Cross field-based segmentation and learning-based vectorization for rectangular windows

Xiangyu Zhuo, Jiaojiao Tian, *Senior Member, IEEE,* and Friedrich Fraundorfer

***Abstract*—Detection and vectorization of windows from building façades are important for building energy modeling, civil engineering, and architecture design. However, current applications still face the challenges of low accuracy and lack of automation. In this paper we propose a new two-steps workflow for window segmentation and vectorization from façade images. First, we propose a cross field learning-based neural network architecture, which is augmented by a grid-based self-attention module for window segmentation from rectified façade images, resulting in pixel-wise window blobs. Second, we propose a regression neural network augmented by Squeeze-and-Excitation (SE) attention blocks for window vectorization. The network takes the segmentation results together with the original façade image as input, and directly outputs the position of window corners, resulting in vectorized window objects with improved accuracy. In order to validate the effectiveness of our method, experiments are carried out on four public façades image datasets, with results usually yielding a higher accuracy for the final window prediction in comparison to baseline methods on four datasets in terms of IoU score, F1 score, and pixel accuracy.**

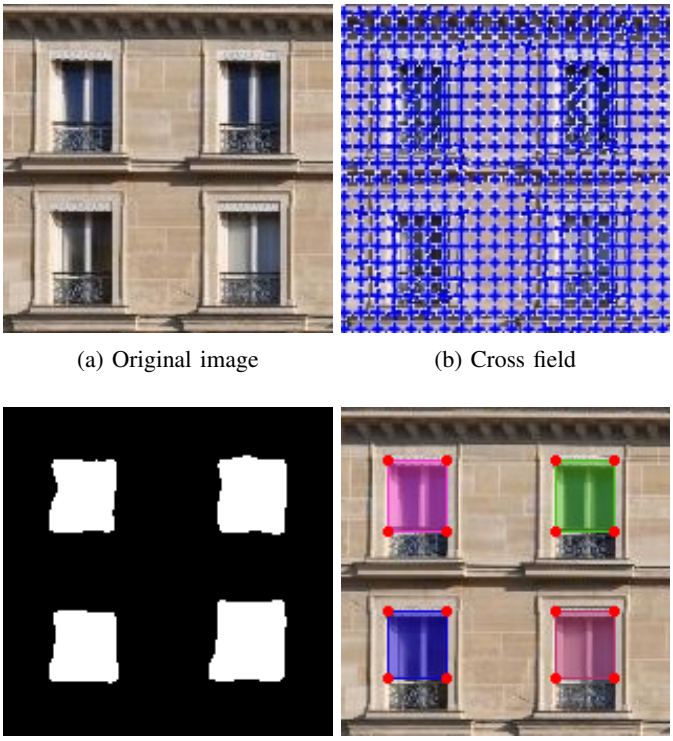
***Index Terms*—window segmentation, vectorization, façade parsing, deep learning.**

I. Introduction

THE location of windows is crucial for semantic and geometric understanding of building façades, and is often demanded in applications such as urban planning and Building Information Modeling (BIM) [1]. However, window segmentation remains a challenging task due to the complexity of building scenes. For example, windows in different buildings have different styles and shapes, and the existence of curtains introduces ambiguities in the windows detection process. Additionally, window detection from street-view images suffers from occlusion caused by trees and vehicles, among other objects [2]. As rectified façade images are more commonly used in building information modeling applications, we focus in this paper merely on rectified façade images, with the assumption that all windows have rectangular shapes.

In recent years, CNN-based methods outperformed traditional segmentation algorithms and are widely applied in a variety of vision-tasks such as face recognition, speech recognition, and vehicle detection. Deep neural network models have been used for building-related image segmentation tasks as well [2], [3]. Although window detection approaches based on deep learning generally outperform traditional methods in terms of standard accuracy, these methods suffer from the limited localization ability of CNNs, often resulting in blob-like segments, smooth corners and noisy object boundaries. This in turn results in an inaccurate representation of the regular shapes of windows, propagating these errors to the vectorization step. In order to achieve more accurate building prediction in the segmentation step, we introduce a novel deep neural network architecture adding a smooth cross field output to a fully-convolutional segmentation network. The integration of the cross field can substantially improve segmentation quality and generate more regular window boundaries.

Typically, CNN-based window detection methods generate pixel-wise window blobs. However, these window segments have to be converted into vector formats (e.g. CityGML) before they can be directly used in building modeling at or above Level of Detail 3 (LoD3), i.e., an architecturally detailed model with openings such as windows and doors. Traditional



(a) Original image (b) Cross field

(c) Window segmentation result (d) Window vectorization result

Fig. 1: Sample result of the proposed window detection method: (a) Original image; (b) Cross field overlaid on image; (c) Window segmentation result; (d) Window vectorization result, with red dots representing vectorized window corners.

X. Zhuo, J. Tian are with the Institute of Remote Sensing, German Aerospace Center, 82205 Weßling, Germany. E-mail: (xiangyu.zhuo@dlr.de, jiaojiao.tian@dlr.de).

F. Fraundorfer is with Institute of Computer Graphics and Vision, Graz University of Technology, 8010 Graz, Austria. E-mail: (fraundorfer@icg.tugraz.at).

vectorization approaches simply apply post-polygonization algorithms such as Douglas-Peucker [4] on rasterized building segmentation results. However, segmentation errors are also taken into account during polygonization, i.e., the quality of input segmentation mask directly affects the quality of the polygonization. For example, a minor segmentation error may lead to a wrong number of vertices or wrong building shapes. Rather than running a post-polygonization regularizer, we propose a regression network learning the location of window corners. The network takes the pixel-wise window prediction from semantic segmentation as well as the original façade image as input, and outputs the coordinates of window corners. While most polygonization methods only consider the geometric distance to the original pixel boundary, and have therefore negligible influence on the semantic accuracy [5], our method both improves the regularity of window shapes and their semantic accuracy, as it takes image features into consideration when predicting vertices. In addition, our vectorization method converts the window representation from pixels into a set of exactly four corners. Figure 1 depicts a sample result of the proposed window detection and vectorization method.

In summary, our main contributions are:

- Modelling the rectified façade image as a cross field aligned to object tangents, which improves segmentation accuracy by enforcing the alignment between segmentation results and the cross field.
- Integrating grid-based gating into the CNN model, further improving the segmentation accuracy.
- A regression neural network learning the position of window corners from rectified façade images based on the window segmentation result, yielding a vectorized representation of the windows.

The paper is organized as follows: we give an overview of related research work in Section II, and explain in depth the proposed window detection pipeline including window segmentation and vectorization in Sections III and IV, respectively. Experiments and implementation details are explained in Section V. We conclude in Section VI.

II. Related Work

This section reviews previous related studies on window segmentation and object vectorization.

1) Window segmentation: Existing window detection methods are broadly divided into three categories: grammar-based, traditional machine learning based, and deep learning based.

Grammar-based methods first generate pixel or object hypotheses, and then use shape grammars to extract window segments from the façade image. They rely on hand-crafted rules which represent structured geometries of buildings or façade objects. Zhao et al. [6] propose to parse registered ground-view images into architectural units for large-scale city modeling. They first decompose the environment into buildings, ground, and sky using a joint 2D-3D segmentation method, and then parse buildings into individual façades. Müller et al. [7] combine procedural modeling pipelines of shape grammars with image analysis in order to derive a meaningful hierarchical façade subdivision. Han and Zhu [8] present an attribute graph grammar for parsing images by maximizing a Bayesian posterior probability, or equivalently minimizing a description length. These grammar based methods usually achieve a pixel-wise accuracy below 85% [9] on the ECP benchmark [10] and suffer from low efficiency in the training and inference steps [11].

Traditional machine learning based façade segmentation approaches mainly rely on empirically designed features such as spatial, spectral and textual features; subsequently, windows are extracted using machine learning classifiers such as RF [12], SVM [13], or a combination of several classifiers [14]. Although a significant progress has been made with respect to previously described methods, these methods are restricted by their limited generalization ability, as they mainly rely on manual feature engineering, with the complex shape and texture properties of façades being difficult to model empirically [12] [13].

Recent advances in deep neural networks (DNNs) have significantly boosted the performance of object detection and semantic image analysis, going beyond traditional explicit feature design and being able to learn discriminative features for image representation. The Fully Convolutional Network (FCN) proposed by Long et al. [15] extends the CNNs to pixel-wise classification and has thus became widely used in building segmentation tasks. For example, Liu et al. [3][16] propose a FCN-8s-based network with a novel symmetric loss function and a region proposal network (RPN) for façade parsing. In [17], two semantic segmentation networks based on the U-Net [18] are designed for two types of façade structures respectively, and assembled to handle class imbalance. Ma et al. [19] propose a pyramid ALKNet for façade parsing, fully employing the regular structures of façades to aggregate non-local structural information, and therefore being able to deal with challenging scenarios such as occlusions and appearance ambiguities. Other CNN-based network structures widely used in façade parsing applications include SegNet [20] and Mask R-CNN [21].

In recent years, a series of techniques such as the attention module [22] and the transformer [23] have been proposed to complement the CNNs. In particular, attention modules can be integrated into backbones [24][25] or head networks [26][27] to encode distant dependencies or heterogeneous interactions, thus boosting the segmentation quality. Zhang et al. [28] employ a dual attentional network (DAN) module to model long-range dependencies, and introduce a novel symmetric loss function to encode prior knowledge improving the predictions of façade elements. To the best of the authors' knowledge, this work is the actual state-of-the-art for façade semantic segmentation.

Though CNN and its variants are still the primary network architectures for semantic segmentation tasks, other backbone structures such as Generative Adversarial Network (GAN) [29] and Recurrent Neural Network (RNN) [30] are also widely used in façade parsing. Yu et al. [31] employ an improved version of GAN to learn image data with similar characteristics and generate façade images, while Abdulnabi et al. [32] propose a RNN-based network for RGB-D scene semantic segmentation, where two RNNs are crossly connected through

transfer layers and trained simultaneously to extract cross-modality features.

In recent years, multi-modal techniques are gaining attention in deep learning-based semantic segmentation domain, especially for the applications to remote sensing. In [33], [34], CNNs are taken as a backbone and augmented by an advanced cross-channel reconstruction module. By means of fusing multiple features across modalities, the assembled network architecture learns more comprehensive representations of different remote sensing data. In [35], CNNs and GCNs are fused to improve the performance of hyperspectral image classification as they can extract different types of hyperspectral features. Although these methods are targeted at multi-modal data, whereas our study focuses on RGB façade imagery, they demonstrate the effectiveness of the fusion of multiple information.

A new trend in façade parsing research exploiting various features in the image is to combine geometric and spectral information. Girard et al. [5] propose a network that learns a frame field from the image and enforces its alignment to ground-truth contours. The additional structural information can effectively improve segmentation quality in building extraction, which motivates us to employ field theory in window segmentation applications. It has to be noted that although windows have generally more regular shapes than buildings, window segmentation is still challenging due to the existence of reflection, occlusion and varying illumination conditions. Since the frame field cannot well represent the structural characteristics of rectangular windows in rectified façade images, we propose in this paper to learn a cross field with constraints on orthogonality to incorporate the structural characteristics of windows. In addition, we augment the segmentation network with attention gates to further improve the segmentation quality.

2) Window vectorization: Representing windows as vectors is an essential step for building model generation at LoD3 or higher. Existing window vectorization methods can be broadly divided into two categories, based on either polygonization or keypoints detection.

Polygonization is a popular topic in computer vision and has many well known implementations. The most basic polygonization pipeline extracts the object contours as a chain of pixels, and then simplifies the resulting shape as a polygon. Popular simplification methods include the Douglas-Peucker algorithm [4] and Delaunay triangulation [36]. These only consider the pixel distance to the initial object contour rather than the geometric properties of the object, often introducing severe losses in accuracy in practice. More advanced polygonization methods, such as the Hough transform [37] and the active contour [38] algorithms, exploit geometric primitives such as line segments, and then assemble them into a polygon. In recent years, several deep neural network-based polygonization methods have been developed. Hatamizadeh et al. [39] proposes a neural network intimately combining the CNN with an Active Contour Model (ACM). Parameters of the ACM energy model are learnable, and can be used to precisely delineate buildings from aerial images.

Directly predicting vertices using neural networks is a new strategy in this field. Compared with traditional polygonization methods, only few vertices are required to represent regions with a large number of pixels. A variety of vertices prediction methods have been proposed. RNN-based networks such as Polygon-RNN [40], Polygon-RNN++ [41], and PolyMapper [42], employ a CNN to extract image features and a RNN to decode vertices. However, they perform beam search while predicting vertices, requiring more predictions than the number of output vertices, resulting in a relevant increase of computational burden. Besides, RNNs are usually more difficult to train. Li et al. [43] propose a novel window corner detection framework, employing a ResNet [44] to learn image features and generate heatmaps, from which locations and relationships of keypoints are decoded; finally, the keypoints are grouped together into final windows. However, this method suffers from frequent cross mismatching of keypoints, as adjacent windows usually exhibit similar patterns. Zorzi et al. [45] propose a CNN-based method for building polygonization and regularization. First, a CNN is employed for building segmentation, and then a GAN is used to regularize the initial segmentation boundaries and learn a building corner probability map, used to predict final vertices. Girard et al. [5] propose a novel method for building polygonization. They employ a CNN to learn a frame field, which provides additional geometric information to regularize building boundaries. To the best of the authors' knowledge, this is the actual state-of-the-art approach for building polygonization.

III. Window Segmentation

Unlike organic objects, windows have in most cases rectangular shapes with sharp corners, especially in rectified façade images. In order to capture this defining geometric information, we propose a neural network to learn a smooth cross field which is aligned to the tangent direction along window boundaries. In addition, we incorporate attention gates to further improve the segmentation quality.

The workflow of the proposed segmentation method is illustrated in Figure 2. Given a RGB façade image as input, we firstly learn feature maps using the attention-U-ResNet model; the feature maps are then passed on to a segmentation head and a cross field learning head, resulting in a segmentation map and a cross field prediction, respectively. It is to be noted that the actual cross field is pixelwise, i.e., each pixel is described with four vectors. For the sake of clarity, Figure 2 shows a sparse cross field plotted with an interval of 10 pixels.

A. Feature extraction

In the last decade, CNNs have been widely used for feature extraction in classification and segmentation tasks [15]. Several excellent neural network architectures have been proposed to further improve segmentation quality, such as U-Net [18], modified U-Net [18], and DenseNet [46]. Generally, our approach can use any deep segmentation model as backbone for feature extraction. For comparison, we tested different neural network architectures, ranging from small models such as U-Net16, a modified U-Net [18] model whose feature vectors are reduced from the original 64 to 16, to large models like

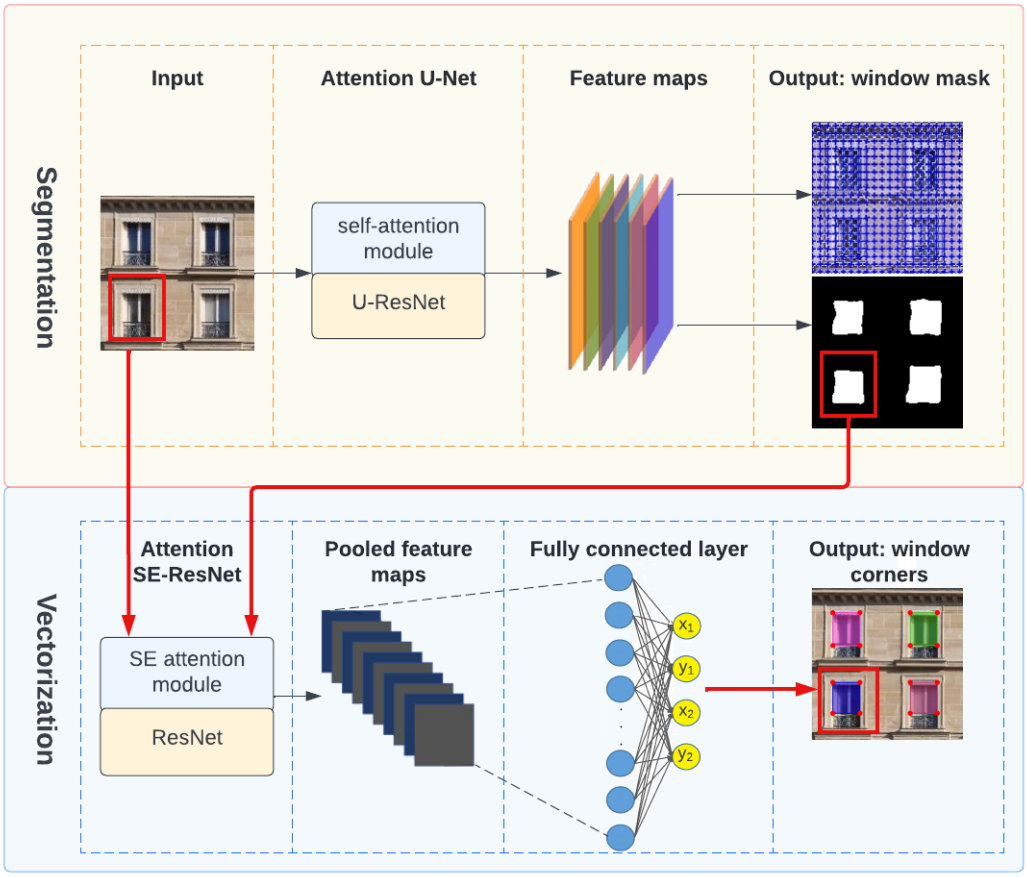


Fig. 2: Workflow of the proposed method. The segmentation model takes a RGB façade image as input and predicts a segmentation mask and a cross field. The predicted window mask and the original façade image are then fed as input to the regression neural network, resulting in rectangular window objects represented by the top left and bottom right corners.

UResNet-101, a modified U-Net [18] model whose encoder is replaced by a ResNet101 [47] pretrained on the ImageNet dataset [48]. As the latter achieves the best performance, we use it as backbone in our experiments as described in Section V.

B. Attention Module

In order to capture sufficient semantic information, feature maps in standard CNN models are gradually downsampled to increase the receptive field. Despite their good representative ability, these architectures suffer from redundant use of model parameters and lead to high computational burdens. In order to solve this problem, we propose to incorporate an attention gate (AG) model into the CNN architecture. The integration of AG can suppress irrelevant regions and focus on salient features by increasing the model sensitivity to foreground pixels. As a result, it can significantly improve segmentation accuracy while preserving computational efficiency.

In this paper we adopt the attention model proposed in [22], which involves grid-based gates to make attention coefficients more specific to local regions, leading to higher segmentation accuracy with respect to the gating based on global feature vectors [49]. Figure 3 shows a block diagram of the Attention U-ResNet segmentation model. It is to be noted that we have modified the original Attention U-Net model [22] to fit our backbone. Particularly, in the encoding part of the model, the input image is first downsampled by a factor of 4 via convolution, and then progressively downsampled by a factor of 2 via max-pooling. Then the propagated features are filtered by attention gates via skip connections. Since the image has been downsampled by 4 in the first step, the last attention gate is omitted.

C. Cross Field

In computer vision many algorithms aim at representing a surface with various features, and direction fields are developed to solve the problem of orienting the features on the surface [50]. Cross fields, as proposed by Hertzmann and Zorin, are maps defined on a surface of which each point is assigned a smoothly varying pair of orthogonal directions on the tangent plane [51]. The topology of a cross field is determined by singular points and separatrix lines connecting them: the singularities divert the flow of tangential directions, and the separatrices divide the surface into uniform patches

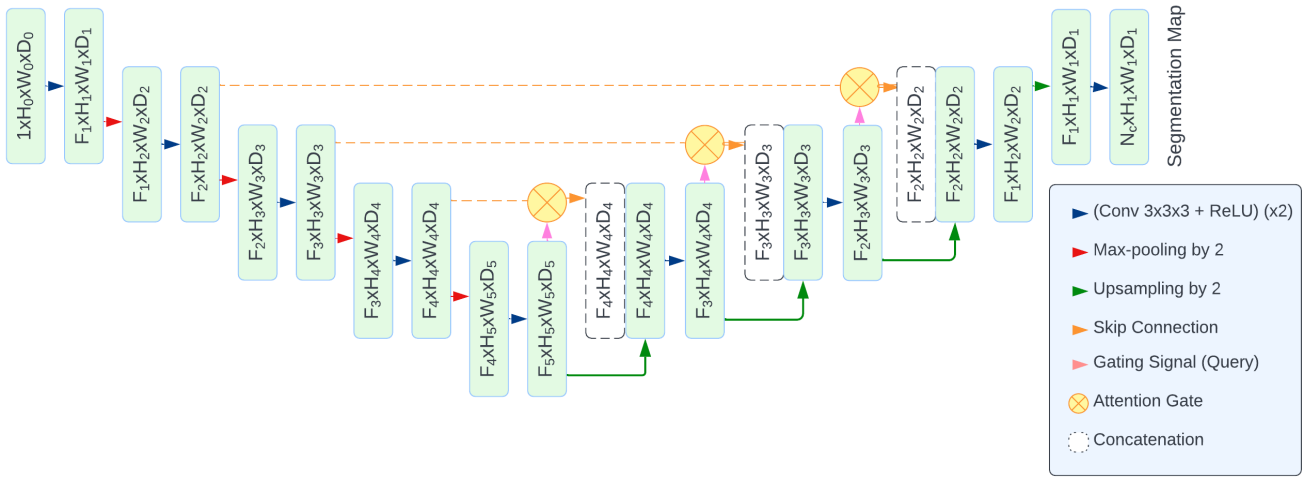
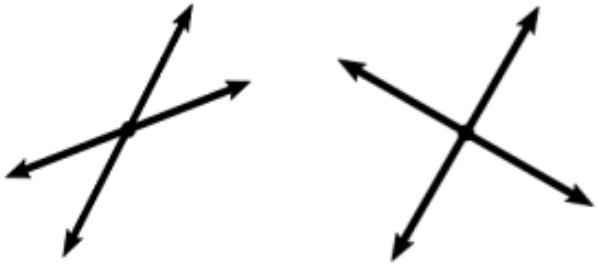


Fig. 3: Architecture of the integrated attention segmentation model. H, W, C denote height, width and depth, respectively, while N_c denotes the number of classes.



(a) Frame field (b) Cross field

Fig. 4: Comparison of frame field (a) and cross field (b).

[52]. Due to the arrangement of these topological features, cross fields can not only be used to represent certain surface characteristics, such as curvature extrema and principal curvature directions, but can also be used with given constraints [50][53]. By contrast, frame fields are a non-orthogonal and non unit-length generalization of cross fields, and can represent smoothly varying linear transformations on tangent spaces of a surface [54]. Figure 4 depicts a comparison of the cross field and the frame field: directions in the cross field are orthogonal, whereas such constraint does not apply for the frame field.

In computer vision applications, both cross fields and frame fields can be used to model the tangents of objects. When it comes to rectified façade images, most windows appear as rectangles with orthogonal corners. As cross fields are invariant to the rotation of $\pi/2$ while frame fields are not, we propose to represent the image tangent plane as a cross field, as it can better capture the orthogonality of windows.

Following the setting in [55], with $\mathbf{u} \in \mathbb{C}$ representing the curve tangent near a given pixel, a cross field at this pixel is defined as set of four vectors $\langle \mathbf{w}, \mathbf{w}^{\perp}, -\mathbf{w}, -\mathbf{w}^{\perp} \rangle$ in cyclic order. In order to avoid relabeling and sign changes, we represent the direction using the following complex polynomial [55]:

$$f(z) = z^4 - w^4 = z^4 + c_0 \tag{1}$$

In (1), $c_0 = -\mathbf{w}^4$ uniquely determines a cross field by its root set $\{ \sqrt[4]{|c_0|} exp(i\frac{k\pi}{2}) \mid 0 \leq k \leq 3 \}$. In the following text, we denote the function in (1) as $f(z; c_0)$. In order to avoid sign and ordering ambiguity, we learn c_0 instead of the vectors $\langle \mathbf{w}, \mathbf{w}^{\perp}, -\mathbf{w}, -\mathbf{w}^{\perp} \rangle$.

In order to compute a smooth cross field, Bessmeltsev et al. [56] propose a variational computation approach using the L-BFGS algorithm. Furthermore, Taktasheva et al. [57] propose a deep learning-based approach for computation. In our work, we solve the cross field variationally by regressing the value of direction vectors at each pixel with a neural network, similarly as what has been explored in [57].

D. Segmentation Network Architecture

Our segmentation network takes a RGB image with size $H \times W$ as input and computes a segmentation map and a cross field as output. In this part, we follow the general network architecture design of [5], which can take any deep neural network model as a backbone, such as DeepLabV3 [58] and ResNet [59], and output a N-dimensional feature map $\hat{y}_{feature} \in \mathcal{R}^{N \times H \times W}$. This feature map is then appended to two blocks, one for segmentation and the other for cross field computation. We integrate the segmentation losses and the alignment losses proposed by [5], but replace the frame field losses by cross field losses. Figure 5 shows the loss functions in the segmentation network, which can be divided into three categories: segmentation losses, cross field losses and coupling losses.

Segmentation losses: For the purpose of segmentation, feature maps are passed on to a fully convolutional block, which consists of a 3×3 convolutional layer, a batch normalization layer, an ELU layer, another 3×3 convolutional layer, and a sigmoid layer. The final output of this segmentation head is a segmentation map $\hat{y}_{seg} \in \mathcal{R}^{2 \times H \times W}$. The segmentation map has two channels, one is window interiors denoted by y_{int} and the other is window boundaries denoted by y_{bnd}, and the corresponding losses of window interiors and window boundaries are L_{int} and L_{bnd}, respectively. It needs to be

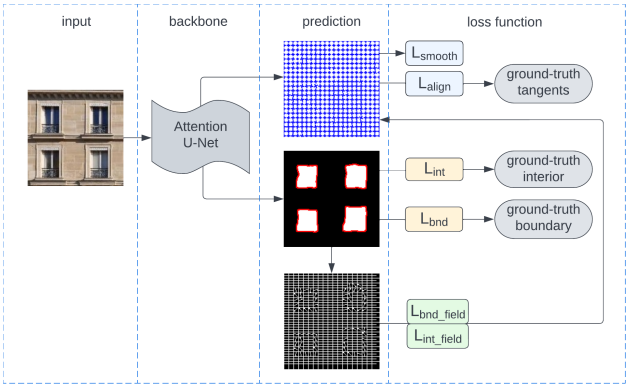


Fig. 5: Loss functions in segmentation network. L_{int} and L_{bnd} are segmentation losses, L_{smooth} and L_{align} are cross field losses, L_{bnd_field} and L_{int_field} are coupling losses.

noted that the training data is also prepared in two sets, one is the ground-truth for the window interiors and the other for window boundaries.

Cross field losses: In addition to the segmentation head, we append another block to the backbone to compute the cross field. This takes the concatenation of the feature map and the segmentation map $[\hat{y}_{feature}, \hat{y}_{seg}] \in \mathcal{R}^{(N+2)\times H\times W}$ as inputs, and outputs parameters $\hat{c}_0$ representing the cross field as output. The ground-truth for training is the tangent direction θ_τ of the contour. Following [56], we take three losses into consideration:

- Alignment. The alignment loss is defined as:

$$L_{align} = \frac{1}{HW}\int_I y_{bnd}(x)|f(e^{i\theta_\tau}; \hat{c}_0(x))|^2 dx \tag{2}$$

  This loss function enforces the alignment of the cross field with the tangent directions. This loss has a lower value when the polynomial has a root near $e^{i\theta_\tau}$, implying that at least one of the field directions $\langle \mathbf{w}, \mathbf{w}^\perp, -\mathbf{w}, -\mathbf{w}^\perp \rangle$ is aligned with the tangent direction τ.
- Smoothness. The smoothness loss is defined as:

$$L_{smooth} = \frac{1}{HW}\int_I \|\nabla \hat{c}_0(x)\|^2 dx \tag{3}$$

  This term is a Dirichlet energy which enforces the value of c_0 to vary smoothly in order to yield a smooth cross field.

Coupling losses: It has been proved in [5] that coupling the losses of segmentation and frame field can increase segmentation accuracy, therefore we also consider the alignment between the segmentation output and the cross field output by minimizing the coupling losses, as follows:

- Alignment between the predicted interior map and the cross field. This loss is defined as:

$$L_{int\_field} = \frac{1}{HW}\int_I |f(\nabla \hat{y}_{int}(x); \hat{c}_0(x))|^2 dx \tag{4}$$

  This loss function measures the consistency between the spatial gradient of the output interior map $\hat{y}_{int}$ and the tangent direction of the output cross field.
- Alignment between the predicted boundary map and the cross field. This loss is defined as:

$$L_{bnd\_field} = \frac{1}{HW}\int_I |f(\nabla \hat{y}_{bnd}(x); \hat{c}_0(x))|^2 dx \tag{5}$$

  This term aligns the spatial gradient of the output window boundary map $\hat{y}_{bnd}$ with the tangent direction of the output cross field.

Finally, we normalize the losses above and sum them. This results in a final loss function, similar to the one described in [5].

IV. Window Corners Regression

A common problem of CNNs is their low localization accuracy, as the output of semantic segmentation is usually pixel blobs with blurred object boundaries and smooth corners. In the case of window segmentation, blob-like window segments cannot well represent regular window shapes and thus need to be vectorized. Traditional window vectorization methods simply apply polygonization algorithms (e.g. Douglas-Peucker algorithm [4]) on the input segmentation masks alone, and have two main disadvantages. First, the spectral information of images is not utilized in the vectorization step, therefore the vectorization has only subtle influence on the segmentation accuracy, and in some cases higher regularization even leads to a slightly lower IoU score than the initial segmentation blobs. Second, traditional vectorization methods also take segmentation errors into account, with the polygonization accuracy greatly relying on the quality of the input segmentation mask. By contrast, we propose to learn the position of window corners using a regression neural network, which learns to predict window corners by taking both image features and initial window predictions into account.

A. Vectorization Network

For the vectorization network architecture, we use a SE-ResNet [27] to extract deep features from original images. As a variant of squeeze-and-excitation networks (SE-Nets), the SE-ResNet consists of a ResNet as backbone, and integrates SE blocks after the non-linearity layer following each convolution. The SE block transformation is used as the non-identity branch of the residual module. Figure 6 illustrates a typical schema of SE-ResNet used in our experiment. It should be stressed that we modified the output layer of the SE-ResNet to a 4-channel fully connected layer, as we formulate the rectified window as a rectangle which can be represented by its top left and bottom right corners.

The workflow for window corner prediction is illustrated in Figure 2. Particularly, the output of the segmentation network is taken together with the original RGB image as input. For each window instance, a Region of Interest (ROI) is cropped from the original façade image. The size of the ROI is proportional to the size of the window instance, indicating the possible area where the actual window may be located. Then feature maps are extracted from the ROIs by the SE-ResNet, and passed on to the fully connected layer, resulting in four regressed values (x_1, y_1, x_2, y_2) corresponding to the top left

(x_1, y_1) and the bottom right corners (x_2, y_2) of the window, respectively.

Whereas traditional polygonization methods take merely binary window masks as input and are prone to segmentation errors, our network utilizes image features in addition and is therefore more robust to such errors. Given an imperfect window segmentation blob as input, our method can refine the position of window corners, resulting in a more accurate window prediction.

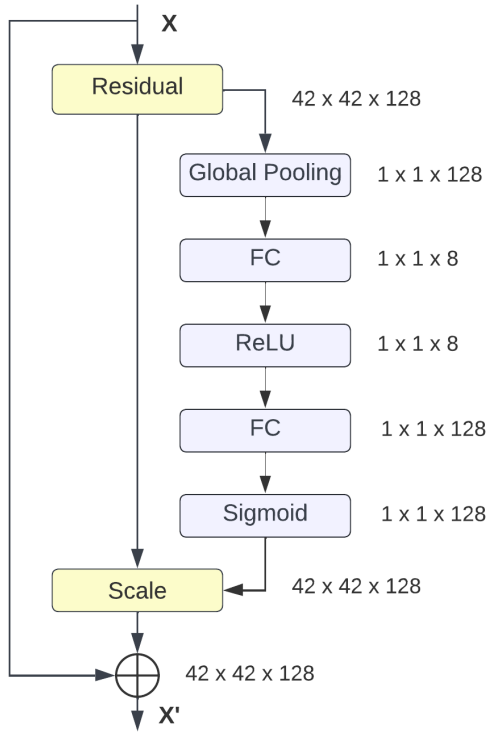


Fig. 6: Schema of the SE-ResNet module.

B. Network Implementation

In our experiment we increase the ROI of each individual window prediction by 10%, and then crop the corresponding patch from the original façade image as input. All input patches are resized to 128×128 for consistency. When passed on to the SE-ResNet, these image patches are firstly downsampled by a factor of 3 to a size of 42×42, and then passed on to the residual module as input. Figure 6 illustrates an attention module in the SE-ResNet, where x denotes the input features with a shape of $42 \times 42 \times 128$. Such schema repetitively occurs in the network for gating, and the input image is progressively filtered and downsampled at each stage from 128 to 42, 14, 7, 3 and 1 pixel(s).

Figure 7 demonstrates the loss calculation in the network. Since we modify the last layer of the SE-ResNet as a 4-channel fully connected layer, the output of the network is four scalars (x_1, y_1, x_2, y_2), standing for the top left (x_1, y_1) and the bottom right (x_2, y_2) corners of the window, respectively. In order to measure the prediction errors, we compare the regressed values with ground-truth values, namely the four window corners $(\hat{x_1}, \hat{y_1}, \hat{x_2}, \hat{y_2})$ extracted from the ground-truth window mask. We use the Smooth L1 loss as defined in Equation 6, where Y denotes the predicted corners vector (x_1, y_1, x_2, y_2) and $\hat{Y}$ denotes the ground-truth corners vector $(\hat{x_1}, \hat{y_1}, \hat{x_2}, \hat{y_2})$. Here $beta$ is a hyper-parameter that needs to be manually tuned. As $beta$ approaches 0, Smooth L1 loss converges to L1 Loss; as $beta$ approaches $+\infty$, Smooth L1 loss converges to a constant 0 loss. In practice, the hyper-parameter $beta$ is usually set to 1, and we follow the same settings in our experiments. We also tested MSE loss and MAE loss, which resulted in similar accuracy as the Smooth L1 loss and therefore not reported in details in this paper.

$$L_\delta = \begin{cases} \frac{1}{2}(Y - \hat{Y})^2/beta, & if \left|(Y - \hat{Y})\right| < beta \\ |Y - \hat{Y}| - \frac{1}{2}beta, & otherwise \end{cases}. \quad (6)$$

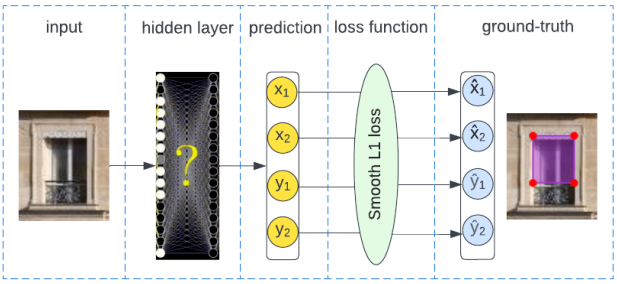


Fig. 7: Loss calculation for vectorization network.

V. Experiment

A. Experiment design

In order to explore the performance of the proposed method, we test our segmentation model on four benchmark façade datasets containing the window class. In addition, we compare the segmentation results with several state-of-the-art approaches using various evaluation metrics.

Our method is implemented in PyTorch [60] trained on four NVIDIA 2080Ti GPUs. During the training, the network is initialized with weights that were pre-trained on ImageNet. Then the network is fine-tuned and tested on the four window datasets. We employed Adam as optimizer for both the segmentation and regression networks. As the ECP dataset, Graz50 dataset and Paris Artdeco dataset are relatively small, the training and validation losses converge quickly to a small value after c.a. 20 epochs. The CMP dataset has a larger size and converges to a small loss after c.a. 50 epochs.

B. Dataset

The **ECP** dataset [10] was published in 2010, it consists of 104 façade images in solely Hausmannian style buildings in Paris with highly regular structures. Unlike most datasets where the façades are in the same plane, the ECP dataset contains several cases of roof windows that stretch out of or

behind the façade plane, as shown in Figure 8a. Images in this dataset are rectified and manually annotated in 8 classes: *window, wall, balcony, door, shop, sky, chimney, roof*. The annotation rule follows uniform Haussmanian-style grammar, i.e., all windows are annotated as rectangles, even though some of them are arc-shaped. This dataset has been widely used to evaluate window detection or façade segmentation approaches [28], [61], [62], [3].

The **CMP** dataset [63] was assembled in 2013 at the Center for Machine Perception. It is comprised of two sub-datasets: the CMP base dataset contains 378 images featuring planar façades with dense/strong regularity, while the CMP extended dataset contains 228 images featuring irregular, non-planar, sparse or substantially occluded façades. These façades images are collected from different cities around the world, portraying diverse architectural styles and various resolutions. Figure 8b depicts a modern-style building in the CMP dataset. All 606 images are rectified and manually annotated in 11 classes: *façade, molding, cornice, pillar, window, door, sill, blind, balcony, shop, deco*. All objects are annotated as rectangles, limited by the image scope in size and position, while overlap is allowed. This dataset has been widely used as benchmark for window detection or façade segmentation tasks [64], [65], [66].

The **Graz50** dataset [67] was published in 2012, and contains 50 rectified images at different spatial resolutions. The images are taken from various locations in the historical Austrian city of Graz and portray buildings of various architectural styles such as Classicism, Biedermeier, Historicism, Art Nouveau and several modern styles. This dataset shows more complex façade layouts with respect to other façade datasets. Besides, unlike the ECP, CMP and ParisArtDeco datasets, roof windows in Graz50 dataset are not annotated, as shown in Figure 8c. The images are generated automatically by extracting a piecewise planar geometry from about 30 perspective images. The dataset includes 4 classes: *wall, door, window, sky* and has been widely used for window detection or façade segmentation studies [28], [17], [3], [66].

The **ParisArtDeco** dataset [68] was published in 2014, and consists of 79 images acquired at different spatial resolutions showing Art Deco-style buildings in Paris. Façades in this dataset are similar to the Hausmannian architecture, but windows are here generally larger. All images are rectified, resulting in some layout inconsistencies, as some windows are protruding in the Art-deco style, as in the case of roof windows in the **ECP** dataset (see example in Fig. 8d). The dataset contains 7 classes: *door, shop, balcony, window, wall, sky, roof*. A large part of the ParisArtDeco dataset are densely occluded by trees or street signs, making it more challenge than other façade segmentation benchmarks. Therefore, it is specifically used to validate the robustness of the segmentation methods in presence of occlusions [2], [3], [69], [62].

The described datasets are designed for façade parsing and contain multiple categories such as doors and balconies. As we are only interested in windows in this study, we converted the multi-class labels into binary window masks. A common problem for these datasets is that they do not take occlusions in consideration and annotate arc-shaped windows as rectangles, therefore the annotations are not precise for these cases. As original images in each dataset have different shapes, we resize all images as well as masks into patches of 300×300. For each dataset, we follow the same design proposed in [28], i.e., data is randomly split into 80% for training and 20% for testing. The comparison experiments are carried out on each dataset.

C. Metrics

We use two types of metrics for evaluation purposes: area-based for segmentation accuracy, and shape-based for vectorization accuracy. regarding area-based metrics, the traditional one to evaluate semantic segmentation is *Pixel accuracy* (also named as overall accuracy) [70], which simply reports the percentage of correctly classified pixels in the image, as defined in Equation 7

$$Pixel\ accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

where, for a given class X, TP denotes True Positive, namely the number of pixels classified correctly as X, FP denotes False Positive, namely the number of pixels classified incorrectly as X, TN denotes True Negative, namely the number of pixels classified correctly as not X, FN denotes False Negative, namely the number of pixels classified incorrectly as not X.

Besides, *F1 score* [71] is the harmonic mean of Precision and Recall and gives a better measure of the incorrectly classified cases with respect to *Pixel accuracy*. The *F1 score* is defined as:

$$F1\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{8}$$

where *Precision* is the fraction of the correctly identified positive cases over all the predicted positive cases, while *Recall* is the fraction of the correctly identified positive cases over all the actual positive cases, as defined, respectively, as:

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

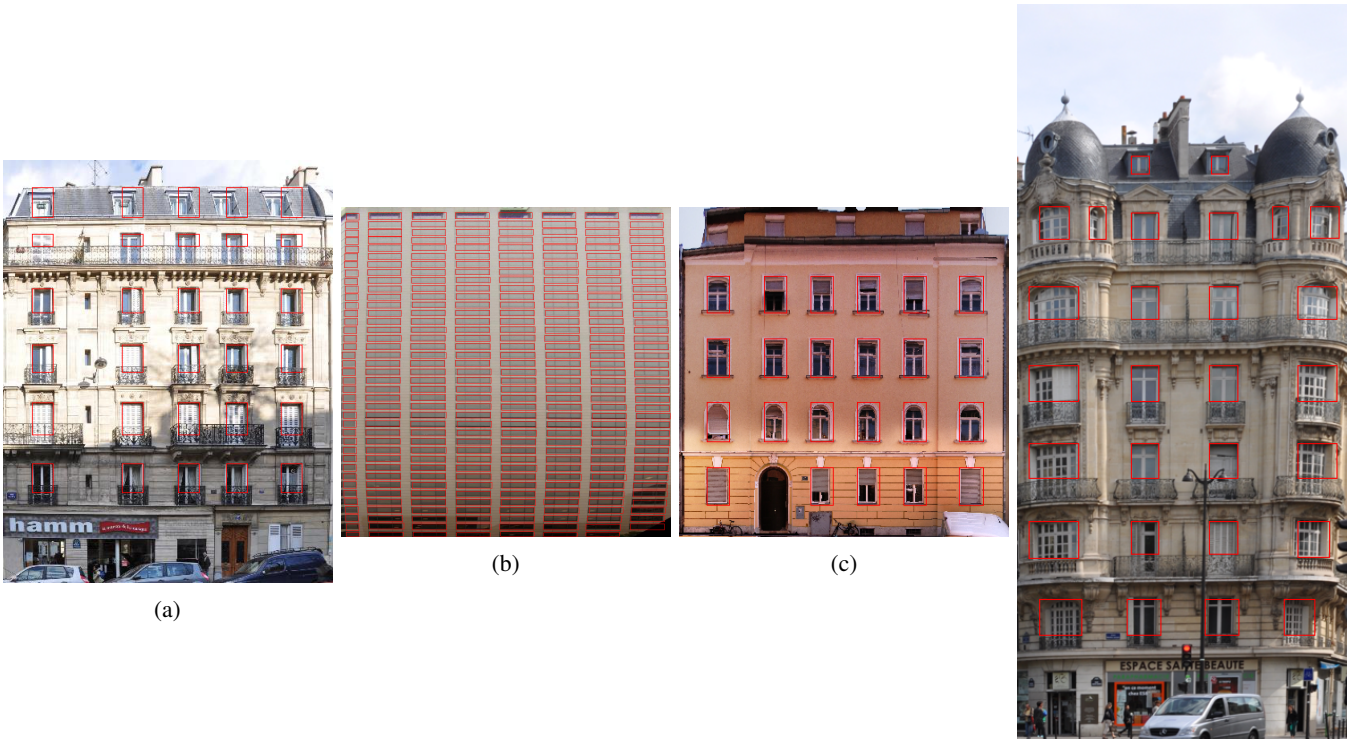
In addition, a widely used metric for evaluating image segmentation accuracy is the Intersection over Union (*IoU*), also referred to as the Jaccard index. The *IoU* value is defined as:

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Unioin} = \frac{TP}{TP + FN + FP} \tag{11}$$

As the datasets used in our experiments involve a relevant number of images, we employ *Mean IoU* to evaluate the average performance of the segmentation accuracy among multiple images. The definition of *mIoU* is given as:

$$mIoU = \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N} (TP_i + FP_i + FN_i)} \tag{12}$$

where N is the number of images involved in evaluation, TP_i the number of true positives of the i_{th} image, FP_i the pixel



(a) (b) (c) (d)

Fig. 8: Sample annotations in ECP, CMP, Graz50 and ParisArtdeco datasets. Red rectangles represent the projection of window annotations on the original images.

number of false positives of the i_{th} image, and FN_i the pixel number of false negatives of the i_{th} image. The IoU metric used in the following text refers to the mean IoU value.

Usually, a prediction with $IoU > 0.5$ is considered as true positive prediction, but a change in the threshold may introduce a bias in the evaluation metric. One way to solve the problem is to use a range of IoU threshold values. For example, in COCO evaluation [72], the IoU threshold ranges from 0.5 to 0.95 [72]. In our experiments, we calculate the average precision (AP) and average recall (AR) at fixed $IoUs$ such as $IoU = 0.5$ and $IoU = 0.75$, which we refer to as $AP50$, $AR50$, $AP75$ and $AR75$, respectively.

Higher IoU and $F1$ values do not always indicate more accurate object representation, especially when assessing vectorized results. In order to better evaluate the position accuracy of the vectorized window corners, we propose to use the Hausdorff Distance [73], a metric defined between two finite point sets $A = \{a_1, ..., a_p\}$ and $B = \{b_1, ..., b_q\}$ as:

$$H(A, B) = max(h(A, B), h(B, A)) \tag{13}$$

where

$$h(A, B) = \max_{a \in A} \max_{b \in B} ||a - b|| \tag{14}$$

$|| \cdot ||$ is an underlying norm on the points of A and B, and we use Euclidean norm in our experiments. A and B stand for the coordinates of predicted window corners and ground-truth window corners, respectively.

D. Segmentation results

We compare the segmentation accuracy of our segmentation model (denoted as *Ours*) with other state-of-the-art methods, including the DeepFacade network [3] (denoted as *DeepFacade*), the refined DAN-PSPNet with symmetric loss function [28] (denoted as *DAN-PSPNet-*L_{sym}), the Frame Field Polygonization network [5] (denoted as *FFP*) and the DeepWindows network [74] (denoted as *DeepWindows*). Among them, we use the source code provided by the authors to implement the FFP [5] and the DeepWindows [74] networks. We cannot reproduce results of the DeepFacade [3] and PSPNet [28] networks, as their codes are either not open-source or written in an outdated deep learning framework. Thus, we directly report their numerical results from the original papers, which, however, do not completely cover all the datasets regarding all metrics. For the FFP and DeepWindows methods, we reproduce the networks and test them on all datasets in order to have complete results.

In addition, in order to demonstrate the improvement in semantic accuracy of our vectorization method, we rasterized the vectorization result for area-based evaluation, indicated as *Ours-refine* in the tables below.

1) ECP dataset: The numeric evaluations of the segmentation results are listed in Table I, where the results of DeepFacade and PSPNet are reported from the original papers, and therefore some values are missing. It can be seen that our model ranks first in terms of IoU score, and achieve about the same $F1$ score and pixel accuracy as the DAN-PSPNet-L_{sym}.

The qualitative results on the ECP dataset is illustrated in

ECP			
Model	mIoU	F1	Pixel Accuracy
DeepFacade[3]	80.3	-	**97.6**
DAN-PSPNet-L_{sym}[28]	81.6	**91**	-
DeepWindows	63.8	79.6	97.2
FFP	77.3	87.2	96.3
Ours	**82.0**	90.1	97.0
Ours_refine	87.4	93.2	99.2

TABLE I: Accuracy comparison on ECP dataset in metrics of IoU score, F1 score and pixel accuracy. *Ours* denotes the result of the proposed segmentation model, *Ours_refine* denotes the refined result of the proposed vectorization model.

CMP			
Model	mIoU	F1	Pixel accuracy
DeepFacade[3]	-	80	95
DAN-PSPNet-L_{sym}[28]	-	-	-
DeepWindows	56.7	71.3	96.5
FFP	60.3	78.6	96.9
Ours	**64.1**	**82.4**	**97.2**
Ours_refine	68.6	84.6	98.3

TABLE II: Accuracy comparison on CMP dataset in metrics of IoU score, F1 score and pixel accuracy. *Ours* denotes the result of the proposed segmentation model, *Ours_refine* denotes the refined result of the proposed vectorization model.

Figure 9. Column (a) is the input image; column (b) is the segmentation result of the DeepWindows network overlaid on the original image; column (c) is the segmentation result of the FFP network overlaid on the original image; column (d) is the result of the proposed segmentation network overlaid on the original image; column (e) is the ground-truth overlaid on the original image. It can be seen that our method can make more accurate and regular predictions with respect to FFP and DeepWindows. DeepFacade achieves a higher pixel-accuracy, while DAN-PSPNet-L_{sym} reaches a higher $F1$ score with respect to our segmentation approach, but *Ours* has a higher IoU value. Furthermore, after the vectorization step, the accuracy results further improved.

Graz50			
Model	mIoU	F1	Pixel acuracy
DeepFacade[3]	71.3	-	88.8
DAN-PSPNet-L_{sym}[28]	-	-	-
DeepWindows	59.9	70.2	93.8
FFP	69.6	81.1	**94.2**
Ours	**73.1**	**84.3**	94.1
Ours_refine	76.9	88.1	96.8

TABLE III: Accuracy comparison on Graz50 dataset in metrics of IoU score, F1 score and pixel accuracy. *Ours* denotes the result of the proposed segmentation model, *Ours_refine* denotes the refined result of the proposed vectorization model.

ParisArtDeco			
Model	mIoU	F1	Pixel accuracy
DeepFacade[3]	70.7	-	95.4
DAN-PSPNet-L_{sym}[28]	**78.9**	**88**	-
DeepWindows	60.5	74.6	96.1
FFP	68.2	84.9	95.1
Ours	72.1	87.7	**96.6**
Ours_refine	77.4	89.1	97.8

TABLE IV: Accuracy comparison on ArtDeco dataset in metrics of IoU score, F1 score and pixel accuracy. *Ours* denotes the result of the proposed segmentation model, *Ours_refine* denotes the refined result of the proposed vectorization model.

2) CMP dataset: The quantitative evaluation results are listed in Table II. Our segmentation method ranks already first in comparison with previous methods in all metrics. To be specific, our method outperforms DeepWindows by c.a. 7% in terms of IoU score and FFP by c.a. 4% in terms of both IoU score and F1 score. Figure 10 shows qualitative comparison of the segmentation results. It can be seen that our method detects fewer false positives with respect to the FFP method and generates more regular and visually pleasing segmentation results. For example, the second and third row show building façades with doors, which have similar appearance as windows. DeepWindows and FFP tend to make false predictions on such scenarios, whereas our method is more robust and can distinguish between windows and doors.

3) Graz50: Table III lists the accuracy evaluations on the Graz50 dataset. Our method achieves 73.1% in terms of IoU score, outperforming the previous best, i.e., DeepFacade by c.a. 2% and FFP by c.a. 3.5%. Besides, our method ranks first in terms of F1 score and outperforms the best competitor by c.a. 3%. Figure 11 presents the qualitative comparison of our method with state-of-the-art methods. It is to be noted that the Graz50 dataset has inconsistent annotations for windows on raised ground floor, for example, in the first and second rows, where openings on the building bottom are not annotated as windows; however, in the third row, similar openings are annotated as windows. Such inconsistent class definition may confuse the network and lead to vulnerable performance on such scenarios. Nevertheless, our method still outperforms FFP and DeepWindows in overall accuracy and achieves more visually pleasing results.

4) ParisArtDeco dataset: Table IV lists the quantitative evaluation results on the ParisArtDeco dataset. Our method achieves an accuracy of 96.6% while the previous best, DeepWindows, has an accuracy of 95.1%. Besides, the F1 score of our method is 87.7%, only 0.3% smaller than the previous best result. As for other experiments, after the additional refinement, we achieve the highest accuracy for all evaluation metrics.

Figure 12 depicts the qualitative comparison of segmentation results for our method and previous methods. It needs to be noted that the ParisArtDeco dataset is more challenging than other datasets as it is largely occluded by vegetation. Typically, it is difficult for neural networks to learn such

(a) Original image (b) DeepWindows (c) FFP (d) Our model (e) Ground-truth

Fig. 9: Segmentation results on ECP dataset: (a) original image; (b) results of DeepWindows network; (c) results of FFP network (d) results of our segmentation model; (e) ground-truth.

hidden information. However, our method has successfully learnt the patterns of window layout and can well predict windows that are occluded by vegetation. As can be seen in Figure 12, the sample façades are all partially occluded by trees. For the façade in the third row, even more than half of it is blocked by trees. Despite the presence of large occlusions, our method makes reasonable predictions for the hidden windows. Although the other methods can also cope with occlusions to some extent, our method is the most robust and yields more regularized shapes for windows.

5) Ablation study of segmentation performance: In addition to the comparison to state-of-the-art approaches, we carry out an ablation study to assess the effectiveness of our network architecture. As our method can take any segmentation model as backbone, we test different ones including U-Net16 (namely a small U-Net [18] with 16 starting hidden features) and UResNet101 (namely a U-Net whose encoder part is replaced by a ResNet-101). In our implementation, the U-Net is randomly initialized whereas the UResNet101 is pretrained on ImageNet [48]. Besides, we also compare our segmentation model to other baseline methods, including the Mask R-CNN [21], UResNet101 and FFP.

In order to have a larger dataset available, we merge the ECP, CMP, Graz50 and ParisArtDeco datasets and randomly split the combined dataset into a 80% training set and a 20% testing set. We conduct the ablation experiments on the merged dataset, with a quantitative assessment reported in Table V. Mask R-CNN and UResNet101 are used as baseline segmentation methods: both of them are pretrained on ImageNet[48], and it can be seen that UResNet101 achieves higher segmentation accuracy with respect to Mask R-CNN. FFP$_{unet}$ refers to the FFP implementation with standard U-Net as backbone; Ours$_{unet}$ (without attention) refers to our model using as backbone the standard U-Net without attention gates; finally, Ours$_{unet}$ refers to our model with the standard U-Net as backbone and with attention gates. As the U-Net is not pretrained, this setting yields generally lower precision and recall score compared to the Mask R-CNN and UResNet101, which use pre-trained weights. However, when using the same backbone, our method outperforms the FFP method in all metrics, demonstrating the effectiveness of our cross field structure. Besides, Ours$_{unet}$ (with attention) achieves higher AP (Average Precision) and AR (Average Recall) scores in all metrics with respect to Ours$_{unet}$ (without attention), proving that the integration of attention gates can effectively improve the segmentation accuracy.

FFP$_{uresnet101}$ refers to the FFP implementation with UResNet101 as backbone; Ours$_{uresnet101}$ (without attention) refers to our model using as backbone the UResNet101 without attention gates; Ours$_{uresnet101}$ refers to our model with the UResNet101 as backbone and with attention gates. It can be seen that these models achieves much higher AP and AR scores than UResNet101 itself. By comparing Ours$_{uresnet101}$ (without attention) and FFP$_{uresnet101}$, it can be seen that

(a) Original image (b) DeepWindows (c) FFP (d) Our model (e) Ground-truth

Fig. 10: Segmentation results on CMP dataset: (a) original image; (b) results of DeepWindows network; (c) results of FFP network (d) results of our segmentation model; (e) ground-truth.

the integration of cross field can effectively improve the segmentation accuracy. Further more, Ours$_{uresnet101}$ (with attention) achieves higher scores than Ours$_{uresnet101}$ (without attention), showing the effectiveness of attention gates.

Overall, the ablation study proves that the integration of both cross field and attention gates can effectively improve segmentation accuracy, especially when employing their combination.

E. Vectorization results

The window segmentation results are vectorized using the proposed vectorization network. The vectorization accuracy is evaluated using both point-based metrics and area-based metrics. For the former class, we evaluate the accuracy of the vectorized window corners using Hausdorff Distance [73] and compare the results with baseline vectorization methods, including the ACM polygonization model used in the FFP network [5], PolyRNN+ model [75] and the Douglas-Peucker method [4]. In order to eliminate the influence of the input mask, we use the same segmentation results as input for all the three vectorization methods. For the latter class, we convert the vectorized window objects back into rasters, and then compare their semantic accuracy with the aforementioned baseline segmentation methods using IoU score, F1 score and pixel accuracy.

The quantitative comparison of different vectorization methods using the Hausdorff Distance metric is shown in Table VI, where DP refers to the Douglas-Peucker algorithm, ACM refers to the Active Contour Model-based polygonization algorithm used in [5], and PolyRNN+ refers to the PolyRNN++ network used in [75]. The window vertices predicted by our network have significantly lower error with respect to DP, ACM and PolyRNN+ on all datasets. Especially on the ECP dataset, our method achieves a very low average Hausdorff Distance of only 2.9 pixels.

The qualitative results of the vectorized windows are illustrated in Figure 13. The Douglas-Peucker method exhibits several redundant vertices, and often a not regularized shape. ACM and PolyRNN+ yields more regular shapes, yet the number of vertices is still redundant at some spots, and the positions of vertices are not accurate. By contrast, our method achieves the most regular window shapes and accurate vertices. It should be noted that the proposed vectorization approach is able to correct some false positives introduced in the segmentation step, as shown at the bottom of the façade. Our vectorization network can well handle such cases and does not make prediction at these spots, while the other baseline methods cannot correct or improve wrong segmentations.

As our method takes both segmentation masks and original images into consideration, it can amend for the initial segmentation errors, achieving a more accurate window prediction. Figure 14 depicts an example of the visual improvement in semantic accuracy, where the blue contour represents the contour of the input window mask. The vectorized window vertices are

(a) Original image (b) DeepWindows (c) FFP (d) Our model (e) Ground-truth

Fig. 11: Segmentation results on Graz50 dataset: (a) original image; (b) results of DeepWindows network; (c) results of FFP network (d) results of our segmentation model; (e) ground-truth.

	AP	AP_{50}	AP_{75}	AR	AR_{50}	AR_{75}
Mask R-CNN	55.0	76.2	59.7	62.2	78.1	60.0
UResNet101	61.6	89.1	74.3	71.6	92.1	79.8
FFP_{unet}	43.0	72.7	46.7	47.9	77.0	52.9
Ours_{unet} (without attention)	45.2	74.5	50.4	50.0	78.8	56.0
Ours_{unet} (with attention)	47.1	75.9	52.2	51.9	80.4	57.1
$\text{FFP}_{uresnet101}$	63.5	91.3	76.8	73.0	92.6	80.2
$\text{Ours}_{uresnet101}$ (without attention)	65.6	92.7	78.9	73.4	94.0	82.1
$\text{Ours}_{uresnet101}$ (with attention)	66.9	94.4	80.2	75.6	95.6	83.4

TABLE V: AP and AR results on the merged dataset of our method and other models (unit: %)

	ECP	CMP	Graz50	ParisArtDeco
DP	7.6	10.6	8.3	9.2
ACM	5.8	7.9	7.0	6.8
PolyRNN+	3.6	5.7	7.2	5.3
Ours	2.9	4.8	3.8	4.1

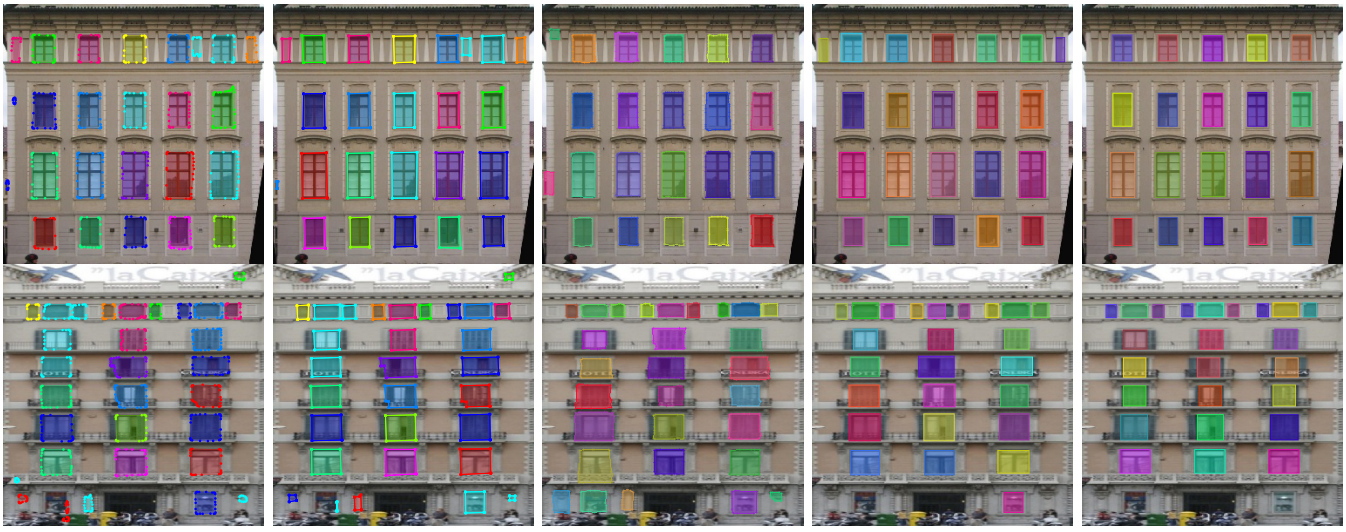
TABLE VI: Comparison of Hausdorff Distance with different vectorization methods on four datasets (unit: pixel). DP refers to the Douglas-Peucker algorithm, ACM refers to the Active Contour Model-based polygonization algorithm used in [5], PolyRNN+ refers to the PolyRNN++ network used in [75], *Ours* refers to the proposed vectorization network

represented as yellow dots and connected by red lines. The Douglas-Peucker algorithm has a negligible influence on the semantic segmentation, while the ACM method exhibits minor improvements, with the results still largely shifted from the ground-truth. By contrast, our method substantially improves the accuracy of the window predictions.

It has been shown that our vectorization method can amend for errors introduced in the input segmentation step, therefore, when the vectorized window objects are converted back to rasters, these exhibit higher semantic accuracy with respect to the initial segmentation masks. In order to validate such improvement, we conducted quantitative comparisons with other segmentation methods on the four data benchmarks, with results listed in Table I, Table II, Table III and Table IV, respectively. Therein, the results of our vectorization method is named *Ours_refine*, which achieve the highest semantic accuracy on all benchmarks.

(a) Original image (b) DeepWindows (c) FFP (d) Our model (e) Ground-truth

Fig. 12: Segmentation results on ParisArtdeco dataset: (a) original image; (b) results of DeepWindows network; (c) results of FFP network (d) results of our segmentation model; (e) ground-truth.



(a) Douglas-Peucker (b) ACM (c) PolyRNN+ (d) Our model (e) Ground-truth

Fig. 13: Sample results of window vectorization: (a) results of Douglas-Peucker algorithm; (b) results of ACM algorithm; (c) results of PolyRNN+ algorithm (d) results of our vectorization model; (e) Ground-truth.

In order to further demonstrate the capacity and limitations of our vectorization method, we report some difficult cases. Figure 15 shows the vectorization results for special window types. The first row shows windows with half-drawn blinds, where the input window mask contains many over-segmentation errors due to the blinds. Our method eliminates several errors in the segmentation, but still has difficulty in fully revising the segmentation errors at the blinds.

(a) Douglas-Peucker (b) ACM (c) Our model (d) Ground-truth

Fig. 14: Improvement in semantic accuracy of our vectorization method: (a) results of Douglas-Peucker algorithm; (b) results of ACM algorithm; (c) results of our vectorization model; (d) Ground-truth. Blue contour represents the original segment contour, while the vectorized window vertices are represented as yellow dots that are connected by red lines.

It should be noted that our method only predicts rectangular windows for all kinds of inputs, which may cause problem for non-rectangular shapes. The second row in Figure 15 shows an example of arc-shaped windows, which are annotated as rectangles in ground-truth as well. It can be seen that the Douglas-Peucker, ACM and PolyRNN+ network do not show higher accuracy than our method, though their outputs are not restricted by the number of vertices.

F. Experiment discussion

The experimental results demonstrate the effectiveness of our method. Regarding semantic segmentation, our model achieves the highest or second highest accuracy compared to state-of-the-art methods, on all datasets and according to all evaluation metrics. The performance on the ParisArtDeco dataset proves that our segmentation network is able to learn the window layout pattern and make correct predictions even in spite of severe occlusions. Regarding window corner vectorization, our vectorization model not only makes regular and sparse predictions for window corners, but also further improves the segmentation accuracy by considering the image features. When compared to the Douglas-Peucker, ACM and PolyRNN+ method, our vectorization network achieves both the highest semantic accuracy and the highest position accuracy on all datasets.

However, the experiments also have some limitations. First, annotations of the four datasets are not precise in some samples. For example, all windows including arc-shaped ones are annotated as rectangles, and a number of annotations are obviously shifted from the actual window locations. Figure 16 illustrates some sample annotation errors in window detection benchmarks, Figures 16a, 16f and 16g show inaccurate annotations for arc-shaped windows, Figures 16b and 16c depict inaccurate position of the annotation, and Figure 16e shows wrong annotations for windows that do not exist. Figures 16d and 16h show windows with large occlusions. Second, the annotation policy is not consistent within each dataset. For example, only in some cases french balconies are annotated as windows. Third, the Graz50 and ParisArtDeco datasets contain only a small number of training samples for the network to learn useful features comprehensively. Furthermore, images can have different sizes, therefore we had to resize all images to the same size and deformed images may vary from the actual appearance, introducing additional challenges for the network.

In this study, we assume that all windows have rectangular shapes, and therefore represent them only by their top-left and bottom-right corners. Such assumption holds for most windows in rectified images: however, when it comes to ground-view images where windows are deformed due to affine transform, the proposed method is no longer applicable. In order to handle such situations, we can modify the output layer of the vectorization network and let it predict four corners instead of only two, so that it can work with any windows having quadrilateral shape.

The source codes for the DAN-PSPNet-L_{sym} and DeepFacade are respectively not available and only compatible with an outdated deep learning framework. Therefore, the original methods cannot be reproduced for comparison and we report their quantitative experimental results from the original papers.

(a) Douglas-Peucker (b) ACM (c) PolyRNN+ (d) Our model (e) Ground-truth

Fig. 15: Vectorization results for special window types: (a) results of Douglas-Peucker algorithm; (b) results of ACM algorithm; (c) results of PolyRNN+; (d) results of our vectorization model; (e) Ground-truth.

VI. Conclusion and Outlook

In this paper we proposed a semantic segmentation network to predict window masks, and a regression network to vectorize the pixel-wise window blobs relying on their corners.

In the segmentation network, we learn a cross field which represents the geometric information of the images in addition to the typical segmentation head, thus improving the overall geometric integrity; secondly, we add attention gates to further improve the learning efficiency. Our segmentation network is efficient as it is constituted by a single FCN. Unlike the training in GANs or RNNs, which is expensive in terms of efforts in tuning and computational resources required, the training of the cross field is straightforward and adds virtually no cost to inference time.

In the window vectorization module, we use the window prediction of the segmentation network together with the original façade image as input, and directly learn the coordinates of the top-left and bottom-right window vertices using a regression neural network adopting a SE-ResNet for feature extraction. The training of the regression network is straightforward and efficient as it is constituted by a single CNN architecture: however, during the inference, the trained model is applied on the ROI of each window individually rather than on the whole image, thing which adds to the computational burden of the inference step.

The quantitative experiments on the benchmark datasets demonstrate that the vectorization network further improves the accuracy and the final results outperform state-of-the-art models significantly. The qualitative experimental results show that our method can achieve more regular and visual pleasing window predictions with respect to other methods.

Although our method has achieved promising results, there are still several challenges left to tackle. First, the vectorization network can only predict the top-left and bottom-right corners, forcing the window to have a rectangular shape, and restricting its application to rectified façade images in which the windows are all rectangular. In order to meet the demands on more diverse street-view images or oblique aerial-view images with heterogeneous window types, we intend to adapt our vectorization network to predict four or more than four vertices of the window, so that any quadrilateral or free-formed windows can be represented. Second, the image data used in our study is acquired from close-range photogrammetry. In the future, we aim to extend it to aerial imagery. Since remote sensing data usually tends to suffer from various degradation, noise effects, or variabilities in the process of imaging [76], coping with the introduced sources of variability will be the focus of our future research.

Acknowledgment

We thank Dr. Daniele Cerra for reviewing this article, and we acknowledge the anonymous reviewers and the editors for their insightful comments.

References

[1] M. Neuhausen, C. Koch, and M. König, "Image-based window detection: an overview," in *Proceedings of Workshop of the European Group for Intelligent Computing in Engineering*, 2016, pp. 1–27.

[2] W. Ma, S. Xu, W. Ma, X. Zhang, and H. Zha, "Progressive feature learning for facade parsing with occlusions," *IEEE Transactions on Image Processing*, 2022.

[3] H. Liu, Y. Xu, J. Zhang, J. Zhu, Y. Li, and S. C. Hoi, "Deepfacade: A deep learning approach to facade parsing with symmetric loss," *IEEE Transactions on Multimedia*, vol. 22, no. 12, pp. 3153–3165, 2020.

[4] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica: the international journal for geographic information and geovisualization*, vol. 10, no. 2, pp. 112–122, 1973.

[5] N. Girard, D. Smirnov, J. Solomon, and Y. Tarabalka, "Polygonal building extraction by frame field learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 5891–5900.

(a) (b) (c) (d)

(e) ECP dataset (f) CMP dataset (g) Graz50 dataset (h) ParisArtDeco dataset

Fig. 16: Sample errors in window detection benchmarks.

[6] P. Zhao, T. Fang, J. Xiao, H. Zhang, Q. Zhao, and L. Quan, "Rectilinear parsing of architecture in urban environment," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 342–349.

[7] P. Müller, G. Zeng, P. Wonka, and L. Van Gool, "Image-based procedural modeling of facades," *ACM Trans. Graph.*, vol. 26, no. 3, p. 85, 2007.

[8] F. Han and S.-C. Zhu, "Bottom-up/top-down image parsing with attribute grammar," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 1, pp. 59–73, 2008.

[9] R. Gadde, R. Marlet, and N. Paragios, "Learning grammars for architecture-specific facade parsing," *International Journal of Computer Vision*, vol. 117, no. 3, pp. 290–316, 2016.

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

[11] M. Kozinski, R. Gadde, S. Zagoruyko, G. Obozinski, and R. Marlet, "A mrf shape prior for facade parsing with occlusions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2820–2828.

[12] K. Rahmani, H. Huang, and H. Mayer, "Facade segmentation with a structured random forest." *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, vol. 4, 2017.

[13] Z. Li, L. Zhang, P. T. Mathiopoulos, F. Liu, L. Zhang, S. Li, and H. Liu, "A hierarchical methodology for urban facade parsing from tls point clouds," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 123, pp. 75–93, 2017.

[14] M. Mathias, A. Martinović, and L. Van Gool, "Atlas: A three-layered approach to facade parsing," *International Journal of Computer Vision*, vol. 118, no. 1, pp. 22–48, 2016.

[15] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[16] H. Liu, J. Zhang, J. Zhu, and S. C. Hoi, "Deepfacade: A deep learning approach to facade parsing," 2017.

[17] M. Dai, W. O. Ward, G. Meyers, D. D. Tingley, and M. Mayfield, "Residential building facade segmentation in the urban environment," *Building and Environment*, vol. 199, p. 107921, 2021.

[18] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[19] W. Ma, W. Ma, S. Xu, and H. Zha, "Pyramid alknet for semantic parsing of building facade image," *IEEE Geoscience and Remote Sensing Letters*, vol. 18, no. 6, pp. 1009–1013, 2020.

[20] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[21] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[22] O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz *et al.*, "Attention u-net: Learning where to look for the pancreas," *arXiv preprint arXiv:1804.03999*, 2018.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[24] M. Yin, Z. Yao, Y. Cao, X. Li, Z. Zhang, S. Lin, and H. Hu, "Disentangled non-local neural networks," in *European Conference on Computer Vision*. Springer, 2020, pp. 191–207.

[25] A. Srinivas, T.-Y. Lin, N. Parmar, J. Shlens, P. Abbeel, and A. Vaswani, "Bottleneck transformers for visual recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 16 519–16 529.

[26] J. Gu, H. Hu, L. Wang, Y. Wei, and J. Dai, "Learning region features for object detection," in *Proceedings of the european conference on computer vision (ECCV)*, 2018, pp. 381–395.

[27] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[28] G. Zhang, Y. Pan, and L. Zhang, "Deep learning for detecting building façade elements from images considering prior knowledge," *Automation in Construction*, vol. 133, p. 104016, 2022.

[29] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[30] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning repre-

sentations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[31] Q. Yu, J. Malaeb, and W. Ma, "Architectural facade recognition and generation through generative adversarial networks," in *2020 International Conference on Big Data & Artificial Intelligence & Software Engineering (ICBASE)*. IEEE, 2020, pp. 310–316.

[32] A. H. Abdulnabi, B. Shuai, Z. Zuo, L.-P. Chau, and G. Wang, "Multimodal recurrent neural networks with information transfer layers for indoor scene labeling," *IEEE Transactions on Multimedia*, vol. 20, no. 7, pp. 1656–1671, 2017.

[33] D. Hong, L. Gao, N. Yokoya, J. Yao, J. Chanussot, Q. Du, and B. Zhang, "More diverse means better: Multimodal deep learning meets remote-sensing imagery classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 5, pp. 4340–4354, 2020.

[34] X. Wu, D. Hong, and J. Chanussot, "Convolutional neural networks for multimodal remote sensing data classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–10, 2021.

[35] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, and J. Chanussot, "Graph convolutional networks for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 7, pp. 5966–5978, 2020.

[36] F. De Goes, D. Cohen-Steiner, P. Alliez, and M. Desbrun, "An optimal transport approach to robust reconstruction and simplification of 2d shapes," in *Computer Graphics Forum*, vol. 30, no. 5. Wiley Online Library, 2011, pp. 1593–1602.

[37] P. Hough, "Method and means for recognising complex patterns', 3069654, 1962," *US patent*.

[38] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International journal of computer vision*, vol. 1, no. 4, pp. 321–331, 1988.

[39] A. Hatamizadeh, D. Sengupta, and D. Terzopoulos, "End-to-end trainable deep active contour models for automated image segmentation: Delineating buildings in aerial imagery," in *European Conference on Computer Vision*. Springer, 2020, pp. 730–746.

[40] L. Castrejon, K. Kundu, R. Urtasun, and S. Fidler, "Annotating object instances with a polygon-rnn," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5230–5238.

[41] D. Acuna, H. Ling, A. Kar, and S. Fidler, "Efficient interactive annotation of segmentation datasets with polygon-rnn++," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 859–868.

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

[43] C.-K. Li, H.-X. Zhang, J.-X. Liu, Y.-Q. Zhang, S.-C. Zou, and Y.-T. Fang, "Window detection in facades using heatmap fusion," *Journal of Computer Science and Technology*, vol. 35, no. 4, pp. 900–912, 2020.

[44] B. Xiao, H. Wu, and Y. Wei, "Simple baselines for human pose estimation and tracking," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 466–481.

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

[46] G. Huang, D. Chen, T. Li, F. Wu, L. Van Der Maaten, and K. Q. Weinberger, "Multi-scale dense convolutional networks for efficient prediction," *arXiv preprint arXiv:1703.09844*, vol. 2, p. 2, 2017.

[47] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[48] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[49] S. Jetley, N. A. Lord, N. Lee, and P. H. Torr, "Learn to pay attention," *arXiv preprint arXiv:1804.02391*, 2018.

[50] N. Ray, B. Vallet, L. Alonso, and B. Levy, "Geometry-aware direction field processing," *ACM Transactions on Graphics (TOG)*, vol. 29, no. 1, pp. 1–11, 2009.

[51] A. Hertzmann and D. Zorin, "Illustrating smooth surfaces," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 2000, pp. 517–526.

[52] D. Bommes, B. Lévy, N. Pietroni, E. Puppo, C. Silva, M. Tarini, and D. Zorin, "Quad-mesh generation and processing: A survey," in *Computer Graphics Forum*, vol. 32, no. 6. Wiley Online Library, 2013, pp. 51–76.

[53] D. Bommes, H. Zimmer, and L. Kobbelt, "Mixed-integer quadrangulation," *ACM Transactions On Graphics (TOG)*, vol. 28, no. 3, pp. 1–10, 2009.

[54] D. Panozzo, E. Puppo, M. Tarini, and O. Sorkine-Hornung, "Frame fields: Anisotropic and non-orthogonal cross fields additional material," *Proceedings of the ACM TRANSACTIONS ON GRAPHICS (PROCEEDINGS OF ACM SIGGRAPH*, 2014.

[55] O. Diamanti, A. Vaxman, D. Panozzo, and O. Sorkine-Hornung, "Designing n-polyvector fields with complex polynomials," in *Computer Graphics Forum*, vol. 33, no. 5. Wiley Online Library, 2014, pp. 1–11.

[56] M. Bessmeltsev and J. Solomon, "Vectorization of line drawings via polyvector fields," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 1, pp. 1–12, 2019.

[57] M. Taktasheva, A. Matveev, A. Artemov, and E. Burnaev, "Learning to approximate directional fields defined over 2d planes," in *International Conference on Analysis of Images, Social Networks and Texts*. Springer, 2019, pp. 367–374.

[58] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[59] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European conference on computer vision*. Springer, 2016, pp. 630–645.

[60] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[61] J. T. Szcześniak, Y. Q. Ang, S. Letellier-Duchesne, and C. F. Reinhart, "A method for using street view imagery to auto-extract window-to-wall ratios and its relevance for urban-level daylighting and energy simulations," *Building and Environment*, vol. 207, p. 108108, 2022.

[62] G. Kong and H. Fan, "Enhanced facade parsing for street-level images using convolutional neural networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 12, pp. 10 519–10 531, 2020.

[63] R. Tyleček and R. Šára, "Spatial pattern templates for recognition of objects with regular structure," in *Proc. GCPR*, Saarbrucken, Germany, 2013.

[64] M. Neuhausen, M. Obel, A. Martin, P. Mark, and M. König, "Window detection in facade images for risk assessment in tunneling," *Visualization in Engineering*, vol. 6, no. 1, pp. 1–16, 2018.

[65] W. Ma and W. Ma, "Deep window detection in street scenes," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 14, no. 2, pp. 855–870, 2020.

[66] J. Femiani, W. R. Para, N. Mitra, and P. Wonka, "Facade segmentation in the wild," *arXiv preprint arXiv:1805.08634*, 2018.

[67] H. Riemenschneider, U. Krispel, W. Thaller, M. Donoser, S. Havemann, D. Fellner, and H. Bischof, "Irregular lattices for complex shape grammar facade parsing," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 1640–1647.

[68] "Learning grammars for architecture-specific facade parsing," Tech. Rep., 2014.

[69] W. Ma, W. Ma, and S. Xu, "Deep facade parsing with occlusions," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 16, no. 2, pp. 524–543, 2022.

[70] A. E. Maxwell, T. A. Warner, and L. A. Guillén, "Accuracy assessment in convolutional neural network-based deep learning remote sensing studies—part 1: Literature review," *Remote Sensing*, vol. 13, no. 13, p. 2450, 2021.

[71] Y. Sasaki *et al.*, "The truth of the f-measure," *Teach tutor mater*, vol. 1, no. 5, pp. 1–5, 2007.

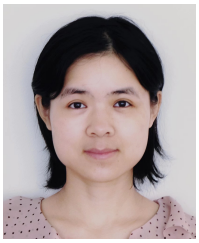
[72] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[73] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, "Comparing images using the hausdorff distance," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 15, no. 9, pp. 850–863, 1993.

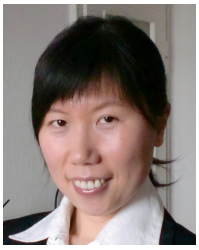
[74] Y. Sun, S. Malihi, H. Li, and M. Maboudi, "Deepwindows: Windows instance segmentation through an improved mask r-cnn using spatial attention and relation modules," *ISPRS International Journal of Geo-Information*, vol. 11, no. 3, p. 162, 2022.

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

[76] D. Hong, N. Yokoya, J. Chanussot, and X. X. Zhu, "An augmented linear mixing model to address spectral variability for hyperspectral unmixing," *IEEE Transactions on Image Processing*, vol. 28, no. 4, pp. 1923–1938, 2018.



Xiangyu Zhuo received her B.S. degree in Geodesy and Geomatrics in 2012 and M. Eng. degree in Photogrammetry in 2014 both from Wuhan University, China. In 2019, she received her Ph.D. degree in Photogrammetry and Remote Sensing from Technical University of Munich (TUM), Germany. Since 2018, she has been with the Photogrammetry and Image Analysis Department, Remote Sensing Technology Institute, German Aerospace Center, Wessling, Germany. Her research interests include UAV-based photogrammetry, aerial image segmentation using machine learning and deep learning techniques.



Jiaojiao Tian received her B.S. degree in geoinformation systems from the China University of Geoscience, Beijing, in 2006, her M. Eng. degree in cartography and geoinformation at the Chinese Academy of Surveying and Mapping, Beijing, in 2009, and her Ph.D. degree in mathematics and computer science from Osnabrück University, Germany, in 2013. Since 2009, she has been with the Photogrammetry and Image Analysis Department, Remote Sensing Technology Institute, German Aerospace Center, Wessling, Germany, where she is currently head of the 3D Modeling Group. In 2011, she was a guest scientist with the Institute of Photogrammetry and Remote Sensing, ETH Zürich, Switzerland. Her research interests include 3D change detection, digital surface model (DSM) generation, 3D point cloud semantic segmentation, object extraction, and DSM-assisted building reconstruction and classification.

Friedrich Fraundorfer received the Ph.D. degree in computer science from the Graz University of Technology (TU Graz), Austria, in 2006. He had a postdoctoral stay at the University of Kentucky, the University of North Carolina at Chapel Hill, and ETH Zürich. From 2012 to 2014, he was the Deputy Director of the Chair of Remote Sensing Technology, Technical University of Munich. He is currently an Associate Professor with the Institute of Computer Graphics and Vision, TU Graz and associated with the Photogrammetry and Image Analysis Department, Remote Sensing Technology Institute, German Aerospace Center (DLR). His main research interests include 3D computer vision, robot vision, and machine learning techniques.