AN END-TO-END DEEP LEARNING WORKFLOW FOR BUILDING SEGMENTATION, BOUNDARY REGULARIZATION AND VECTORIZATION OF BUILDING FOOTPRINTS

S. Šanca [1]*, S. Jyhne [2], M. Gazzea [1], R. Arghandeh [1]

[1] Western Norway University of Applied Sciences, Bergen, Norway - (simon.sanca, michele.gazzea, reza.arghandeh)@hvl.no
[2] Norwegian Mapping Authority, Hønefoss, Norway - sander.jyhne@kartverket.no

KEY WORDS: Deep learning, building segmentation, boundary regularization, MapAI, vision transformers, vectorization, QGIS

ABSTRACT:

Automatic building footprint extraction from remote sensing imagery is a widely used method, with deep learning techniques being particularly effective. However, deep learning approaches still require additional post-processing steps due to pixel-wise predictions, that contribute to occluded and geometrically incorrectly segmented buildings. To address this issue, we propose an end-to-end workflow that utilizes binary semantic segmentation, regularization, and vectorization. We implement and assess the performance of four convolutional neural network architectures including U-Net, U-NetFormer, FT-UnetFormer, and DCSwin on the MapAI Precision in Building Segmentation competition. To additionally improve the shape of the predicted buildings we apply regularization on the predictions to assess whether regularization further improves the geometrical shape and improve the prediction accuracy. We aim to produce accurate predictions with regularized boundaries that can prove useful in many cartographic and engineering applications. The regularization and vectorization workflow is further developed into a working QGIS-plugin that can be used to extend the functionality of QGIS. Our aim is to provide an end-to-end workflow for building segmentation, regularization and vectorization.

1. INTRODUCTION

With increasing digitalization and automation, there is a need to develop automatic methods to maintain and update public information stored in spatial databases. Public, building related information is stored in the building register. The building register is the fundamental record for storing information and other relevant data necessary for taxation, public planning and emergency services. Up-to-date building footprint maps are essential for many geospatial applications including disaster management, population estimation, monitoring of urban and impervious areas, 3D city modeling, detection of illegal construction cases (Bakirman et al., 2022), updating topographical databases on a country-wide level and assessing the damage after natural disasters (Takhtkeshha et al., 2023). Although machine learning methods have achieved accurate results in the past in building segmentation, current trends have moved towards the utilization of deep learning for building footprint extraction, that require minimal post-processing after segmentation has been performed. One of the ongoing challenges in building footprint extraction is the accurate recreation of the polygonal boundary of the building footprint either in 2D (Li et al., 2021, Li et al., 2022) or in 3D space (Wang et al., 2021), while at the same time extracting the vectorized building mask as output to be directly used in various GIS software. In the past different approaches have been developed for building extraction from various data sources including satellite, aerial or drone images and the use of LiDAR point clouds. Additionally many different challenges and competitions for building segmentation have been organized and publicly available building datasets have been developed. The most popular ones include the DeepGlobe (Demir et al., 2018), The Wuhan building dataset (Ji et al., 2019), SpaceNet (Etten et al., 2019), CrowdAI (Mohanty et al., 2020) and the most recent MapAI building segmentation dataset (Jyhne et al., 2022). Having different building segmentation competitions with open access to data aids and encourages the development and improvement of methods for accurate building segmentation. However there is still a demand for developing better methods that can extract building footprints in and end-to-end fashion, enabling the user to segment, regularize and vectorize the detected building footprints to make the results applicable within the GIS domain.

Building footprint extraction from remote sensing imagery applying deep learning techniques can be achieved by using either instance segmentation or semantic segmentation, also known as pixel wise labeling (Neupane et al., 2021). Both of these methods have shown great potential and have boosted the performance of building footprint extraction but are lacking the capability to delineate structured building footprints (Zorzi et al., 2021). The extracted features also require further post-processing labour which hinders the applicability and the practical use of the results.

The purpose of our research is to develop an end-to-end workflow for accurate segmentation of building footprints including three major steps: (1) binary semantic segmentation with a CNN, (2) applying building boundary regularization and (3) vectorization. The dataset used for building segmentation is the NORA MapAI: Precision in Building Segmentation dataset (Jyhne et al., 2022). We have developed an implementation for building segmentation using open-source software libraries including Python, PyTorch, the Geospatial Data Abstraction Library (GDAL), QGIS and QtDesigner. Our approach implements the *projectRegularization* repository from (Zorzi and Fraundorfer, 2019, Zorzi et al., 2021) on a semantic segmentation task. The novelty of our approach is applying the regularization task on an entirely new building dataset, while adding our own implementation for the vectorization part. In addition the entire workflow has been developed in an end-to-end manner, that can be applied on different datasets and sets of problems for binary semantic segmentation. Our code can be further de-

* Corresponding author

veloped and improved, users are able to train their own binary semantic segmentation models. Furthermore *projectRegularization* is developed into a QGIS plugin, that can regularize and vectorize any building instance predicted with a CNN that is stored either as a *.tif* or *.png* file.

1.1 Deep learning methods for image segmentation

Deep learning methods for image segmentation can be divided into: (1) semantic segmentation and the more sophisticated (2) instance segmentation. Both methods can be multi-class or binary. In multi-class segmentation different classes of buildings can be segmented, while in binary classification the goal is to extract only the building class from the provided image.

Semantic segmentation is a computer vision task that involves dividing an image into distinct regions and assigning a semantic label to each pixel within those regions. In the case of building segmentation the goal is to distinguish between building and background pixels. Several neural network architectures can be applied for semantic segmentation, including different variations of the U-Net, FCN and SegNet. Recently proposed semantic segmentation architectures include the application of advanced vision transformers for semantic segmentation. GeoSeg[1] is one of the open-source semantic segmentation toolboxes for various image segmentation tasks. The repository has 7 different models, that can be used for either multi-class or binary semantic segmentation tasks, including four vision transformers: U-NetFormer, FT-U-NetFormer, DCSwin, BANet and three regular CNN models: MANet, ABCNet, A2FPN.

The second method that can be applied for building footprint extraction is instance segmentation, which takes a step further in segmenting the building in the image by proposing a bounding box around the detected building and giving each instance of a building a class probability score (Šanca et al., 2021). Instance segmentation can be achieved through a wide variety of methods, which include the region-based approaches such as Mask R-CNN and its predecessors: R-CNN, Fast R-CNN and Faster R-CNN. While the implementation of instance segmentation can be more challenging and computationally heavier, the approach can be more effective in densely populated urban areas, where buildings may be close or overlapping (Zhao et al., 2020).

Both instance and semantic segmentation is trained in a supervised manner using image and ground truth pairs. The resulting segmentation mask is often highly irregular and is not applicable in cartographic applications before it has been vectorized. In many cases, especially when the buildings are occluded by vegetation, shadows, clouds or have different light conditions the predicted segmentation maps can be far different from the real building footprints and need further post-processing steps to be practically applicable in many cartographic and other engineering applications (Zorzi et al., 2021).

1.2 Building boundary regularization methods

Previous attempts at building segmentation used textures, lines, shadows, or more sophisticated and empirically designed methods. However, most of them were not successful at automating and improving the regularization technique of building boundaries. Boundary regularization is a technique used in various computer vision applications to improve the accuracy of image segmentation. Boundaries between different objects can be ambiguous, making it difficult for deep learning models to accurately segment them. In addition, real-world remote sensing images can be noisy, having shadows and different light conditions. Furthermore there is a need for large amounts of training data to achieve accurate segmentation maps with CNNs (Tang et al., 2018). In machine learning, regularization is defined as a method to reduce the generalization error during training (Goodfellow et al., 2016). In the GIS domain regularization or shape-refinement is understood as a normalization process to improve the geometry of the building footprint in a post-processing manner (Zhao et al., 2020). Applying regularization for building segmentation maps constrains the building footprints to be smoother, with clearly defined and straight edges. This makes the building footprint more even if occluded and visually more appealing. In recent studies regularization techniques have been applied by (Zhao et al., 2018). They applied boundary regularization with Mask R-CNN using Minimum Description Length (MDL) optimization. A CNN-based segmentation and empirical polygonal regularization on the Wuhan building dataset using the MA-FCN CNN architecture preprocessed by a boundary extraction algorithm was proposed by (Wei et al., 2020). For the boundary extraction step the Marching Cubes algorithm and for the regularization the Douglas-Peucker algorithm has been used. In their study coarse- and fine adjustment techniques were applied to improve the geometry of the building footprints. In order to achieve higher prediction accuracy (Zhao et al., 2020) developed a new instance segmentation workflow called Hybrid Task Cascade (HTC) as a baseline model for building detection and segmentation. They integrated the Convex hull and Douglas Peucker algorithms for regularization, to obtain accurate building segmentation maps. Their method was tested on the CrowdAI dataset. In contrary Zorzi et al., (2021) approached the problem differently, they trained an unsupervised GAN regularization network using adversarial, potts and normalized cut losses to ingrain knowledge about building boundaries into the neural network. Their implementation was tested with instance segmentation, applying the Mask R-CNN architecture for building segmentation and comparing it with a R2U-Net semantic segmentation architecture. Their implementation is publicly available as *projectRegularization*[2]. Because their implementation has open access, is straightforward to implement and can be used for both semantic- and instance segmentation tasks we have chosen to test it and incorporate it into our end-to-end workflow for the MapAI dataset.

1.3 The MapAI dataset

The proposed end-to-end workflow has been tested and evaluated on the MapAI: Precision in Building Segmentation competition dataset. The competition was arranged by the Norwegian Artificial Intelligence Research Consortium (NORA) in collaboration with Center for Artificial Intelligence Research at the University of Agder (CAIR), the Norwegian Mapping Authority, AI:Hub, Norkart, and The Danish Agency for Data Supply and Infrastructure. The dataset provides data sources for segmentation of buildings using aerial images and LiDAR data. The dataset is split into training, validation and two test sets with image shapes of 500x500 and resolution of 0.25 m. The training dataset consists of several different locations in Denmark, while the test dataset consists of seven locations in Norway, including urban areas: Bergen, Kristiansand, Oslo, Stavanger, Tromsø and a rural area: Rana. The dataset includes

[1] https://github.com/WangLibo1995/GeoSeg

[2] https://github.com/zorzi-s/projectRegularization

a wide variety of buildings with different sizes, shapes and complexities, this ensures a diverse dataset with different environments and building types (Jyhne et al., 2022).

There are two test sets divided into task 1 and task 2 to evaluate the accuracy of the trained models. The test set for task 1 is used for testing the segmentation approach using only aerial images as data source, while the test set for task 2 is used to test the combined approach using aerial and LiDAR images. In total there are: 7000 instances of buildings in the training set, 1500 instances of buildings in the validation set, 1369 instances of images in the task 1 test set, 978 instance of images in the task 2 test set. The dataset can be dowloaded from HuggingFace[3]. Figure 1 shows an example from the the training dataset.

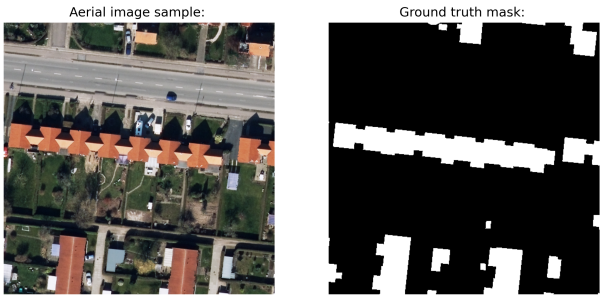

Figure 1. An aerial training sample from the MapAI dataset

2. METHODS

We provide an end-to-end workflow for building extraction, while also improving the predicted building footprints by boundary regularization. Our workflow consists of three steps, that are merged together end-to-end:

1. First, we utilize four convolutional neural network architectures to train binary semantic segmentation models on the MapAI dataset and make predictions on the test set 1.

2. Second, we apply *projectRegularization* proposed by (Zorzi and Fraundorfer, 2019, Zorzi et al., 2021) to regularize the predicted building footprints and improve their geometry.

3. In the final step we perform the vectorization process converting the regularized building masks to polygons ready to be used in any GIS-environment.

Steps (2) and (3) are implemented into our developed QGIS-plugin. Our workflow was developed in Python, using the PyTorch library for the application and development of deep learning models. We used GDAL (Geospatial Data Abstraction Library) to vectorize the predictions in step 3. QtDesigner and QGIS have been used to develop and test the plugin. Each step of our workflow is further described in the following subsections. The complete workflow for model training, prediction and regularization is presented on figure 3.

2.1 Dataset preparation

The MapAI dataset was downloaded from Huggingface and saved locally as a cached Parquet file, which can be accessed with the PyTorch *DataLoader* library. Since the dataset contains some mislabeled images in the training and validation sets we have removed them according to previous work by (Kaliyugarasan and Lundervolt, 2023). The names of the images from the training and validation sets are stored inside two text files in our repository. We provide simple bash scripts for their removal from the original dataset.

2.2 Semantic segmentation with CNNs

The initial stage of our methodology involves identifying and delineating the boundaries of buildings depicted from aerial images. We have decided to apply the basic U-Net neural network architecture and three vision transformers including U-Net-Former, FT-UNet-Former and DCSwin.

2.2.1 Model training. U-Net, proposed by (Ronneberger et al., 2015) has been successfully applied in the past for various image segmentation tasks both in the medical and remote sensing domain. The following three architectures are vision transformers (ViT). In a ViT the input image is divided into a sequence of patches, which are flattened and fed into the transformer encoder network. The network consists of a stack of self-attention layers, which enable the network to target different parts of the image when making predictions (Dosovitskiy et al., 2021). The key idea behind a vision transformer is to use a multi-scale hierarchical approach for image segmentation, where low-level transformers process raw images and high-level transformers operate on down-sampled images. This approach enables to capture information on different scales and preserve rich contextual information. In contrary, traditional CNNs gradually decrease the spatial resolution of an image, which leads to loss of detail (Liu et al., 2021). The second neural network we have applied is the U-NetFormer (Petit et al., 2021), which is a unified network consisting of two architectures: a 3D Swin Transformer based encoder network and transformer based decoder network, that allows higher accuracy and lesser computational cost during training. The CNN architecture integrates skip connections between the encoder and decoder network. This enables the use of deep supervision, that can help to mitigate the vanishing gradient problem, improve the overall stability of the training process and enable more accurate and efficient learning (Wang et al., 2022). The third applied model is FT-Unet-Former, which is a fully transformer-based network architecture, without any additional recurrent or convolutional layers, meaning that the model only uses self-attention and feed-forward layers to process the input sequence, making it highly parallelizable and computationally efficient. The final neural network that we applied is the DCSwin. It is a hierarchical vision transformer using a shifted window approach proposed by (Liu et al., 2021).

We trained four binary semantic segmentation models on the MapAI dataset using the hyperparameters listed in Table 1. The training was performed locally with CUDA 11.7 on an NVIDIA GeForce RTX 3070 graphics card with 8 GB of memory. The trained models were saved as *.pth* PyTorch files. A *.pth* file is a binary file that stores the weights and biases of a trained PyTorch model. Predictions were performed on test set 1 on 1369 images.

[3] https://huggingface.co/datasets/sjyhne/mapai_dataset

Hyperparameters used	Value
Input image size	512 x 512
Batch size	2
Learning rate	0.0001
Optimizer	Adam
Number of epochs	25
Weight decay	0.001
Decode channels	[64, 128, 256]
Dropout rate	0.5

Table 1. Hyperparameters used during training

We used the Adam optimizer with Binary Cross Entropy Loss with logits during training to measure the difference between the predicted output and the ground truth. The loss function is defined as:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(\sigma(z_i)) + (1-y_i)\log(1-\sigma(z_i))\right] \quad (1)$$

where N is the batch size, y_i is the ground truth image for sample i, z_i is the logit output of the model for sample i and σ is the Sigmoid function.

2.3 Applying regularization on predictions

Once the predictions are generated using the trained models, regularization is applied as a post-processing step to further improve the geometry and the accuracy of the predicted building masks. Since pixel-based classification leads to results with rounded corners and occluded edges on the predictions, regularization is a crucial step to further improve the predictions. Implementing *projectRegularization* is straightforward, some parts of the code needed to be changed in order to choose between between the segmentation tasks. Since we have worked with semantic segmentation architectures, we have chosen this option and changed the code accordingly. *ProjectRegularization* is written in PyTorch and can use both *.png* and *.tif* images as input. It applies a GAN (Generative Adversarial Network) composed by two different neural networks. The (1) generator creates a regularized building footprint from the predicted building mask and the (2) discriminator examines if the generated building footprint is real or fake. The generator and the discriminator work together in a competitive and collaborative manner to produce the final output, which is the regularized building footprint, with improved geometrical shape.

The steps to calculate the GAN objective function are summarized from (Zorzi and Fraundorfer, 2019, Zorzi et al., 2021):

The regularization learning process - $L(G, R, D)$

1. The generator $G(x, y)$ learns the mapping function from the segmented building footprints - X and the ideal building footprints from the training set - Y.
2. The intensity images Z are exploited from the dataset.
3. Regularization is performed $G : (X, Z) \rightarrow Y$.
4. The regularized building footprints are produced by the encoder E_G and the residual decoder F.
5. The discriminator D estimates whether the regularized images are ideal.

The final and full objective function to jointly train the generator path G and the reconstruction path R is a linear combination between the adversarial, regularized and the reconstruction losses, expressed as:

$$\begin{aligned} L(G,R,D) = \alpha L_{GAN}(G,R,D) + \\ \beta L_{rec_G}(G) + \gamma L_{rec_{R(R)}} + \\ \delta L_{Potts}(G) + \epsilon L_{rec_{ncut}}(G) \end{aligned} \quad (2)$$

The above are created by connecting the encoders E_R and E_G to the residual decoder F for each iteration. The final, regularized building mask is generated after E_G, E_R and F are jointly updated.

2.4 Performing the vectorization with GDAL

The vectorization part is straightforward. First the regularized image as *.tif* or *.png* is read using GDAL drivers and opened. Next the raster band is acquired from the image and the appropriate driver to be used for the vector file needs to be defined. GDAL supports a vast amount of vector drivers[4]. We choose GeoPackage, mostly because it is an open, standards-based, platform-independent, and portable format. After the vector driver is defined the pixel values from the image are saved as a column in the attribute table, and the vectorized geometry of the building footprint is saved as a polygon. Since the predicted and the regularized building masks have only two pixel values, where pixel values 255 represent the buildings and pixel values 0 represent the non-building information, they need to be removed after the vectorization. To vectorize the image, the GDAL Polygonize function was used. In the final step the Extract by Attribute tool was used to keep the buidling polygons with pixel values 255. Figure 2 shows the vectorization process.

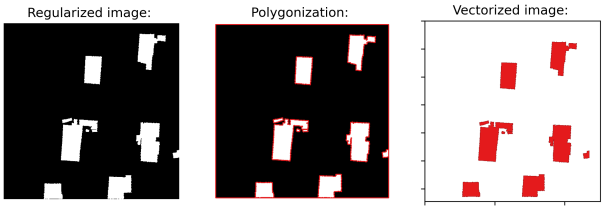

Figure 2. Performed vectorization with GDAL.

3. RESULTS AND DISCUSSION

3.1 Evaluation metrics

The performance of our developed workflow applying *projectRegularization* is evaluated based on the metrics proposed in the MapAI: Precision in Building Segmentation challenge (Jyhne et al., 2022). Intersection-over-Union (IoU) or the Jaccard index, is the ratio of the intersection area of the predicted and ground truth mask to their union:

$$IoU = \frac{Intersection}{Union} = \frac{|G \cap P|}{|G| + |P| - |G \cap P|} \quad (3)$$

where G is the ground truth mask and P is the prediction. Boundary Intersection-over-Union (BIoU) calculates the IoU of the boundary of the prediction and ground-truth:

$$BIoU = \frac{Area(|G_d \cap G|) \cap (|P_D \cap P|)}{Area(|G_d \cap G|) \cup (|P_D \cap P|)} \quad (4)$$

[4] https://gdal.org/drivers/vector/index.html

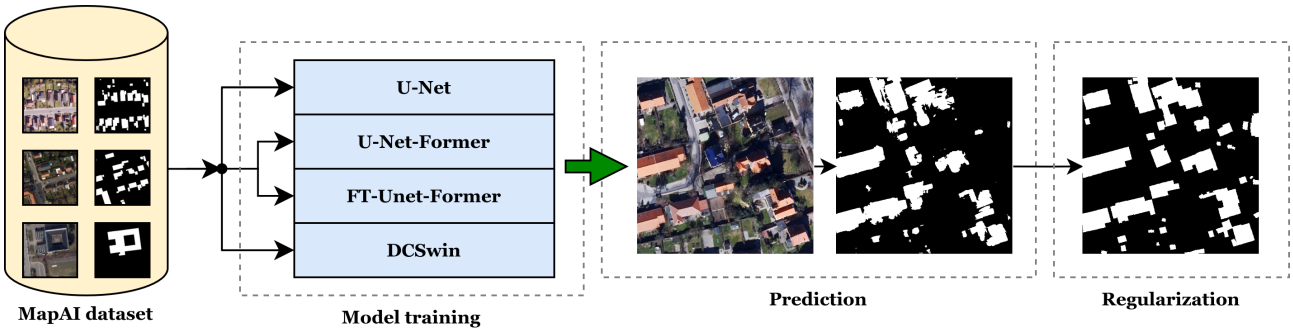

Figure 3. Our end-to-end workflow for building segmentation and regularization.

where G and G_d denote the ground-truth and the edge of the ground truth with thickness d. Similarly to G, P and P_d the predicted mask and the edge of the predicted mask with thickness d. To evaluate the submissions for the MapAI competition, the final score is a combination of Intersection-over-Union (IoU) and Boundary Intersection-over-Union as noted below.

$$S = \frac{BIoU + IoU}{2} \tag{5}$$

We provide the metrics for the predictions using our trained models and for the regularizations separately in order to compare the difference and assess whether regularization improves the predicted building footprints or not.

Model	IoU	BIoU	S
U-Net	37.93	34.44	36.19
U-Net-Former	39.48	35.15	37.32
FT-U-Net-Former	39.95	37.66	38.81
DCSwin	45.19	40.74	42.96

Table 2. Predictions without regularization.

Model	IoU	BIoU	S
U-Net	38.87	35.05	36.96
U-Net-Former	40.07	35.50	37.78
FT-U-Net-Former	40.17	37.80	38.98
DCSwin	45.64	41.04	43.34

Table 3. Predictions with regularization.

Our lowest performing model was the simple U-Net achieving a 37.93 IoU without and 38.87 IoU with regularization. Its extended transformer architectures U-Net-Former and FT-Unet-Former performed better. FT-Unet-Former was slightly better than U-Net-Former, achieving 39.95 IoU without regularization and 40.17 IoU with regularization. The reason for its improved performance is the fully-transformer based architecture. The best performing model as expected was the DCSwin model achieving 45.19 IoU without regularization and 45.64 IoU with regularization. The reason for its improved performance is the shifted window approach for hierarchical feature representation. The Swin Transformer divides the input image into smaller patches and processes them hierarchically in a series of stages, each of these stages operate at different spatial resolution and are better at feature extraction, which improves the final segmentation accuracy. The results show, that applying regularization slightly improved the performance of our models by a small margin, on average around 0.5 % depending on the test image. We applied regularization on a wide variety of predictions. In cases where the prediction is of poor quality, the regularization will be the same. In contrary, the tested regularization method can help to improve the geometry of the buildings, but cannot be used to significantly improve the prediction accuracy. Although we did not use data augmentation techniques to further improve our results, we can conclude that data augmentation is a necessary step to improve the prediction accuracy, especially on the test images for Tromsø, where many of the buildings have shadows and are low-contrast images. The next step would be to apply transfer learning to further improve our results and perform the combined aerial-LiDAR segmentation task.

3.2 Developed QGIS plugin

Our developed QGIS plugin, that can be used to regularize any binary semantic segmentation image is presented on figure 4. The user can choose between two options: (1) regularization option, which will regularize and further improve the prediction and the (2) vectorization option, that enables the user to vectorize any predicted or already regularized building footprint from a raster format to a vector format. The graphical user interface for the developed plugin is simple. On the top, the user provides the path for the raster file, that will be regularized or vectorized. The loaded raster is shown in the middle. The two checkboxes can be used to choose which process will be executed. The *Restore Defaults* resets the plugin interface and removes any stored data. Additional instructions on how to use the plugin can be found by clicking the *Help* button. Both *Regularize building footprint* and *Vectorize building footprint* options automatically save the generated file. The regularization option will save the file in the same folder where the original raster file for regularization is located. It adds the prefix *reg-* and uses the same image type as the original. After the regularization, the checkbox automatically changes to the option *Vectorize building footprint*, which can be used to save the regularized image or even just the prediction as a vector file. If the user runs the plugin once more the regularization raster is automatically converted into a polygon and is automatically saved as a GeoPackage in the corresponding folder. The development of our proposed plugin can be followed online, accessing its GitHub repository[5]. We encourage everyone to test the plugin, provide feedback, new ideas, suggest improvements and contribute to further development.

[5] https://github.com/s1m0nS/QGIS-Regularize-Building-Footprints

Figure 4. Visual comparison of predictions and regularizatons for our trained models.

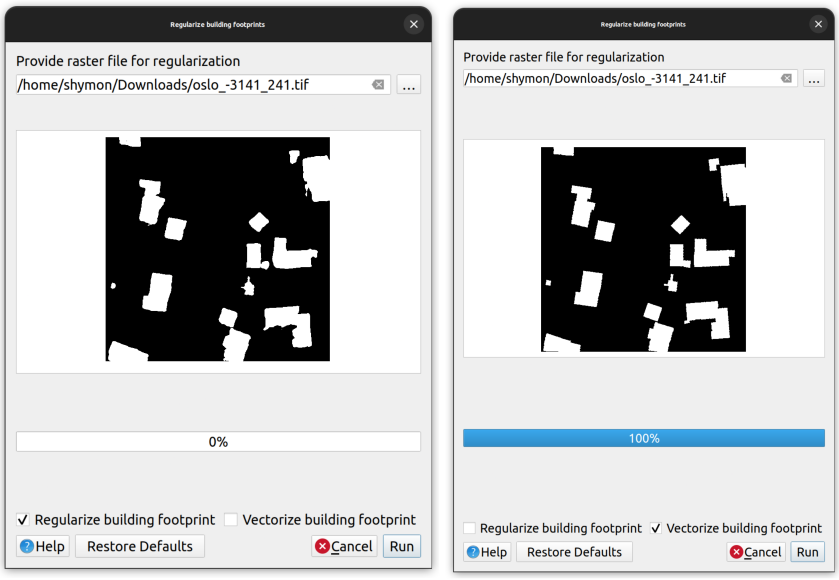

Figure 5. The GUI of our developed QGIS plugin and the options available to the user.

4. CONCLUSION

The main purpose of our study was to develop an end-to-end workflow for building footprint segmentation, apply regularization and vectorization on the results in order to provide a GIS-ready solution. We conclude that *projectRegularization* additionally improves the segmentation accuracy by an average value of 0.55 IoU, 0.35 in BIoU and 0.44 in S metric. Regularization not only improves the predictions, but also improves the geometrical shape of the building footprints. Furthermore the vectorization part contributed to the practical aspect of combining deep learning models and open-source GIS software. Our QGIS-plugin can be used to regularize buildings from predictions and convert them to vector files, which can be help in areas where practical application is of outmost importance. Our workflow is accessible online on GitHub: https://github.com/s1m0nS/mapAI-regularization and tested. We provide Jupyter Notebooks for easier work management with explanations. The development of our QGIS-plugin can be followed on GitHub: https://github.com/s1m0nS/QGIS-Regularize-Building-Footprints. We encourage everyone to try out our QGIS plugin and provide feedback, or contribute to the code repository.

REFERENCES

Bakirman, T., Komurcu, I., Sertel, E., 2022. Comparative analysis of deep learning based building extraction methods with the new VHR Istanbul dataset. *Expert Systems with Applications*, 202, 117346.

Demir, I., Koperski, K., Lindenbaum, D., Pang, G., Huang, J., Basu, S., Hughes, F., Tuia, D., Raskar, R., 2018. DeepGlobe 2018: A challenge to parse the earth through satellite images. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, IEEE.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N., 2021. An image is worth 16x16 words: Transformers for image recognition at scale.

Etten, A. V., Lindenbaum, D., Bacastow, T. M., 2019. Spacenet: A remote sensing dataset and challenge series.

Goodfellow, I., Bengio, Y., Courville, A., 2016. *Deep Learning*. MIT Press.

Ji, S., Wei, S., Lu, M., 2019. Fully Convolutional Networks for Multisource Building Extraction From an Open Aerial and Satellite Imagery Data Set. *IEEE Transactions on Geoscience and Remote Sensing*, 57(1), 574-586.

Jyhne, S., Goodwin, M., Andersen, P. A., Oveland, I., Salveson Nossum, A., Ormseth, K., Ørstavik, M., Flatman, A., 2022. MapAI: Precision in Building Segmentation - Nordic Machine Intelligence.

Kaliyugarasan, S., Lundervolt, A. S., 2023. LiDAR and aerial image-based building segmentation using U-Nets. *Nordic Machine Intelligence*, 2, 23-25.

Li, Q., Zorzi, S., Shi, Y., Fraundorfer, F., Zhu, X. X., 2021. End-to-end semantic segmentation and boundary regularization of buildings from satellite imagery. *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, 2508–2511.

Li, Q., Zorzi, S., Shi, Y., Fraundorfer, F., Zhu, X. X., 2022. RegGAN: An End-to-End Network for Building Footprint Generation with Boundary Regularization. *Remote Sensing*, 14(8).

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B., 2021. Swin transformer: Hierarchical vision transformer using shifted windows.

Mohanty, S. P., Czakon, J., Kaczmarek, K. A., Pyskir, A., Tarasiewicz, P., Kunwar, S., Rohrbach, J., Luo, D., Prasad, M., Fleer, S. et al., 2020. Deep Learning for Understanding Satellite Imagery: An Experimental Survey. *Frontiers in Artificial Intelligence*, 3.

Neupane, B., Horanont, T., Aryal, J., 2021. Deep Learning-Based Semantic Segmentation of Urban Features in Satellite Images: A Review and Meta-Analysis. 13(4), 808.

Petit, O., Thome, N., Rambour, C., Soler, L., 2021. U-net transformer: Self and cross attention for medical image segmentation.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 9351, 234–241.

Takhtkeshha, N., Mohammadzadeh, A., Salehi, B., 2023. A Rapid Self-Supervised Deep-Learning-Based Method for Post-Earthquake Damage Detection Using UAV Data (Case Study: Sarpol-e Zahab, Iran). *Remote Sensing*, 15(1).

Tang, M., Perazzi, F., Djelouah, A., Ayed, I. B., Schroers, C., Boykov, Y., 2018. On regularized losses for weakly-supervised CNN segmentation.

Wang, L., Li, R., Zhang, C., Fang, S., Duan, C., Meng, X., Atkinson, P. M., 2022. UNetFormer: A UNet-like Transformer for Efficient Semantic Segmentation of Remote Sensing Urban Scene Imagery. 190, 196–214.

Wang, Y., Zorzi, S., Bittner, K., 2021. Machine-learned 3d building vectorization from satellite imagery.

Wei, S., Ji, S., Lu, M., 2020. Toward Automatic Building Footprint Delineation From Aerial Images Using CNN and Regularization. 58(3), 2178–2189.

Zhao, K., Kamran, M., Sohn, G., 2020. Boundary regularized building footprint extraction from satellite images using deep neural network.

Zhao, K., Kang, J., Jung, J., Sohn, G., 2018. Building extraction from satellite images using mask r-CNN with building boundary regularization. IEEE, 242–2424.

Zorzi, S., Bittner, K., Fraundorfer, F., 2021. Machine-learned regularization and polygonization of building segmentation masks. *2020 25th International Conference on Pattern Recognition (ICPR)*, IEEE, 3098–3105.

Zorzi, S., Fraundorfer, F., 2019. Regularization of building boundaries in satellite images using adversarial and regularized losses. *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*, IEEE, 5140–5143.

Šanca, S., Mangafić, A., Oštir, K., 2021. Building detection with convolutional networks trained with transfer learning. *Geodetski vestnik*, 559-593.