Deep interactive segmentation of satellite imagery

Tanguy MAGNE
tanguy.magne@minesparis.psl.eu

Fayçal REKBI
faycal.rekbi@ens-paris-saclay.fr

April 7, 2022

Abstract

Image segmentation involves detecting and classifying individual objects within an image. Recently a new method known as interactive segmentation has emerged and aroused strong interest in Deep Learning. Unlike traditional methods, it enables users to select objects of interest accurately, allowing high-quality labelization of large datasets. In this project, we propose a first approach to use interactive segmentation for remote sensing imagery.

1 Introduction

Currently, deep learning has revolutionized computer vision and has led to significant improvements across different tasks. Image segmentation is a key topic in image processing and computer vision. It involves partitioning images or video frames into multiple segments or objects. In the case of satellite imagery, these objects may be buildings, roads, or forests. Segmentation plays a central role in a broad range of applications, from analyzing traffic to monitoring environmental changes.

In general, the development of Deep Learning 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. Therefore, the more practical approaches are based on click-based interactive segmentation. They allow for improving the segmentation result by iteratively marking the mislabeled areas and refining the segmentation masks. The article we are considering [7] explores a new segmentation architecture. The authors elaborate a click-based interactive segmentation that allows modifying the segmentation masks interactively, and that is trained iteratively.

In this report, we will first conduct a small literature review, to be able to understand the model used in the framework of classical supervised segmentation methods and understand what methods already exist to perform interactive segmentation. Then we will detail the contribution of the article considered. Finally, we will apply this method to satellite imagery, and present the results thus obtained.

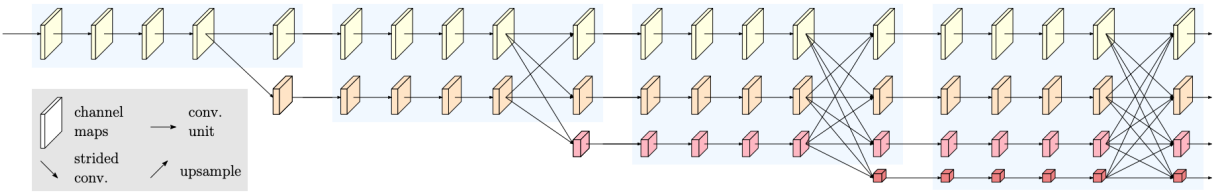


Figure 1: HRNet architecture

2 Related Work

2.1 Deep Learning - Based Image Segmentation Models

Many segmentation models have been proposed [5]. These architectures can be classified into different categories, for example, architectures with an encoder/decoder or skip connections.

2.1.1 Fully Convolutional Networks

The first segmentation network was based on a fully convolutional network (FCN) [2] which includes only convolutional layers. This enables it to take an image as input and produce a segmentation map. For this purpose, the authors have used classical architectures such as a VGG where they have replaced all fully-connected layers with fully-convolutional layers. Therefore, the model outputs a spatial segmentation map instead of classification scores.

2.1.2 Encoder-Decoder Based Models

UNet Several models were initially developed for medical/biomedical image segmentation inspired by FCN models. More sophisticated CNN architectures for semantic segmentation are DeconvNet and U-Net [6]. Both networks share a similar architectural idea: the network consists

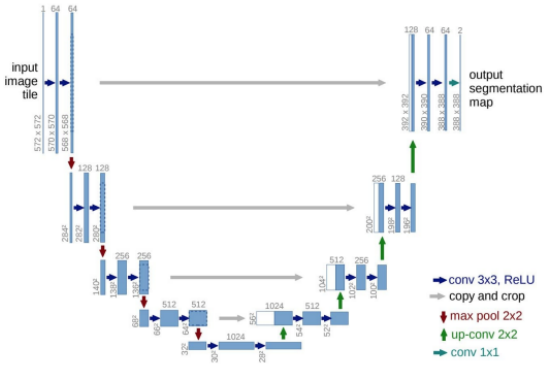


Figure 2: UNet architecture

of encoder and decoder parts. The U-Net architecture consists of two paths, a path to capture context, and a symmetric expanding path that allows precise localization. The U-Net architecture introduces skip connections, which allow duplicating and concatenating previous encoder feature maps and corresponding upsampled decoder feature maps so that decoder convolutional layers process them simultaneously (Figure 2). This allows the network to output fine details that are lost in the smaller encoded version of the image.

HRNet Another popular family of deep models for image segmentation is based on the convolutional encoder-decoder architecture [5].

Most of the DL-based segmentation methods use some kind of encoder-decoder model. One of the most popular model in this category is the recently developed segmentation network, the High-Resolution Network (HRNet). In addition to recovering high-resolution representations in U-Net for instance, HRNet preserves high-resolution through encoding by concatenating high- and low-resolution convolutional streams in parallel and repeatedly exchanging information across resolutions (Figure 1).

2.1.3 Other models

There are many segmentation models that have evolved over time (Figure 3), such as Multi-Scale and Pyramid Network-Based Models, R-CNN Based Models, Recurrent Neural Network Based Models, Attention-Based Models, Generative Models and Adversarial Training, CNN Models With Active Contour Models or Convolutional Models With Graphical Models. [5]

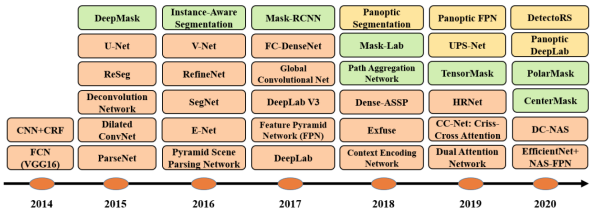


Figure 3: Timeline of DL-based segmentation model. Orange, green and yellow refer to semantic, instance and panoptic segmentation algorithms.

2.2 Interactive segmentation methods

As explained in the introduction of this report, iterative and interactive segmentation allow to refine the prediction mask iteratively and thus let the user give feedback to improve the segmentation. 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, CNN-based methods were introduced, with strategies to simulate users' clicks at training time. They allowed some improvement but were still far from being perfect. More recently, new groups of methods have emerged and are based on the Backpropagating Refinement Scheme (BRS), which allows optimizing the segmentation by minimizing energy. This method shows evidence of greater performance but also improve greatly inference time, which is an important problem.

For each of these methods, many strategies can be used to take into account the inputs of the user :

- Bounding boxes are easy to determine but don't give a lot of information to the model.
- Approaches based on extreme points allow compact representation but it is rather difficult for the user to determine these extreme points.
- Scribble-based methods give a lot of information but are hard to simulate at training time when training a neural network.

For all these reasons, click-based methods are used in the article, as they are easy to simulate, allow for compact representation, and users can easily click on parts wrongly labeled.

3 Deep interactive segmentation

Interactive segmentation algorithms [7] allow users to explicitly control the predictions using interactive input at several iterations. 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.

Model Architecture The interactive segmentation task is similar to semantic segmentation from an architectural point of view. Indeed, the traditionally used networks take as input a high-resolution image and provide as output the different segmentation masks. However, within the network the information processing is different. Indeed, the architecture as such is unchanged but it is necessary to modify some parts. In interactive segmentation, it is common to use positive and negative clicks. The latter are represented by their coordinates in an image that is encoded in a spatial form. There are different types of encoding and the most relevant one is the one based on disks with a small fixed radius.

Most ImageNet pre-trained models only 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 that Conv1S is the best. With this solution, both the RGB image and the 2 channels of the encoded clicks are first passed to a convolution layer which outputs two images with the same number of channels. These two results are then added and passed to the backbone model.

Training strategy The strategy commonly used until now 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 should be placed for a given prediction in the areas where errors are made. In the paper we are studying, an iterative sampling of the different clicks is implemented. This way, the generated clicks are similar to the interaction with a real user. However, full iterative sampling is very computationally expensive. Random sampling is therefore used to initialize the algorithm, and then some clicks are added using the iterative sampling procedure.

The iterative sampling process is as follows:

- Each point is sampled from the center of a mislabelled region and the region obtained by applying the morphological erosion operation to the mislabelled region.
- We do not save the clicks simulated during training and we simulate the user's clicks for each batch.
- Random (for initialization) and iterative sampling is used to train the models.

Loss Binary cross entropy (BCE) is commonly used to train a semantic segmentation model. However, it has a major drawback which is to treat correctly segmented areas in the same way as those that are not. To overcome this problem, the author of the article used the focal loss function, as is often the case when trying to address semantic segmentation. Let $\hat{M}$ the output of the

network and $p_{i,j}$ the confidence of prediction at point (i, j)

$$FL(i,j) = -(1-p_{i,j})^{\gamma} \log p_{i,j}$$

We can also express the total weight which decreases when the accuracy of the prediction increases :

$$P(\hat{M}) = \sum_{i,j} (1-p_{i,j})^{\gamma}$$

The total gradient of the focal loss fades with time. Therefore, the normalized focal loss (NFL) is used :

$$NFL(i,j,\hat{M}) = -\frac{1}{P(\hat{M})}(1-p_{i,j})^{\gamma} \log p_{i,j}$$

This provides a better loss function as the gradient does not attenuate over time. In addition, it provides better accuracy and convergence.

4 Contributions

We want to apply this method to satellite imagery. The first step is to build the dataset that will be used to evaluate the performance of our model. To do this, we will use the Inria Aerial Image Labeling dataset [3]. The dataset was constructed by combining public domain imagery and public domain official building footprints. The data will be partitioned into a training and test set.

In a second step, we will apply the model trained on natural images, directly to the test set. We will compare this model to a model partially re-trained, fine-tuned on the training images, and evaluated on the test set. Then, we will compare the results with the non-interactive segmentation method as a reference.

Finally, we will analyze the performance and relevance of such models for remote sensing imagery.

5 Experiments

In this section, we will explain the experiments we ran, and the result we obtained by applying the methodology described by [7] on satellite images.

5.1 Preprocessing of datas

Dataset used The first step was to get the dataset and preprocess the images. We used the Inria Aerial Image Labeling dataset [3]. This dataset addresses the problem of the pixel-wise labeling of aerial imagery. More particularly, it 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. We, therefore, focus on this part of the dataset to be able to get metrics. The dataset covered five areas in the world: Chicago is a dense urban area, Austin is an American suburb, Vienna is a less dense European urban area, and Tyrol and Kitsap are more rural areas. There are 180 images of 5000×5000 pixels each. With each of these images, there is a mask image, which is a binary image of the same size. In this mask image, *building* pixels are white and *not building* pixels are black. We used the 5 first images of each location for the test set, so we have 25 test images and 155 train images.

Preprocessing The original size of the images is prohibitive. They are too big to be given di-

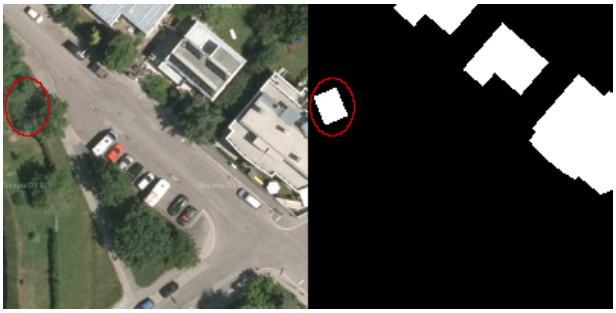
rectly to a neural network. Therefore, the first step was to crop them. As in the paper, the authors use images of size 320×480, we decided to crop images into 500×500 pixels non-overlapping images. It allows us to have 100 cropped images per original image, without losing any part of it, while keeping the image size small enough, and close to that of the original article. Note that this division creates images with no positive labels (in rural areas). This is not a real problem, it corresponds to the user doing no clicks. Because of the split explained before, we are left with 15500 cropped train images and 2500 cropped test images.

Discussion on the dataset Several criticisms can be made about this dataset. The first obvious one is the fact that it contains only two categories, which are *buildings* and *not buildings*. We would have like to also have the possibility to segment trees, forests, roads, or lakes.

The second and probably most important problem is the fact that it contains many false positives and negatives. The mislabeled pixels are due to three things :

- Wrong labeling of parts of the image as building, while there are no buildings, as can be seen in Figure 4a.
- Buildings that are not labeled, as can be seen in Figure 4b.
- Shift between the image and the mask. The buildings are indeed labeled but in the wrong place. Sometimes, it's just a mistake, but sometimes there is a constant shift. The two cases are shown in Figure 4c.

Other critics can be made about the labeling of the dataset. Indeed, on one hand, we have a lot of false-positive and negative labels, and on the other hand, the dataset is sometimes too precise. Indeed, sometimes, *buildings* are separated by a thin line of *not buildings* labels. Even as a human it is hard to say if the buildings are separated or attached, as presented by Figure 5a. Moreover, sometimes, the roofs of the build-



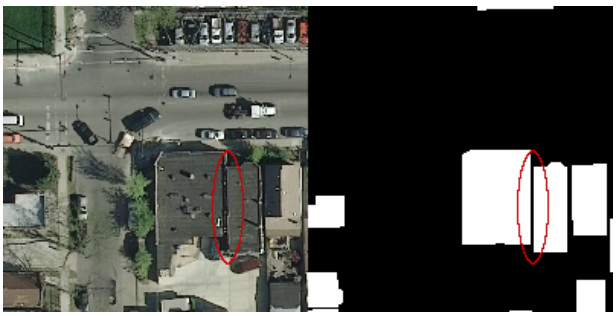
(a) Created buildings (False Positive)

(b) Missing buildings (False Negative)

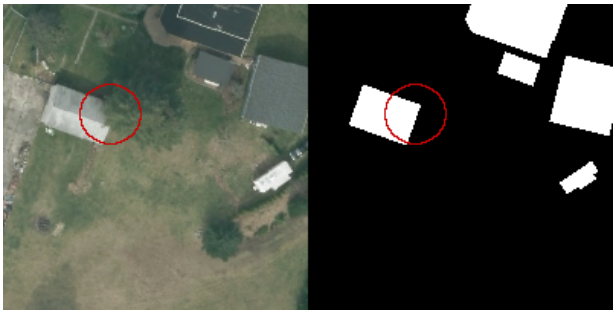
(c) Misalignment between the image and the masks

Figure 4: Example of false positive and negative

ings are covered by vegetation such as trees. In this case, the label can be *building* while we can see that there is a tree. Even if as humans we can correctly understand that there is indeed a building below the tree, we wouldn't necessarily want to label these pixels as buildings because if we were able to classify the pixels into more categories, we would probably have classified them as trees. This can be seen in Figure 5b.



(a) Separated buildings



(b) Tree hiding a building

Figure 5: Example of flaws in the dataset labelling

Then, about the images themselves, some of them have a watermark. It is the case with all Vienna images. This can be seen in Figure 6. It is a quite huge problem because, in practice, real images that we acquire from satellites or planes won't have any watermark. This introduces a variation between the training and test dataset.

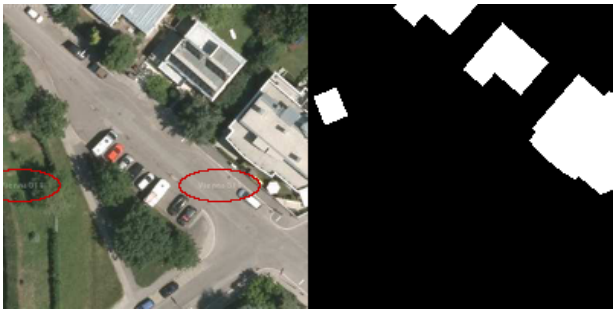


Figure 6: Watermark inside an image

Nevertheless, and even if we point out these few flaws, we can say that the dataset is of high quality, and we can be happy to have it.

5.2 Experiments

To evaluate the performance of iterative training for interactive segmentation of satellite images, we did 3 experiments:

- The first idea was just to use the model trained on non-satellite images, that was made available by the authors. This experiment is referred to as ***No Training*** in this report.
- We also try to train from scratch the model architecture proposed by the authors, using the same training routine, but on the dataset of satellite images. This experiment is referred to as ***Training from scratch*** in this report.
- The last experiment was to finetune the pretrained weights given by the authors on the dataset of satellite images. This experiment is referred to as ***Finetuning*** in this report.

To understand the benefit of interactive segmentation over simple supervised segmentation,

we also compared our models to classical supervised segmentation models trained on the dataset of satellite images. This is referred to as ***Not Interactive*** in this report.

5.3 Results

Metrics Used To assess the quality of the results we obtained, we used, as it is done in the paper, the Intersection over Union (IoU) metric. This is a widely used metric for assessing the quality of a segmentation model. To calculate IoU metrics using ground truth (GT) and predicted mask (PM), as well as true positives (TP), false positives (FP), and false negatives (FN), we use the following equation:

$$\text{IoU} = \frac{\text{TP}}{\text{TP+FP+FN}}$$

Here, because we are performing a particular case of segmentation, which is interactive segmentation, we used derived metrics from the IoU metric.

The first one is the number of clicks required to achieve a certain IoU threshold. This metric is named NoC_{20}@90 for instance. NoC stands for Number of Clicks. @90 means that the threshold IoU we want to achieve is 90%. Finally $_{20}$ means that the maximum number of clicks allowed is 20. If the IoU achieved on an image after 20 clicks is less than the threshold (here 90%), then we consider that it requires 20 clicks to achieve this value of IoU. Therefore NoC_{20}@90 is a lower bound of NoC@90.

To check the quality of this lower bound, we can look at the metric that gives the percentage of samples in the test sets that haven't reached a certain IoU after a certain number of clicks. This metric is named $\geq$ 20 @85 for instance. @85 means that the threshold used here corresponds to an IoU of 85%. $\geq$ 20 corresponds to the number of clicks. Therefore $\geq$ 20 @85 corresponds to the percentage of the test samples that haven't an IoU greater than 85% after 20 clicks.

Finally, a more simple metric that we can use, and that we use to compare with a non-interactive model, is just the average IoU after a certain number of clicks. This metric is called Avg_{20} IoU in this report. The number corresponds to the number of clicks after which we compute the IoU. Note that for this metric, we also give the standard deviation on the test set inside parenthesis.

Test set The test set is composed of the 5 first big images of each of the locations, which corresponds to 500 cropped images for each of the locations. It is thus composed of 2500 images, as explained before. However, out of these 2500 images, 560 have no building on them. We drop these images for our metrics, as they artificially increase the performance of the model since we get an IoU of 100% without any clicks. We are therefore left with 1940 test images.

Quantitative Results The results obtained are presented in Table 1. We can observe several things. Firstly, what is striking is that ***Finetuning*** clearly gives the best result. The difference between the model trained from scratch and the model given by the authors of the papers but not retrained are pretty important. After 20 clicks, the mean IoU is increased by more than 5 points.

Therefore to get the best results, we need to have a model that is already specialized in segmenting images. But this is not enough, we also

	$\text{NoC@}_{20}80\% \downarrow$	$\text{NoC}_{20}\text{@}90\% \downarrow$	≥ 20 @80% $\downarrow$	Avg_{20} IoU $\uparrow$
No Training	16.22	18.60	0.648	0.718 ($\pm$0.166)
Training from scratch	10.81	18.05	0.333	0.827 ($\pm$0.084)
Finetuning	5.66	14.58	0.135	0.875 ($\pm$0.072)

Table 1: Results obtained

need to specialize this model on the dataset of satellite images. This is easily understandable. Indeed, the model that was trained on classical images has learned to segment classical shapes. The shapes we can see in classical images are different than building seen from space. Also, having a model already specialized in segmenting help, because it doesn't have to learn the task to accomplish, it just has to specialize.

Qualitative Results The figures presenting the results visually are presented in appendix A because they are quite big.

Several things can be observed in these results, but the first obvious one is that we indeed recover the quantitative results observe before. Indeed the Finetune model is consistently better than the model trained from scratch, which is better than the model that hasn't been retrained.

In more detail, looking at the result on Austin (Figure 9) and Kitsap (Figure 12), we can notice that the Finetuned model allows recovering straighter lines and border of buildings. It also recovers finer details. In the Chicago example (Figures 10 and 11), we can notice that the finetuned model is capable of differentiating the buildings individually, which was not the case for the previous model. The model can also find buildings even if no click was made on it. This is a really important point because there can be many buildings on a single image, and we don't want to have to click on each of them. The results on Tyrol (Figure 13) allow us to find that for images that had already good results with the model not retrained, the number of clicks required to achieve the same level of IoU is decreased. Finally, the results on the Vienna (Figure 14) image are a combination of everything we already said. Indeed, in this case, the corners are less round with the finetuned model, all buildings are distinct, and we can recover even tiny buildings, thanks to the clicks.

On the influence of clicks In this paragraph, we want to look at the influence of the clicks, and more generally, we want to be able to quantify the benefits of interactive segmentation compared to classical supervised segmentation. To do so, we need to have a baseline. A ranking of the different methods proposed is available on the dataset website. Looking at two of them [1] and [4] we observe that they achieve similar results on the validation set, which is an IoU of approximately 0.82. Note that we tried to reproduce their results, but they don't provide the weights of their models, and the training of this model takes approximately two weeks, which is not tractable for us.

An interesting curve to look at is the mean IoU over the test set as a function of the number of clicks made. This graph is shown in Figure 7 for all of the three models. The first thing we can notice is that using clicks, and with the

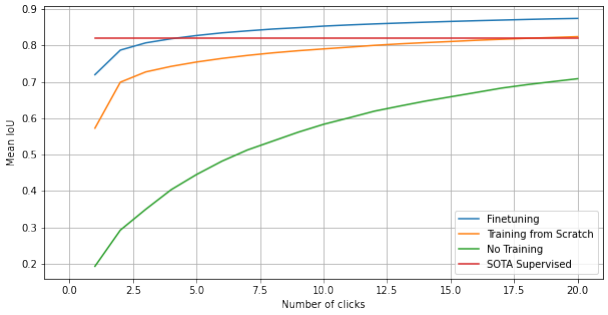


Figure 7: Mean IoU depending on the number of clicks made on each images for each of the three experiments

finetuned model, we get better results than the state-of-the-art results in the simple supervised settings.

Another thing to notice on these curves is that each of the clicks is indeed useful because all of the three curves corresponding to interactive models are increasing. This means that each click indeed improves the segmentation, and is taken into account to compute the masks.

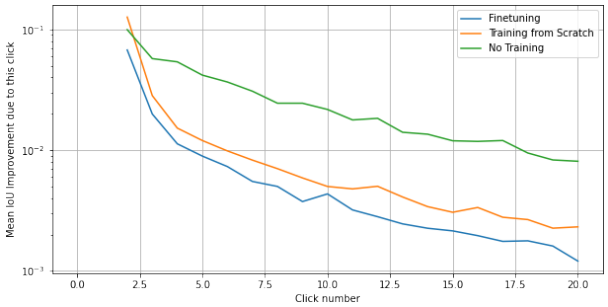


Figure 8: IoU improvement brought by adding a click depending on how many clicks were already made

But we also want to have a look at how much each click improves the results. Figure 8 shows the improvement of the IoU due the n^{th} clicks. On this curve, we clearly see that the first few clicks indeed improve results a lot, while the last ones have a less important impact. We also notice that the more specialized the model is at segmenting satellite images (*Finetuning* is more specialized than *Training from scratch* which is more specialized than *No Training*), the less each clicks brings. This is expected, because a model that is already good, will have less to gain from a click. New clicks will focus on tiny details for models that are already good, while they will focus on more important regions for worse models.

6 Conclusion

As we have seen through this report, using deep interactive segmentation for satellite imagery is a really good idea. It gives better results compared to classical supervised segmentation methods using only a few clicks. It could therefore be used to label large datasets efficiently and very accurately.

Note that one way of using this method would be to re-annotate the dataset we have used. Indeed, as we have seen it, the dataset used is quite poorly labeled. However, training an interactive segmentation model on it works well. We could thus use this model to re-annotate the dataset, removing all false positives and false negatives.

Finally, a way to further improve the performance of this method would be to combine the best supervised segmentation model with interactive segmentation. Indeed, by doing this, we would get the best performance even with no clicks, and the clicks would just improve over the state-of-the-art methods in the classical supervised settings. Moreover, this can be done

quite easily, as it only requires to use the right backbone in the iterative training of the interactive model.

References

[1] Bodhiswatta Chatterjee and Charalambos Poullis. Semantic segmentation from remote sensor data and the exploitation of latent learning for classification of auxiliary tasks. *Computer Vision and Image Understanding*, 210:103251, 2021. 9

[2] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 9:3431, 2015. 2

[3] Emmanuel Maggiori, Yuliya Tarabalka, Guillaume Charpiat, and Pierre Alliez. Can semantic labeling methods generalize to any city? the inria aerial image labeling benchmark. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE, 2017. 5

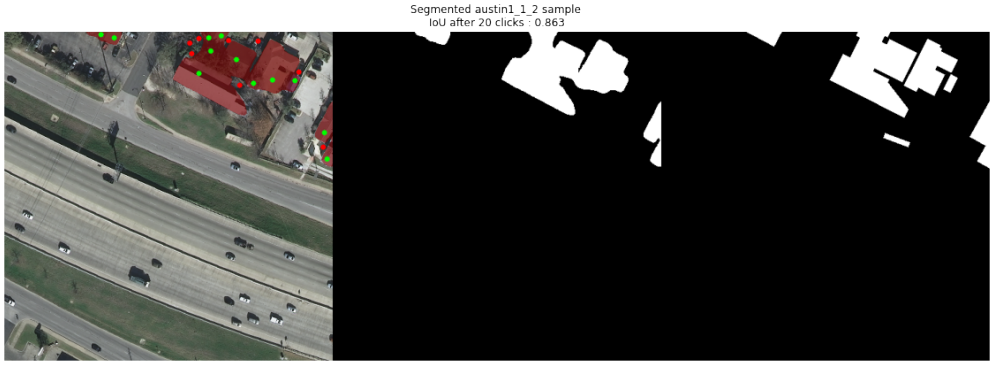
[4] Aleksandar Milosavljević. Automated processing of remote sensing imagery using deep semantic segmentation: A building footprint extraction case. *International Journal of Geo-Information*, 9:486, 08 2020. 9

[5] Shervin Minaee, Yuri Boykov, Fatih Porikli , Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. *CoRR*, abs/2001.05566, 2020. 2, 3

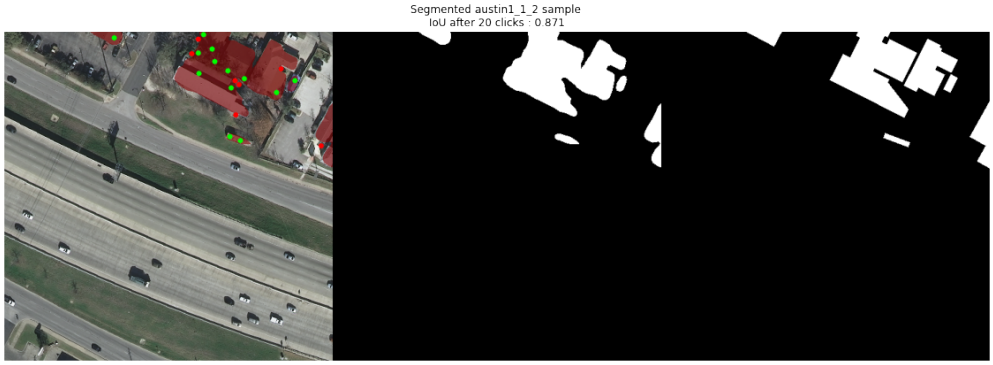
[6] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. *International Conference on Medical image computing and computer-assisted intervention*, 9:234, 2015. 2

[7] Konstantin Sofiiuk, Ilia A. Petrov, and Anton Konushin. Reviving iterative training with mask guidance for interactive segmentation. *CoRR*, abs/2102.06583, 2021. 1, 4, 5

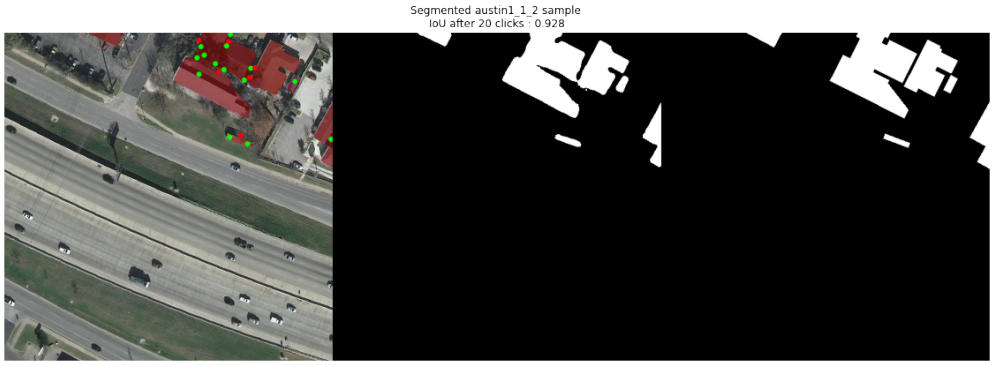
A Qualitative Results - Figures



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



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



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

Figure 9: Results of the differents experiment on image 1 from Austin location (crop number 1-2)

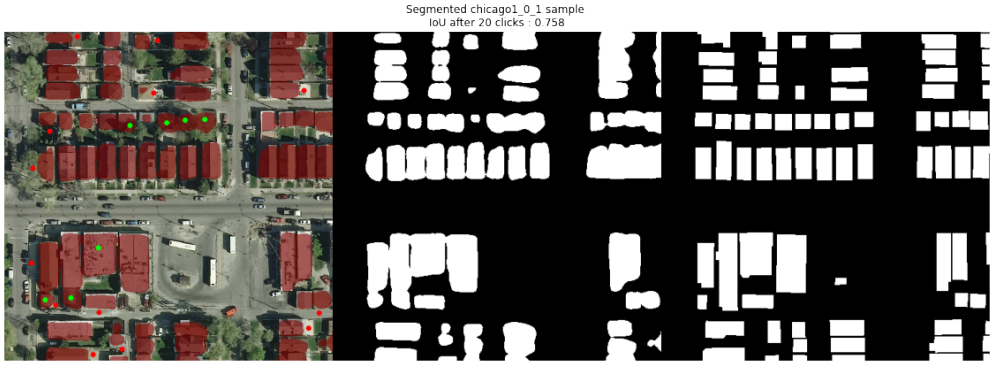
(a) Results of the ***No Training*** experiment

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

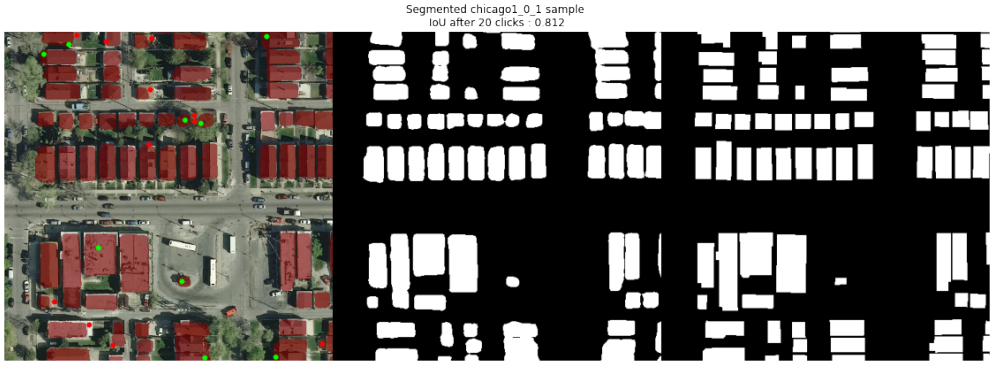
(c) Results of the ***Fintuning*** experiment

Figure 10: Results of the differents experiment on image 1 from Chicago location (crop number 0-0)

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



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



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

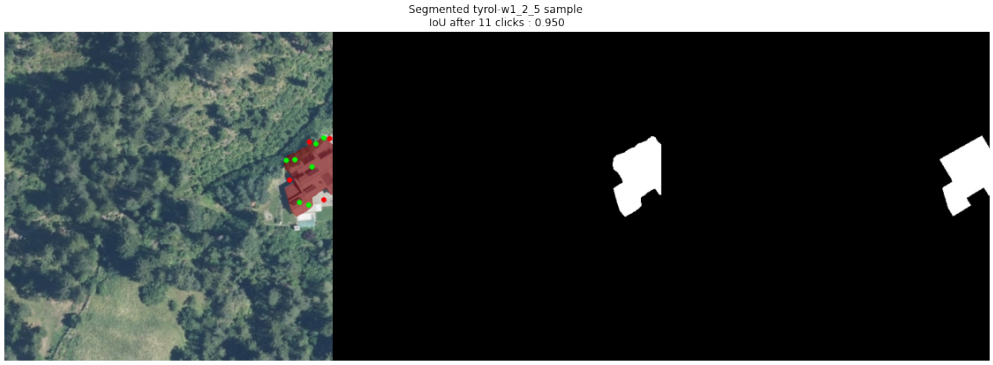
Figure 11: Results of the differents experiment on image 1 from Chicago location (crop number 0-1)

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

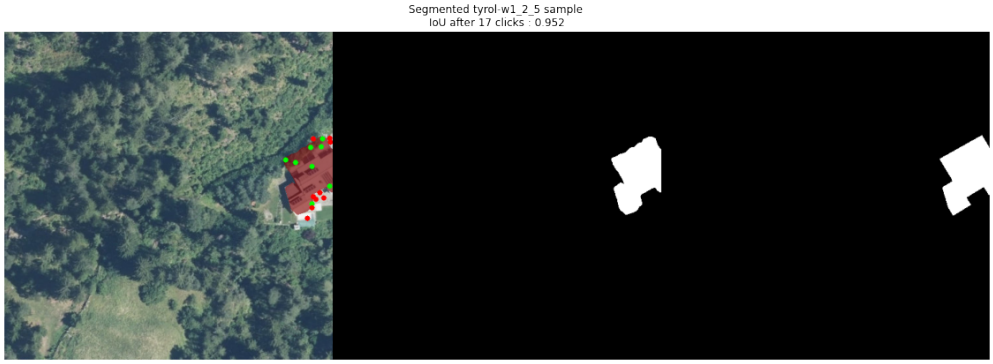
(b) Results of the ***Training from scratch*** experiment

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

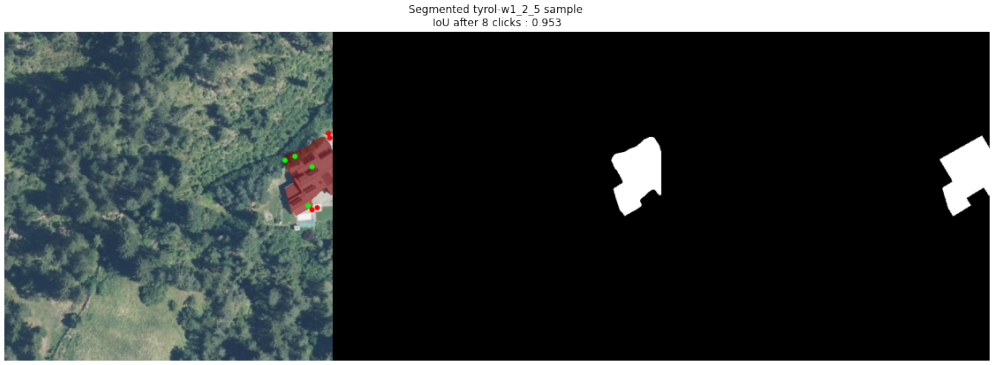
Figure 12: Results of the differents experiment on image 1 from Kitsap location (crop number 2-6)



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



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



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

Figure 13: Results of the differents experiment on image 1 from Tyrol location (crop number 2-5)

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

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

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

Figure 14: Results of the differents experiment on image 1 from Vienna location (crop number 0-0)