Published in Image Processing On Line on 2024–09–19.
Submitted on 2022–07–04, accepted on 2023–07–27.
ISSN 2105–1232 © 2024 IPOL & the authors CC–BY–NC–SA
This article is available online with supplementary materials,
software, datasets and online demo at
https://doi.org/10.5201/ipol.2024.499

On the Domain Generalization Capabilities of Interactive Segmentation Methods

Franco Marchesoni-Acland, Tanguy Magne, Fayçal Rekbi, Gabriele Facciolo

Université Paris-Saclay, ENS Paris-Saclay, Centre Borelli, Gif-sur-Yvette, France
{marchesoniacland, gfacciol}@gmail.com

Communicated by Pablo Musé *Demo edited by* Franco Marchesoni

Abstract

Interactive image segmentation (IIS) methods are usually trained over segmentation datasets containing natural images. They are also usually evaluated over natural images. However, the most common use case is the annotation of new images from a different domain. Yet, the performance of IIS methods on a different domain is seldom reported. In this work, we evaluate a state-of-the-art IIS method trained with natural images over an aerial image dataset. Its performance is compared to the performances the method achieves when being trained/finetuned with aerial images. The comparison reveals that there is a big domain generalization gap.

Source Code

This is an MLBriefs article, the source code has not been reviewed!
The original implementation of the method is available here[1].

Keywords: interactive-image-segmentation; out-of-distribution; domain-adaptation

1 Introduction

Deep Learning (DL) has revolutionized most of computer vision by leading to significant improvements across different tasks. Image segmentation, i.e. pixel classification, is a key topic in image processing and computer vision. It plays a central role in a broad range of applications, from analyzing traffic to monitoring environmental changes. In the case of satellite imagery, the objects to be segmented may be, for example, buildings, roads, or forests.

However, the development of DL models for image segmentation requires a huge amount of annotated data to achieve optimal performance. In the case of segmentation, the effort to obtain annotations by drawing pixel-accurate masks is significant and makes the collection of this training data difficult. Moreover, users usually need to correct the segmentation results which are not satisfactory enough.

[1] https://github.com/SamsungLabs/ritm_interactive_segmentation

A more practical alternative are click-based interactive segmentation methods. They enable the improvement of the segmentation result by iteratively indicating the mislabeled areas and refining the segmentation masks. One such state-of-the-art interactive image segmentation (IIS) model is `ritm` [9], which presents superior performance, robustness, and simplicity when compared to competitors. The current work will analyze the domain adaptation capabilities of the `ritm` method. There is much interest in using this kind of methods for annotating new data. However, this new data is most frequently not of the same kind as the original data used to train the IIS models. This paper shows two things: i) the evaluated IIS method presents a large domain generalization gap and ii) how the IIS method compares to a traditional supervised model in performance. The domain generalization gap is barely mentioned in the literature, although it is very important for real annotation use cases.

2 Methods

Interactive image segmentation (IIS) methods allow users to define a binary mask by iteratively interacting with the algorithm (although not indefinitely). Early attempts at doing this, even before deep learning became trendy, were trying to minimize a specific loss on a single image, using classical optimization methods. With the development of deep learning, methods based on convolutional neural networks were introduced, with strategies to simulate users' clicks at training time. They brought some improvement but were still far from perfect. More recently, new groups of methods have emerged based on the Backpropagating Refinement Scheme (BRS) [3], that find the binary segmentation by minimizing an energy at test-time. The state-of-the-art, however, has come back to simple deep-learning schemes without test-time optimizations [9].

2.1 Input Modalities

For each of the IIS methods, classical and current, many strategies can be used to take into account the inputs of the user. A non-exhaustive list of user input options is:

- Bounding boxes.
- Extreme points.
- Scribbles.
- Roughly painting borders.
- Positive/negative clicks.

Click-based methods are used in the article of the evaluated model [9], as they are easy to simulate, enable a compact representation, and users can refine the prediction mask iteratively and thus easily click on parts wrongly labeled by the model. In this modality, the user provides guidance in the form of positive (negative) clicks, indicating which pixels belong to the foreground (background) region. A positive (negative) click is obtained by pressing the left (right) mouse button.

2.2 Description of `ritm`

IIS algorithms [9] allow users to explicitly control the predictions using interactive input at several iterations. This is in contrast to common semantic and instance segmentation algorithms that can only input an image and output a segmentation mask in one pass. Such interaction makes it possible to select an object of interest and correct prediction errors.

2.3 Model Architecture

An IIS method such as `ritm` is a tool that allows a user to quickly achieve a desired binary segmentation. Binary segmentation implies determining a binary mask, i.e. differentiating foreground pixels from background pixels. As mentioned above, in this framework, the user guides the segmentation with positive or negative clicks, indicating which pixels belong to the foreground or the background region, respectively. The IIS method represents the user's clicks as a click map, concatenates the click map with the image over the channel dimension, and then inputs them to a neural network that outputs a probability map.

From an architectural point of view, the interactive segmentation task is similar to semantic segmentation. The traditional segmentation networks take as input a high-resolution image and provide as output the different segmentation masks, i.e. a pixel classification. However, image segmentation networks do not allow for interactivity. For IIS, the backbone architecture as such is unchanged, but it is necessary to adapt it to integrate the user input. The user inputs are clicks, represented by their coordinates in an image and later encoded in a spatial map. There are different types of encoding and the reportedly most performing one is encoding clicks as disks with a small fixed radius. Most image segmentation (not necessarily binary) models take RGB images as input. To process the information from a user's encoded clicks, it is necessary to increase the weights of the first convolutional layer of a pre-trained model to accept an N-channel input instead of an RGB image. To do so, many strategies exist, but ablation studies in the article showed the best early-fusion variant: the click map is passed through a convolutional layer and the RGB image is passed through another convolutional layer with the same output dimension. These two results are then added together and passed to the backbone model, which is, in this case, the HRNet [10].

3 Training and Data

3.1 Training Strategy

In the past, a commonly used strategy to simulate users' clicks at training time was to generate positive and negative clicks randomly without taking into account the relationships between them. In practice, each new click would be placed, for a given prediction, in the areas where errors are made. In the paper under study, an iterative sampling of the different clicks is implemented. In this way, the generated clicks are more similar to the interaction with a real user. However, full iterative sampling is computationally very expensive. Curated random sampling is therefore used to provide an initial set of clicks to the algorithm, and afterward, some clicks are added using the iterative sampling procedure. The iterative sampling process determines that each next click is sampled from the region obtained by applying the morphological erosion operation to the mislabeled region. The training loss is the normalized focal loss [5], which pays more attention to misclassified pixels.

3.2 Data

3.2.1 Dataset

To test the models' domain generalization capabilities, they were applied to a remote sensing problem. The problem is defined by the INRIA Aerial Image Labeling dataset [2]. The dataset was constructed by combining public domain imagery and public domain official building footprints. The problem here addressed is pixel-wise labeling of aerial imagery. In particular, the dataset focuses on the classification of pixels between two categories, building or not building.

This dataset is composed of 810 km^2 of images. Only 405 km^2 of them have available labels. The dataset covers five areas in the world: Chicago (a dense urban area), Austin (an American city with suburbs), Vienna (a less dense European urban area), Tyrol, and Kitsap (more rural areas). There are 180 images of 5000×5000 pixels each. Associated with each of these images, there is a mask image, which is a binary image of the same size, and far from perfect. In this mask image, building pixels are white, and the other pixels are black.

3.2.2 Data Preprocessing

The original size of the images is prohibitive. They are too big to be given directly to a neural network. Therefore, the first step was to crop them. In the original `ritm` paper [9], the authors used images of size 320×480. Informed by that, the preprocessing here presented involves the partition of the images in non-overlapping crops of size 500×500 pixels. This enables us to have 100 cropped images per original image, without losing any part of them, while at the same time keeping the image size small enough and close to that in the original article. Note that this division creates some images with no positive labels (in rural areas). This is not a real problem, it corresponds to the user doing no clicks.

3.2.3 Experiments

For the experiments, a train/test split was made. The 5 first original images of each location were used for the test set so that there are 25 full-size test images (5 images per 5 sites) and 155 full-size train images. Because of the preprocessing explained before, there are in the end 15500 cropped train images and 2500 cropped test images, all of 500×500 pixels size. However, out of the 2500 test images, 560 have no building on them. These images are dropped out from the computation of the metrics (presented below), as they artificially increase the performance of the model by yielding an IoU of 100% without any clicks over those images. There are, in the end, 1940 test images over which the metrics are computed.

4 Experiments

Here three IIS models are evaluated under the metrics presented in the next section. The evaluation is done over the dataset presented in Section 3.2. The models are:

1. **No training:** The model is not trained on aerial images, but only pretrained on natural images.

2. **Finetuning:** The same model but fine-tuned on the training aerial images.

3. **Training from scratch:** A model trained from scratch on the training aerial images.

4.1 Metrics

Our metrics are the usual metrics for image segmentation. As image segmentation means pixel classification, metrics for classification are also useful here. They are both based on the same fundamental quantities.

	NoC_{20}@ 80% ↓	NoC_{20} @90% ↓	≥ 20 @80% ↓	Avg 20 IoU ↑
No Training	16.22	18.6	0.648	0.718 (±0.166)
Training from scratch	10.81	18.05	0.333	0.827 (±0.084)
Finetuning	**5.66**	**14.58**	**0.135**	**0.875 (±0.072)**

Table 1: Results on NoC for some IoUs.

4.1.1 Classification Metrics

Let TP be the number of true positive predictions, i.e. the number of pixels predicted as foreground (positive) that are indeed foreground according to the ground truth (true). Analogously, we define the False Positive (FP), True Negative (TN), and False Negative (FN) numbers. Then:

- **Accuracy**= $\frac{TP+TN}{TP+TN+FP+FN}$, measures the exactitude of the predictions.
- **Precision**= $\frac{TP}{TP+FP}$, measures how many of the foreground predictions were correct.
- **Recall**= $\frac{TP}{TP+FN}$, measures how many of the foreground pixels were correctly detected.

4.1.2 Segmentation Metrics

These metrics come from spatial intuition but can be written in terms of the basic TP, FP, TN, and FN:

- **Dice**= $\frac{2\times TP}{(TP+FP)+(TP+FN)}$, the Dice coefficient measures the ratio between the double of the intersection of estimated and ground truth masks and the sum of their areas.
- **IoU**= $\frac{TP}{TP+FN+FP}$, the Intersection over Union measures the ratio between the intersection of estimated and ground truth masks and their union.

Some other metrics can be derived from these base ones. The first one is the number of clicks required to achieve a certain IoU threshold. This metric will be denoted by $\text{NoC}_{\texttt{NoC}}$@`IoU`. As an example, if we write NoC@90 we mean the number of clicks needed to reach an IoU of 90%. and NoC_{20}@90 means the minimum between 20 and NoC@90. Therefore NoC_{20}@90 is a lower bound of NoC@90. We add a maximum number of clicks to ensure the existence of a metric for all possible IoUs.

To check how tight this lower bound is, one can look at the metric that gives the percentage of samples in the test set for which some specific IoU was not reached after a certain number of clicks. This metric is denoted ≥ `NoC@IoU`, e.g. ≥ 20 @85 is the percentage of the test samples that have not reached an IoU greater than 85% after 20 clicks.

Finally, a simpler and more comprehensive metric that can be used to compare with a noninteractive model is the average IoU after a certain number of clicks. This metric is named $\text{Avg}_{\texttt{Noc}}$ IoU in this paper. The `NoC` corresponds to the number of clicks for which the IoU is computed. Note that for this metric, the standard deviation on the test set is given inside parenthesis.

5 Results

5.1 Quantitative Results

The results obtained are presented in Table 1 and Figure 1. Several things can be observed. Firstly, finetuning gives the best result. The difference between the model trained from scratch (orange) and

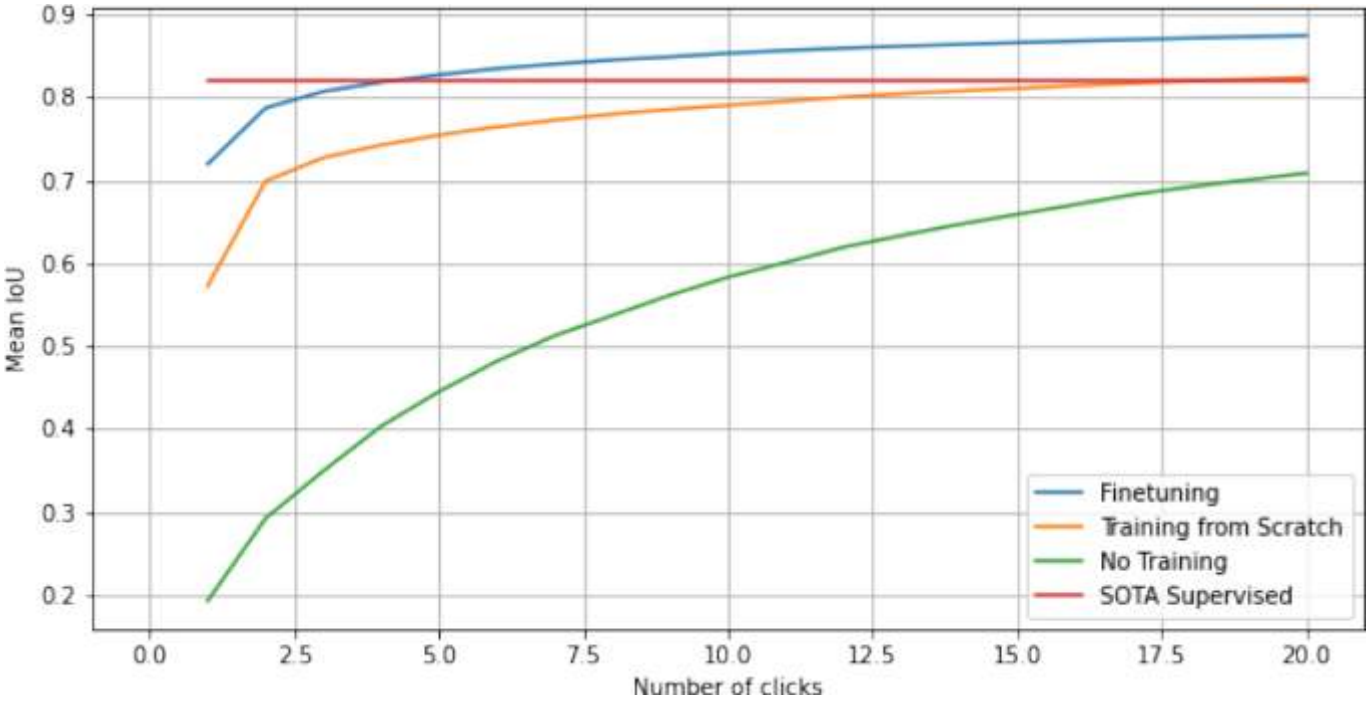


Figure 1: IoU vs NoC for all models over the test dataset.

the model pretrained with natural images only (green) is very large. This domain generalization gap is given by the difference between these orange and green lines in Figure 1. The gap is 0.4 IoU in the beginning and it is reduced until 0.1 IoU when 20 clicks are reached and presents a substantial performance drop that is not being considered by the literature.

Therefore, to get the best results, a model specialized in the target domain is needed. On top, such a model can strongly benefit from a pretraining on another type of images, preferably a diverse set. Indeed, the model that was trained on natural images has already learned to segment many different shapes. The shapes we can see in natural images are different from those of buildings seen from high above. However, many priors are maintained and useful, such as considering borders and higher-level descriptors. Also, having a model already specialized in interactive segmentation helps because it does not have to learn the task to accomplish, i.e. how to condition a segmentation on the clicks, it just has to further specialize in the extraction of visual features relevant to the domain.

5.2 Traditional Supervised Segmentation

A ranking of the different supervised methods that have been proposed is available on the dataset website[2]. Two good-performing traditional supervised segmentation models, [1] and [7], reach an average IoU between 0.80 and 0.82. These models are based on a modification of the U-net [8] and on the FPN [4], which are traditional image segmentation architectures.

The curves in Figure 1 show the state-of-the-art supervised performance with a red line, which is around 0.80 - 0.82 mIoU. This performance surpasses the IIS' trained from scratch while using strictly less information: the same labels but no annotator feedback. Even the best performing training method, finetuning, is inferior in performance than the non-interactive supervised model when using less than three clicks. However, the method can improve over the traditional supervised method when enough clicks are provided.

[2] https://project.inria.fr/aerialimagelabeling/leaderboard/

5.3 Qualitative Results

The qualitative results recover the quantitative results presented above. Indeed, for aerial images, the finetuned model is consistently better than the model trained from scratch, which is better than the model that has not been retrained.

In more detail, looking at the result on Austin (Figure 2) and Kitsap (Figure 5), it can be noticed that the finetuned model allows recovering straighter lines as the border of buildings. It also recovers finer details. In the Chicago example (Figures 3 and 4), the finetuned model shows the capability of differentiating the buildings individually, which was not the case for the model trained on natural images. The model can also find buildings even if no clicks were made on them. This is an important point because there can be many buildings on a single image, and it is not desirable to have to click on each of them. The results on Tyrol (Figure 6) suggest that for images that had already good results with the model pretrained on natural images only, the number of clicks required to achieve the same level of IoU is smaller. Finally, the results on the Vienna (Figure 7) image show a combination of every comment above. Indeed, in this case, the corners are less rounded with the finetuned model, all buildings are distinct, and even tiny buildings can be recovered.

Figures 8, 9 and 10 show the output of the methods for a natural image when imposing a target IoU of 0.95. The results are visually similar because clicks are made until the output mask is correct. However, the pretrained model uses only 2 clicks, the finetuned model uses 7 clicks, and the model trained from scratch uses 3 clicks. Surprisingly, the pretrained model fine-tuned for aerial images performs worse than the model trained for aerial images from scratch. This is surprising, but consistent, as shown in Figures 8, 9 and 10, where the target IoU is 0.9, the pretrained model uses 2 clicks, the finetuned model uses 11 clicks, and the model trained from scratch uses 9 clicks.

6 Demo

The demo allows one to choose any pre-trained model and see how it performs over any input image. These can be, in particular, natural images or aerial images, that have been seen by some of the models. A second image from which to generate the target mask is also required as input, as the target mask is needed to guide the automated clicking that is part of the inference. This image will be first converted to grayscale and second binarized relative to a threshold of 128. After the input image and the target mask are available, the automatic annotation will run by making clicks until a target IoU is reached or 20 clicks are made. In the demo, positive (negative) clicks, which intend to indicate foreground (background), are marked with green (red) circles. The clicks are made following the evaluation procedure in [9], which involves clicks near the center of the largest misclassified region. Example results are shown in Figures 8, 9, 10, 11, 12, and 13.

(a) Results of the ***No Training*** experiment

(b) Results of the ***Training from scratch*** experiment

(c) Results of the ***Fintuning*** experiment

Figure 2: Results of the different experiments on image 1 from Austin location (crop number 1-2). Left: initial image overlaid with the predicted mask (in red), negative clicks (in red) and positive clicks (in green). Middle: Predicted mask. Right: Ground truth mask.

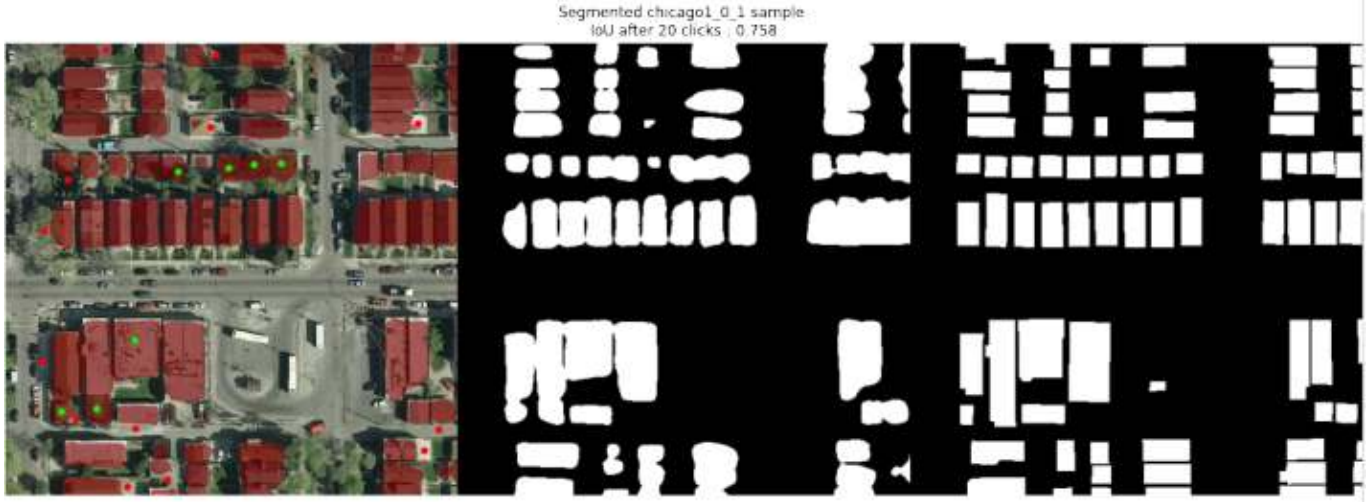
(a) Results of the ***No Training*** experiment

(b) Results of the ***Training from scratch*** experiment

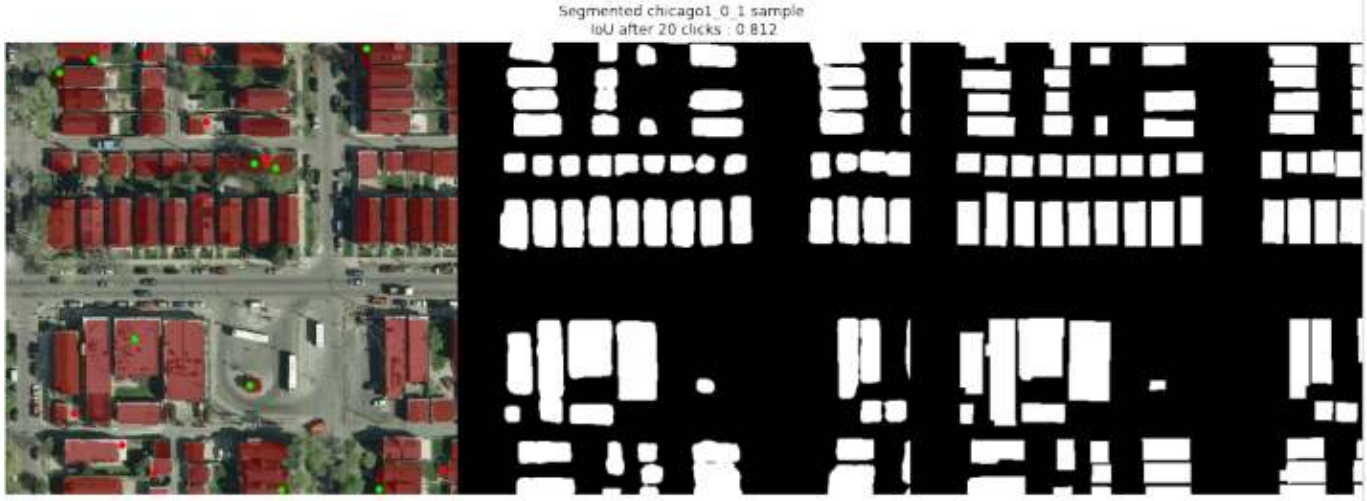
(c) Results of the ***Fintuning*** experiment

Figure 3: Results of the different experiments on image 1 from Chicago location (crop number 0-0). Left: initial image overlaid with the predicted mask (in red), negative clicks (in red) and positive clicks (in green). Middle: Predicted mask. Right: Ground truth mask.

(a) Results of the ***No Training*** experiment



(b) Results of the ***Training from scratch*** experiment



(c) Results of the ***Fintuning*** experiment

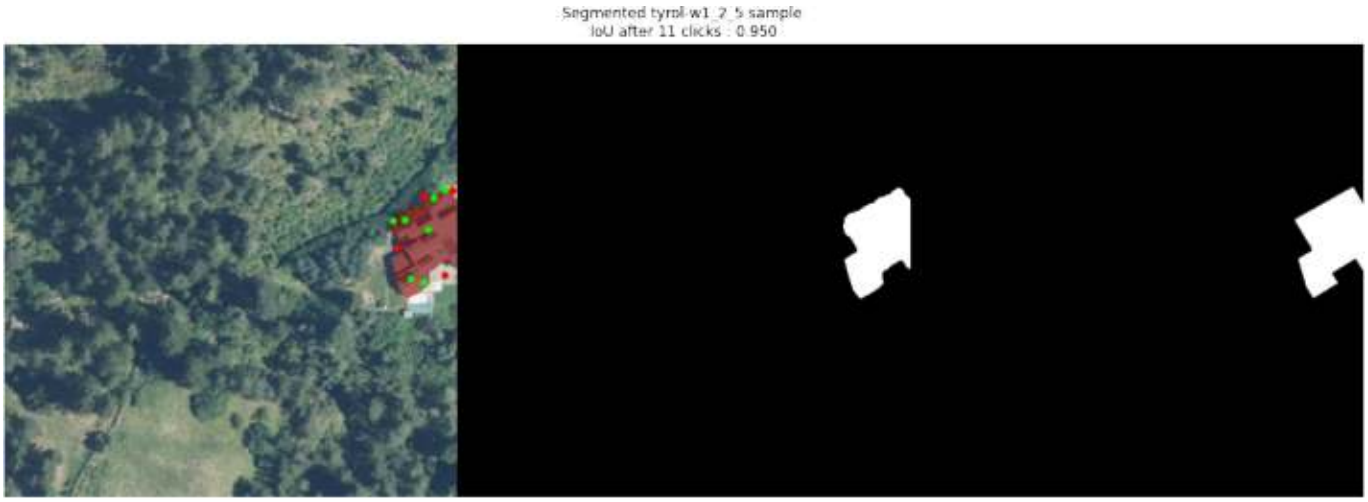
Figure 4: Results of the different experiments on image 1 from Chicago location (crop number 0-1). Left: initial image overlaid with the predicted mask (in red), negative clicks (in red) and positive clicks (in green). Middle: Predicted mask. Right: Ground truth mask.

(a) Results of the ***No Training*** experiment

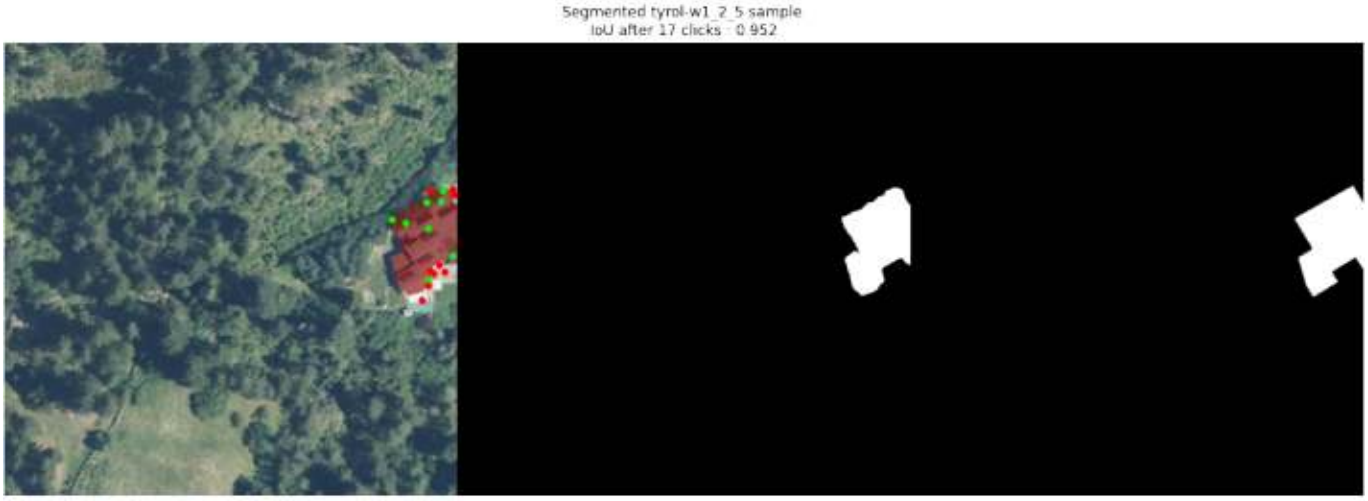
(b) Results of the ***Training from scratch*** experiment

(c) Results of the ***Fintuning*** experiment

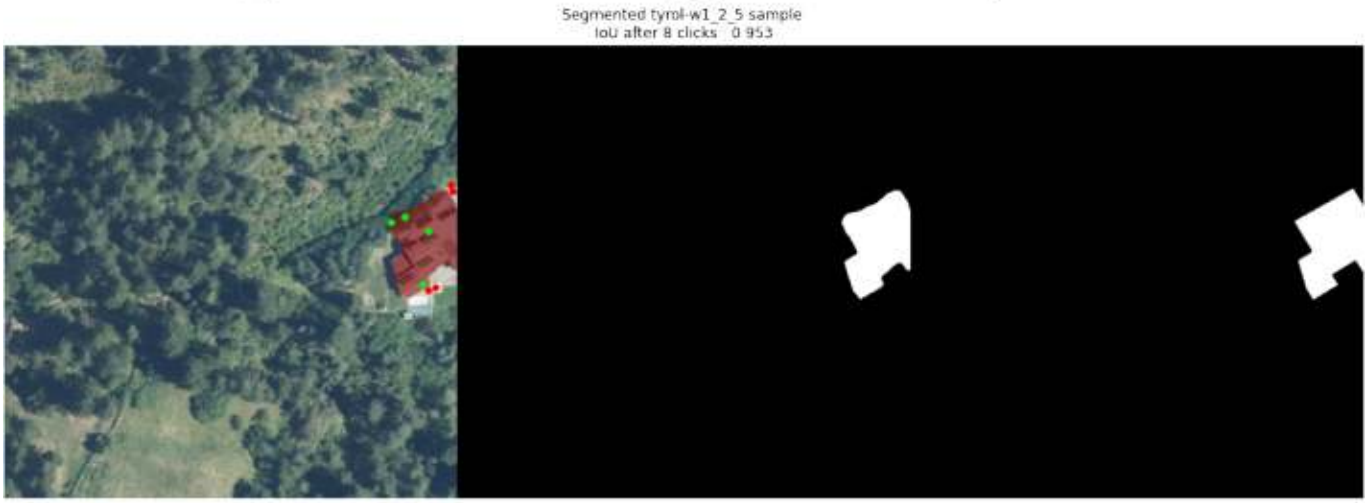
Figure 5: Results of the different experiments on image 1 from Kitsap location (crop number 2-6). Left: initial image overlaid with the predicted mask (in red), negative clicks (in red) and positive clicks (in green). Middle: Predicted mask. Right: Ground truth mask.



(a) Results of the ***No Training*** experiment



(b) Results of the ***Training from scratch*** experiment



(c) Results of the ***Fintuning*** experiment

Figure 6: Results of the different experiments on image 1 from Tyrol location (crop number 2-5). Left: initial image overlaid with the predicted mask (in red), negative clicks (in red) and positive clicks (in green). Middle: Predicted mask. Right: Ground truth mask.

(a) Results of the ***No Training*** experiment

(b) Results of the ***Training from scratch*** experiment

(c) Results of the ***Fintuning*** experiment

Figure 7: Results of the different experiments on image 1 from Vienna location (crop number 0-0). Left: initial image overlaid with the predicted mask (in red), negative clicks (in red) and positive clicks (in green). Middle: Predicted mask. Right: Ground truth mask.

Figure 8: Finetuned model. Target IoU 0.95. On the left, we can see the input image overlaid with the output mask (red), the positive (green), and the negative (red) clicks. On the right, we can see the pixels classified into true positives (white), true negatives (black), false positives (blue), and false negatives (red).

Figure 9: Trained from scratch model. Everything else as in Figure 8.

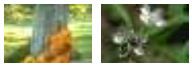
Figure 10: Pretrained model. Everything else as in Figure 8.

Figure 11: Finetuned model. Target IoU 0.90. On the left, we can see the input image overlaid with the output mask (red), the positive (green), and the negative (red) clicks. On the right, we can see the pixels classified into true positives (white), true negatives (black), false positives (blue), and false negatives (red).

Figure 12: Trained from scratch model. Everything else as in Figure 11.

Figure 13: Pretrained model. Everything else as in Figure 11.

Image Credits

 Extracted from [6].

 Extracted from [2].

References

[1] B. Chatterjee and C. Poullis, *Semantic Segmentation from Remote Sensor Data and the Exploitation of Latent Learning for Classification of Auxiliary Tasks*, CoRR, (2019), https://doi.org/10.48550/arXiv.1912.09216.

[2] B. Huang, K. Lu, N. Audeberr, A. Khalel, Y. Tarabalka, J. Malof, A. Boulch, B. Le Saux, L. Collins, K. Bradbury, S. Lefèvre, and M. El-Saban, *Large-Scale Semantic Classification: Outcome of the First Year of INRIA Aerial Image Labeling Benchmark*, in IEEE International Geoscience and Remote Sensing Symposium (IGARSS), IEEE, 2018, pp. 6947–6950.

[3] W.-D. Jang and C.-S. Kim, *Interactive Image Segmentation Via Backpropagating Refinement Scheme*, in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 5297–5306. https://openaccess.thecvf.com/content_CVPR_2019/html/Jang_Interactive_Image_Segmentation_via_Backpropagating_Refinement_Scheme_CVPR_2019_paper.html.

[4] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, *Feature Pyramid Networks for Object Detection*, in IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2117–2125, https://doi.org/10.1109/CVPR.2017.106.

[5] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, *Focal Loss for Dense Object Detection*, 2017, https://doi.org/10.48550/ARXIV.1708.02002.

[6] D. Martin, C. Fowlkes, D. Tal, and J. Malik, *A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics*, in IEEE International Conference on Computer Vision (ICCV), vol. 2, IEEE, 2001, pp. 416–423, https://doi.org/10.1109/ICCV.2001.937655.

[7] A. Milosavljević, *Automated Processing of Remote Sensing Imagery Using Deep Semantic Segmentation: A Building Footprint Extraction Case*, ISPRS International Journal of Geo-Information, 9 (2020), https://doi.org/10.3390/ijgi9080486.

[8] O. Ronneberger, P. Fischer, and T. Brox, *U-Net: Convolutional Networks for Biomedical Image Segmentation*, CoRR, (2015), https://doi.org/10.48550/arXiv.1505.04597.

[9] K. Sofiiuk, I. A. Petrov, and A. Konushin, *Reviving Iterative Training with Mask Guidance for Interactive Segmentation*, 2021, https://doi.org/10.48550/arXiv.2102.06583.

[10] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao, *Deep High-Resolution Representation Learning for Visual Recognition*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 43 (2020), pp. 3349–3364, https://doi.org/10.1109/TPAMI.2020.2983686.