Extracting Masonry Building Facades through Polygon Image Segmentation

Bryan Pantoja*, Vinitra Swamy*, Marija Sakota*
Supervised by Katrin Beyer, Earthquake Engeeniering and Structural Dynamics Laboratory
EPFL, Lausanne, Switzerland
** denotes equal contribution*

***Abstract*—3D modeling of buildings is an important task for various fields such as urban planning, architecture and engineering analysis. This has led to development of many techniques for automatic generation of building models. Among them, we are focusing on a computer vision segmentation approach to group pixels into facades of buildings. We compare two types of approaches to this problem. The first one relies on facade segmentation to mask the relevant portion of the image, while the second detects the corners of the facade which are then connected to a polygon. We show that for the case of building facade segmentation, our approach using a convolutional neural network to generate polygons has the potential to give better results than standard segmentation.**

I. Introduction

The evolution of deep learning methodologies and development of computer machines has made possible the progress in numerous research fields such as computer vision. Image processing is one of the benefited areas in special through the use of Convolutional Neural Networks (CNN) in which tasks such as classification, detection or segmentation are performed using image data sets. In these models, a sequence of convolutional operations are applied to images extracting features through an optimization process taking into account the spatial distribution of pixels in the data [1].

On the other hand, the necessity of 3D modeling of buildings for various applications such as urban planning, architecture development or engineering analysis, has motivated the investigations of techniques for an automatic generation [2]. Among these techniques, the use of image data and multiple view geometry (and its products, e.g., point clouds, meshes) has been a constant. Then, when it comes to three-dimensional modeling of buildings using image data, it is necessary to have a good understanding of their contents [3]. Therefore, the use of segmentation techniques to classify and to group image pixels into objects such as a facade, windows, and doors is imperative.

In this work, our purpose is to contribute part of the process of 3D building modeling implementing a deep learning methodology for facade segmentation. We compare two approaches - facade segmentation and corner detection. For the first approach we use a CNN to output the facade mask directly. For the second, the framework, which uses a CNN, was trained to take as input an image of a free-standing building and produce as output corners of the facade of the building. Later, corners are ordered and used as vertices to draw a polygon that represents the final building facade contours. Since we are working with only buildings with regular shapes (masonry buildings), we opted for a simpler approach for ordering the vertices, rather than using recursive neural networks (RNN) to connect the vertices. We compare our approach to PolyRNN++, which is trained on the cityscapes dataset, and does use an RNN to order the vertices [4]. The model is tested and evaluated using dice score comparing a mask generated with the output polygon and the ground truth. Both CNNs are based on a U-Net architecture [5].

II. Related work

Semantic segmentation consists in the assignment of labels to each image pixels as an object or background. With this goal, [6] proposed a Fully Convolutional Network (FCN) composed by two convolutional parts (encoder and decoder) to produce image masks of segmented objects from an image. A relevant work using these types of models is presented in [5]. In this, a model, namely U-Net, was proposed to apply data augmentation techniques to overcome the problems of limited data. Based on this model, in [7] the authors demonstrate that the use of pre-trained encoders improved the results using as application for the segmentation of buildings from aerial images. Fig. 1 shows the architecture proposed in [7].

Aiming for the understanding of buildings and facades, various studies have been performed to segment facades in the different objects that compose them. Among those studies, the use of machine learning techniques, specifically randomized forest, and grammar shapes are used in [3]. In this a facade is decomposed in simpler objects which combined using rules and restrictions can rebuild the building layout. The use of CNN models for facade segmentation and interpretations was introduced in [8] in which the authors segmented the facade in three classes (i.e., walls, windows, doors), demonstrating that the use of transfer learning leads to the use of small data sets for training. Authors in [9] presented a framework composed of three CNN to segment facade elements exploiting their

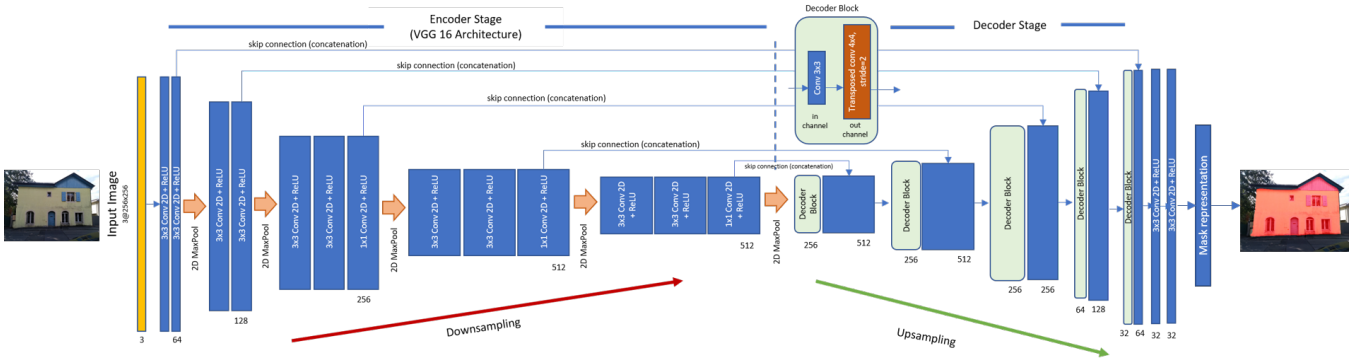

Figure 1. Architecture of our U-Net16 model to generate filled polygon masks. U-Net11 is similar to U-Net 16, except that it uses pretrained VGG11 as an encoder instead of VGG16 and a few of the conv layer dimensions are different [7]

unique characteristics (e.g., windows rectangular shapes). A most recent study can be found in [10] where the potential symmetry in the windows and doors is taken into account in the formulation of the loss function of a CNN model. A variation to CNN and facade grammars is found in [11] where it is presented a model based on atrous large kernel which is able to overcome difficulties such as image regions with occlusions or ambiguities.

Sometimes the representation required as output is not just a mask of the segmented building objects. Authors from [12] presents an interesting work in which buildings and roads are segmented from aerial images in a vector representation using CNN and RNN models. Although we are not using RNN, our work is inspired by the building segmentation presented in that study, in which buildings are segmented as polygons (vertices and connection sequence). Instead of using aerial images, our data set is composed by images taken from the ground level of free-standing building having as a goal the segmentation of the facade as a polygon.

III. Methodology

We compare two approaches for facade segmentation. The first approach is a classic segmentation approach where the output of the model is a facade mask. The second approach detects the corners of the building facade from the image and the connects them to a polygon, representing the contours of the building facade. The reason why we wanted to try with polygons for this task is because most buildings from our dataset have a regular polygonal shape. Traditional segmentation methods are not able to generate sharp edges so polygons were a logical choice. Both of the approaches rely on CNNs, in particular U-Net based encoder-decoder architectures presented in [7], named U-Net16 and U-Net11.

For the first approach, we are compare results that we get with U-Net16 and U-Net11 CNN models. These models are composed from the encoder and the decoder, where the encoder is VGG16 or VGG11 network, respectively, pretrained on ImageNet dataset. The decoder is composed of deconvolutional layers where each layer takes as the input previous layer output and corresponding encoder layer output. The encoder is supposed to capture general characteristics of the image and give them as a feature mask (Figure 1). The input for these models are processed RGB images of buildings, while the target is a building facade mask. U-Net16 showed better results than U-Net11 for this specific task.

To detect vertices, we first try with the same U-Net architecture. The only difference is the target which in this case is a mask of vertices of the polygon. We again tried two types of U-Net architecture. In this case U-Net11 generated better results, but none of the two managed to detect all the vertices in the image.

Since this basic model did not manage to detect corners correctly, we tried with a modified U-Net architecture. Because U-Net11 worked better with vertices in previous model, we decided to use it for this one as well. Instead of generating only the vertices mask, this model outputs the facade mask along with it. Facade mask is a mask that represents the whole building facade. The only difference in the architecture is last layer having 2 output channels, one for the facade and the other one for the corners, instead of 1 channel as the original model. The idea behind this modification is to make the detection of corners easier by directing the model to learn features that correspond to the building facade mask as well.

While this model was able to generate a satisfactory facade mask, it did not perform well on the vertex detection task. We decided to try the same architecture of the model (VGG11 + U-Net11) with a different mask. This time we used a boundary mask instead of facade mask in second channel. The boundary mask represents the contours (outline) of the facade mask. This modification improved the results and this model is able to detect all the corners

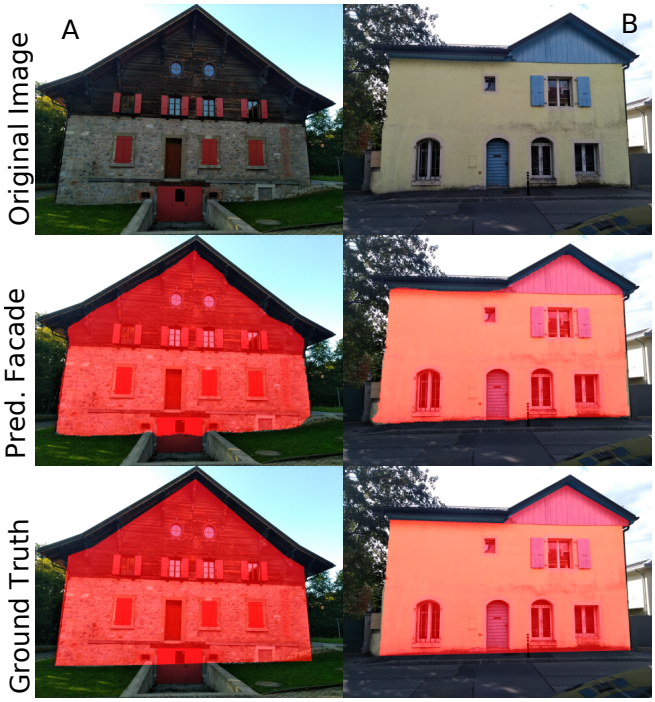

Figure 2. Predictions model 1. Semantic facade segmentation.

of the polygon of the building facade.

Instead of modifying the last layer, we also tried to add another convolutional block after the U-Net model layers and have the U-Net output either facade or boundary mask while the additional block outputs vertices. Even though we were hoping for this model to produce even better results than the previous, taking into consideration additional layers that could learn mappings from facade mask to vertices, it did not manage to do that.

To extract a polygon from the vertices mask, we first find regions that represent vertices and obtain their properties to be able to find their exact coordinates. Coordinates are found as centroids of each vertex region in the mask. These coordinates are then ordered in the way they would be connected in a polygon. This is done by finding the centroid of the polygon from the vertices and then ordering them by the polar angles. Other methods, such as PolyRNN++ from the University of Toronto, use an RNN to determine the order of the vertices for polygon segmentation [4]. The PolyRNN model is used as a comparative baseline against our CNN vertex-polygon approach.

A. Implementation details

As mentioned in the previous section, base model for all our models is U-Net, more precisely two different variations of it - U-Net16 and U-Net11 [7]. Architecture of these models is shown in the Figure 1. The code was written using the high performance deep learning library

Figure 3. Predictions model 2. Semantic segmentation of facade vertices.

PyTorch [13]. All of the models take as input images that have shape $(256, 256, 3)$. Output masks also have these dimensions. All models are trained using dice loss.

For the training of all the models we use Adam optimizer [14] with batch size 8, learning rate $1e-5$, and default β_1 and β_2. Learning rate is decreased by factor 0.5 when learning stagnates for 200 epochs. Models are trained for 200 epochs on NVIDIA GeForce RTX 2080Ti GPU. Training time for all the models is around 8 hours. During the inference, we use batch size 16.

To find regions of vertices in corner mask, we use *regionprops* function from scikit-image library. As coordinates of each vertex, we choose to take a centroid of each region that we have. After ordering vertices in a way they need to be connected, we use OpenCV to draw a polygon from them.

IV. Experiments

A. Dataset and evaluation metrics

We collected our data from ground level images of buildings. Our training dataset contains 270 images, while test contains 12. The size of images vary depending on the camera source. 20% of the training dataset is used for validation, while the rest is used for training.

Images were annotated using VGG Image Annotator (VIA) by marking the polygons that represent the building facade [15]. For our experiments we extracted facade masks as filled-in polygons, boundary masks which represent contours of the polygon and corners of the polygon with VIA output. All images are resized to 256×256 dimension.

Models were evaluated using one of the most common metrics for image segmentation - dice score. This score is used to estimate the similarity between two samples. It is

Building	U-Net 16 - Mask	U-Net 11 - Polygon	PolyRNN++
A	0.9821	0.9647	0.7775
B	0.9831	0.9606	0.7743

Table I
DICE SCORES FOR U-NET16, U-NET11, AND POLYRNN++.

defined as

$$D = \frac{2|a \cdot b|}{|a|^2 + |b|^2},$$

where a and b are binary vectors. Score can range between zero and one, where bigger score is better. Because we have limited number of samples in the dataset, manual evaluation has also been conducted.

B. Results

Figures 2, 3, and 4 depict a visual representation of the trained models (U-Net16 to predict facade masks, U-Net11 to predict vertices, U-Net11 with two prediction channels). From the images that represent the output of the first model (Figure 2), we can see that it accurately captures the model, but struggles to output straight edges. When it comes to the second model (Figure 3), it correctly detects corners, just does not manage to detect all of them. The third model (Figure 4) successfully identifies corners and produces reasonably sharp edges. Figure 5 compares the polygons produced from the third model to the modified baseline of PolyRNN++, where it is clear that the U-Net11 model covers more area of the building facade.

In Table IV-B we show dice scores for the first and third model on testing dataset. To make them comparable, the polygon obtained by the third model is simply filled in to generate a facade mask. Although the polygon model gives slightly lower dice score, it is still a remarkable result. Regardless of the scores, the U-Net11 Polygon model has the advantage of producing regular straight lines, a requirement of the building segmentation problem space, which the U-Net16 facade mask model is failing to do.

V. CONCLUSION

When it comes to 3D modeling of buildings using a base image, it is important to have knowledge about their contents. Image based segmentation techniques help to classify images pixels to an important building element. In this work we have presented deep learning based approaches to semantically segment facades of buildings. Initially, a model was trained to predict the facade as a mask. Then, as some types of buildings, such as masonry buildings, present facades with certain polygonal regularity, it is convenient to just predict the facade as a polygon. With this, another approach is presented in which a simplified version of the facade is reached after post-processing of building vertices detected by deep learning model. The qualitative and quantitative results shown in the present study let us conclude that deep learning approaches, especially the use

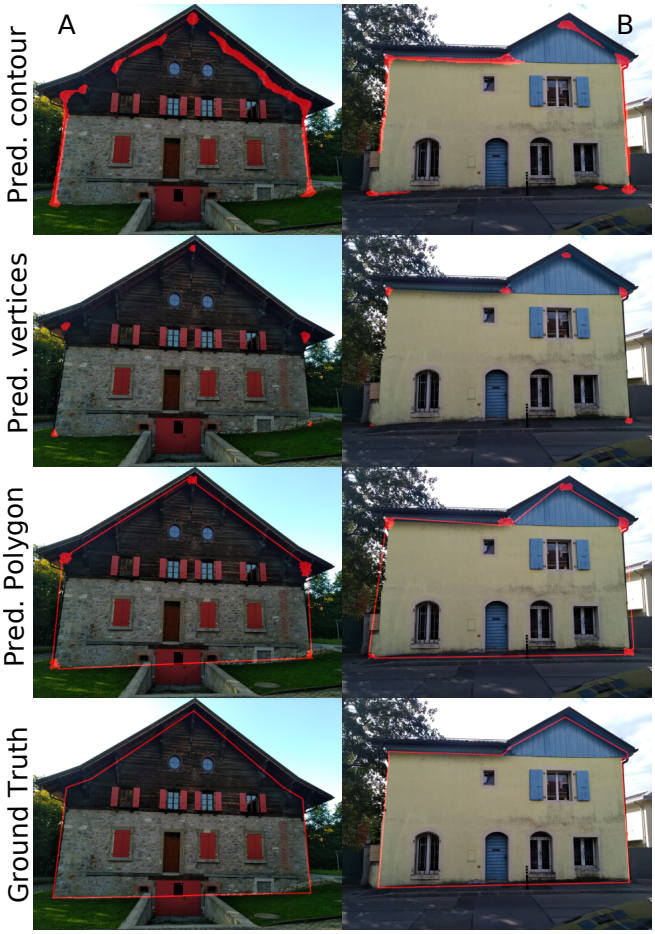

Figure 4. Predictions for U-Net11 model 3, semantic segmentation of the facade contours and vertices, and generating polygons from the vertices.

Figure 5. Comparing the predictions of PolyRNN++ and U-Net11 model's generated polygons, we can see why the U-Net11 model has a higher dice score.

of convolutional models, are appropriate to identify and segment facades aiming for a better understanding of the buildings' composition.

REFERENCES

[1] Z. Qin, F. Yu, C. Liu, and X. Chen, "How convolutional neural networks see the world - A survey of convolutional neural network visualization methods," *Conference on Computer Vision and Pattern Recognition*, vol. 1, no. 2, pp. 149–180, 2018.

[2] C. A. Vanegas, D. G. Aliaga, and B. Beneš, "Building reconstruction using manhattan-world grammars," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 358–365, 2010.

[3] O. Teboul, L. Simon, P. Koutsourakis, and N. Paragios, "Segmentation of building facades using procedural shape priors," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3105–3112, 2010.

[4] D. Acuna, H. Ling, A. Kar, and S. Fidler, "Efficient interactive annotation of segmentation datasets with polygon-rnn++," 2018.

[5] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham: Springer International Publishing, 2015, pp. 234–241.

[6] J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," oct 2015, pp. 847–856. [Online]. Available: https://ieeexplore.ieee.org/document/9022208/

[7] V. Iglovikov and A. Shvets, "TernausNet: U-Net with VGG11 Encoder Pre-Trained on ImageNet for Image Segmentation," 2018. [Online]. Available: http://arxiv.org/abs/1801.05746

[8] M. Schmitz and H. Mayer, "A convolutional network for semantic facade segmentation and interpretation," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, vol. 41, no. July, pp. 709–715, 2016.

[9] J. Femiani, W. R. Para, N. Mitra, and P. Wonka, "Facade segmentation in the wild," *arXiv*, 2018.

[10] H. Liu, J. Zhang, J. Zhu, and S. C. Hoi, "Deepfacade: A deep learning approach to facade parsing," *IJCAI International Joint Conference on Artificial Intelligence*, vol. 0, no. 12, pp. 2301–2307, 2020.

[11] W. Ma, W. Ma, S. Xu, and H. Zha, "Pyramid ALKNet for Semantic Parsing of Building Facade Image," *IEEE Geoscience and Remote Sensing Letters*, pp. 1–5, 2020.

[12] Z. Li, J. D. Wegner, and A. Lucchi, "Topological map extraction from overhead images," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2019-Octob, pp. 1715–1724, 2019.

[13] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlche Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[14] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 12 2014.

[15] A. Dutta and A. Zisserman, "The VGG image annotator (VIA)," *CoRR*, vol. abs/1904.10699, 2019. [Online]. Available: http://arxiv.org/abs/1904.10699