to these architectures easily.

C. Dice-penalized cross entropy loss

The proportion of roads is low compared to the rest, so the segmentation is unbalanced. To take this into account, we add to the binary cross entropy loss (Eq. 2) a Dice coefficient penalty (Eq. 3), inspired from the one introduced in [16].

$$\mathcal{L}_{CE}(\hat{p}, y) = \sum_i y_i \log(\hat{p}_i) + (1 - y_i) \log(1 - \hat{p}_i) \quad (2)$$

$$\mathcal{L}_D(\hat{p}, y) = 1 - 2\frac{\epsilon + \sum_i y_i \hat{p}_i}{\epsilon + \sum_i y_i + \hat{p}_i} \quad (3)$$

$$\mathcal{L}(\hat{p}, y) = \mathcal{L}_{CE}(\hat{p}, y) + \mathcal{L}_D(\hat{p}, y) \quad (4)$$

where ϵ is a smoothing parameter and $\hat{p}$ the probability map at the output of the network, ranging between 0 and 1.

As roads sometimes consist in small structures that are difficult to detect accurately, we added a label-dependent pixel-wise weight on the loss following Kervadec et al [17]. It consists in giving more importance to non-road pixels close to roads.

IV. Training procedure

A. Training

Following Sun et al. [18] we first train our UNet on the entire DiverCity dataset by merging all tasks into one big set (Step 1). From this pre-trained model we then perform a number of meta-iterations on the individual tasks of the DiverCity dataset (Step 2) before fine-tuning on CIL (Step 3). Note that zero meta-iterations before fine-tuning essentially means transfer learning. We argue that performing meta-learning on the pre-trained model brings it to an initialization that allows for faster convergence on new tasks, as the underlying model is already in a good performance region.

This fast, initial convergence means that a model's final state heavily depends on the first few optimization steps in which the tasks are sampled, bringing some stochasticity when searching the feature-space. Nichol et al. [7] stated that the amount of stochasticity can be controlled to some extent by adjusting the number of inner epochs per meta-iteration. In section V-A, we experimentally find the best working values.

B. Post-processing

Inspired by the work of Mosinska1 and al. in [19], we increase the accuracy of the predictions obtained by our model by recursive refinement. To this purpose, we build a post-processing UNet and train it with the following procedure.

Algorithm 1: Iterative Refinement

```
Data: D = {(image, label)}
Let seg_model be the UNet used for image segmentation;
Let refine_model be the UNet used for iterative
  refinement;
Let refine_iterations = 3;
while refine_model not converged do
    for (image, label) in D do
        loss = 0;
        mask = seg_model(image);
        Add random noise to mask;
        input = mask ⊕ image;
        for i ← 1 To refine_iterations do
            output = refine_model(input);
            loss += i * loss_function(input, output);
            input = σ(output) ⊕ image;
        end
        loss /=
          0.5*refine_iterations*(refine_iterations+1);

        Adam optimization step on loss and
          back-propagation;
    end
end
```

Figure 2: Image and label samples from our DiverCity dataset.

where $\oplus$ denotes tensor concatenation along the channel dimension. To teach the model to recover from the erroneous "blob" predictions, we add random sized white circles and rectangles to the masks. Furthermore, we weight the loss to give more importance to the final iterations. We train the refinement model for 400 epochs with a learning rate of 0.01 and a decay schedule.

V. Experiments

A. Hyperparameters of meta-training

In this experiment, we aim to find the right number of inner epochs and meta iterations.

On Figure 3, we plot the cosine similarity between weights obtained after given meta iterations, for two different numbers of inner epoch. With one inner epoch, similarity between successive weights is quite high, and final weights end up not being so different from the initial weights : this shows that one epoch is not sufficient for the task to have an impact on the global weights, since the inner parameters are not trained enough. On the other hand, at four inner epochs, the model seems to explore more of the parameter space, especially at the beginning of training. We also use this finding as a measure of convergence to choose a sensible number of meta-iterations (200 meta-iterations and 4 inner epochs).

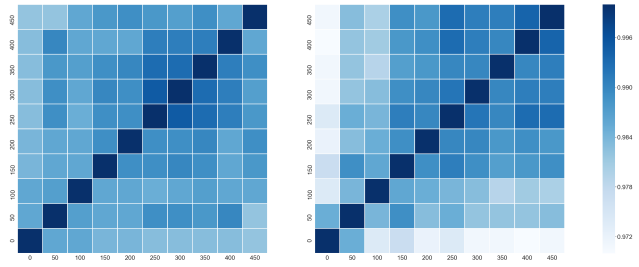


Figure 3: Cosine similarity of model weights upon fine-tuning after different number of iterations. Left: 1 inner epoch, Right: 4 inner epochs.

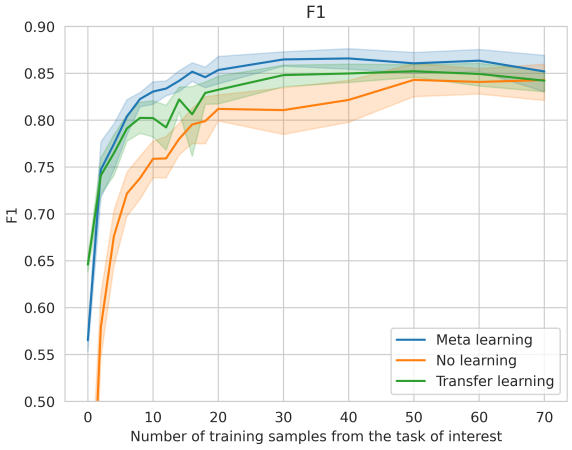


Figure 4: F1 score after fine-tuning different models on a given number of samples from CIL dataset.

B. Comparison of different approaches in the context of few-shot learning

We compare F1-scores after fine-tuning on the CIL dataset for classic transfer learning (TL) and meta learning (ML) procedures, as a function of the number of CIL samples used to fine-tune. We use a model where the first layers were trained on ImageNet (No Learning, NL) as a baseline. Note that we don't apply post-processing here.

The sample size ranges from 1 to 70, with a granularity of 2 below 20 and of 10 above. For each sample size and each method, we ran 10 trainings (35 epochs) with randomly chosen training and validation samples. Results are displayed in Figure 4. We observe that ML outperforms TL when trained with more than 2 samples. This suggests that ML is more suited for extracting task-specific information on a small number of samples. The lower performance of ML on very-low regime (0-1 samples) is a consequence of meta-learning finding a good initialization from which, in a few gradient steps, it is easy to reach good performance

Figure 5: Segmentations images obtained, before post-processing. Top row : Images with label overlays. Bottom row : Images with prediction overlays

on all tasks: for such low regime, we don't move from initialization.

C. Computation time vs. segmentation quality

We compare the performance and execution time of the three approaches presented above, using 70 samples from the CIL dataset and report the results in Table I. Execution time were obtained on Leonhard.

The meta-learning approach outperforms the other two, but with a margin that is not necessarily worth the extra computational time. Interestingly, the extra data samples used to train TL don't help to increase the quality.

Table I: Comparison of No Learning (NL), Transfer Learning (TL) and Meta Learning (ML), trained on 70 samples, in terms of execution time and F1 score.

	Step 1	Step 2	Step 3		
Time	2h00	4h00	0h10	**Total time**	**F1**
NL			✓	0h10	0.843
TL	✓		✓	2h10	0.842
ML	✓	✓	✓	6h10	0.852

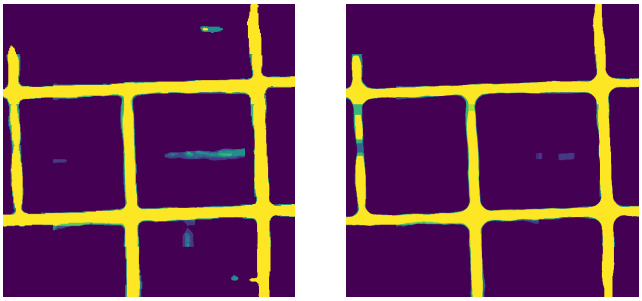
D. Visual inspection of the segmentations

Figure 5 shows the segmentation produced on the validation set before post-processing, while Figure 6 exemplifies how post-processing improves the result. The latter step seems to smoothen the roads and give an overall more homogeneous segmentation.

VI. Discussion

We have presented a way to do Road Segmentation given only small-sized (100 samples) datasets, using a few-shot meta-learning approach, and some post-processing. We have shown evidence of the relevance of such a technique compared to a classic transfer learning approach, both in terms of convergence stability and performance. Even though the results are satisfactory, the experiments show that such a procedure would be more suited for tasks with a smaller number of samples. Another technical limit could also be the use of the U-Net architecture. It can be replaced by a more recent state-of-the art model, but we chose not to focus on this aspect.

We would like to emphasize that DiverCity is a new dataset specifically tailored for meta-learning, where most of the work in this area is currenlty benchmarked on Omniglot, a dataset for meta-symbol recognition. It can therefore be used to train prospective meta-learning models. However, even though we included satellite imagery from locations all around the world, it remains biased towards cities in the global north-west. Out of fear of complicating the task, we visually inspected the datasets and decided on which ones to keep based more on resolution of the images and quality of the labels, rather than on geographical location.



(a) Original segmentation (b) Post-processed

Figure 6: Contribution of post-processing

REFERENCES

[1] "Satellite Database | Union of Concerned Scientists," library Catalog: www.ucsusa.org. [Online]. Available: https://www.ucsusa.org/resources/satellite-database

[2] "List of countries by road network size," Jul. 2020. [Online]. Available: https://en.wikipedia.org/w/index.php?title=List_of_countries_by_road_network_size

[3] "SpaceNet Roads Dataset." [Online]. Available: https://spacenetchallenge.github.io/datasets/spacenetRoads-summary.html

[4] S. Wang, M. Bai, G. Mattyus, H. Chu, W. Luo, B. Yang, J. Liang, J. Cheverie, S. Fidler, and R. Urtasun, "TorontoCity: Seeing the World with a Million Eyes," in *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, Oct. 2017, pp. 3028–3036.

[5] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image Segmentation Using Deep Learning: A Survey," Apr. 2020, arXiv: 2001.05566.

[6] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM Comput. Surv.*, vol. 53, no. 3, Jun. 2020.

[7] A. Nichol, J. Achiam, and J. Schulman, "On First-Order Meta-Learning Algorithms," Oct. 2018, arXiv: 1803.02999.

[8] C. Finn, P. Abbeel, and S. Levine, "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks," Jul. 2017, arXiv: 1703.03400.

[9] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," May 2015, arXiv: 1505.04597.

[10] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Insights into Imaging*, vol. 9, no. 4, pp. 611–629, Aug. 2018.

[11] T. Falk, D. Mai, R. Bensch, O. Çiçek, A. Abdulkadir, Y. Marrakchi, A. Böhm, J. Deubner, Z. Jäckel, K. Seiwald, A. Dovzhenko, O. Tietz, C. Dal Bosco, S. Walsh, D. Saltukoglu, T. L. Tay, M. Prinz, K. Palme, M. Simons, I. Diester, T. Brox, and O. Ronneberger, "U-Net: deep learning for cell counting, detection, and morphometry," *Nature Methods*, vol. 16, no. 1, pp. 67–70, Jan. 2019, number: 1 Publisher: Nature Publishing Group.

[12] V. Iglovikov and A. Shvets, "TernausNet: U-Net with VGG11 Encoder Pre-Trained on ImageNet for Image Segmentation," Jan. 2018, arXiv: 1801.05746.

[13] A. Chaurasia and E. Culurciello, "LinkNet: Exploiting encoder representations for efficient semantic segmentation," in *2017 IEEE Visual Communications and Image Processing (VCIP)*, Dec. 2017, pp. 1–4.

[14] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," Oct. 2016, arXiv: 1511.00561.

[15] J. Fu, J. Liu, Y. Wang, J. Zhou, C. Wang, and H. Lu, "Stacked Deconvolutional Network for Semantic Segmentation," *IEEE Transactions on Image Processing*, pp. 1–1, 2019, conference Name: IEEE Transactions on Image Processing.

[16] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation," in *2016 Fourth International Conference on 3D Vision (3DV)*, Oct. 2016, pp. 565–571.

[17] H. Kervadec, J. Bouchtiba, C. Desrosiers, E. Granger, J. Dolz, and I. B. Ayed, "Boundary loss for highly unbalanced segmentation," in *International Conference on Medical Imaging with Deep Learning*, May 2019, pp. 285–296, iSSN: 1938-7228 Section: Machine Learning.

[18] Q. Sun, Y. Liu, T.-S. Chua, and B. Schiele, "Meta-Transfer Learning for Few-Shot Learning," *arXiv e-prints*, p. arXiv:1812.02391, Dec. 2018.

[19] A. Mosinska, P. Márquez-Neila, M. Kozinski, and P. Fua, "Beyond the pixel-wise loss for topology-aware delineation," *CoRR*, vol. abs/1712.02190, 2017. [Online]. Available: http://arxiv.org/abs/1712.02190