Element-Arrangement Context Network for Facade Parsing

Yan Tao, Yiteng Zhang, Xuejin Chen
National Engineering Laboratory for Brain-inspired Intelligence Technology and Application
University of Science and Technology of China, Hefei, Anhui, China
{ty990813, zytcc}@mail.ustc.edu.cn, xjchen@ustc.edu.cn

Abstract

Facade parsing aims to decompose a building facade image into semantic regions of the facade objects. Considering each architectural element on a facade as a parameterized rectangle, we formulate the facade parsing task as object detection, allowing overlapping and nesting, which will support structural 3D modeling and editing for further applications. In contrast to general object detection, the spatial arrangement regularity and appearance similarity between the facade elements of the same category provide valuable context for accurate element localization. In this paper, we propose to exploit spatial arrangement regularity and appearance similarity of facade elements in a detection framework. Our Element-Arrangement Context Network (EACNet) consists of two unidirectional attention branches, one to capture the column-context and the other to capture row-context to aggregate element-specific features from multiple instances on the facade. We conduct extensive experiments on four facade datasets. The proposed EACNet produces more concise and structured parsing results than existing facade segmentation methods. Both quantitative and qualitative evaluation results demonstrate the effectiveness of our dual unidirectional attention branches to parse facade elements.

Keywords: Facade parsing, detection, layout regularity, spatial context.

1. Introduction

Facade parsing aims to find regions of building facade components and annotate them with distinctive semantic categories (*e.g*. window, sill, balcony, and molding) in a given street-view facade image. This task potentially supports many real-world applications, especially for urban street reconstruction. However, facade parsing faces many challenges in natural urban scenes. Firstly, the facade style varies a lot among buildings. The diversity of texture and element structure makes it difficult to generate robust and accurate parsing results. Moreover, parsing a facade image may be more challenging due to shadows, illumination, perspective effect, and occlusions caused by cluttered objects. Most importantly, since the arrangement regularity of various building facade elements is naturally existing and widely presented, the parsing results should globally follow regular arrangement.

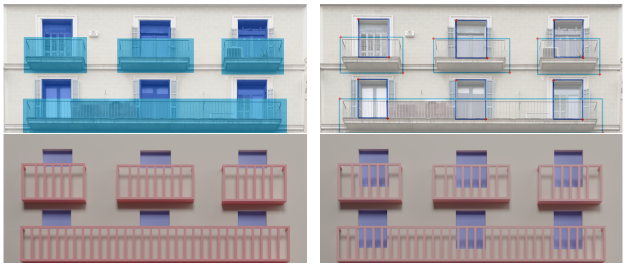
(a) Semantic region masks (b) Parameterized bounding boxes

Figure 1. Our method aims to produce compact parameterized bounding boxes instead of dense pixel-wise semantic labels (a). Based on the parameterized bounding boxes, 3D facade models can be generated more efficiently while allowing structural overlapping of multiple elements (b).

Facade parsing has been attracting lots of interest over the past few years. Traditional approaches usually combine architectural priors with image segmentation. The facade structural priors, such as element sizes, the spacing between elements, and hard alignment constraints, are encoded in the parsing procedure to introduce essential architectural information. Some grammar-based methods [26, 35, 38, 37, 43] perform top-down parsing procedures to model facades with predefined primitive shapes and grammar rules. Some other works [4, 24, 25] utilize low-level information extracted by per-pixel classification to produce facade segmentation. Though these methods consider facade regularities, they rely highly on hand-crafted knowledge priors. Consequently, the global holistic structural information is not always at work, especially for complex scenes.

Recent progress in deep learning and deep convolutional neural networks has made it possible to extract and utilize high-level features and global structural information of a building facade. Several deep learning-based facade parsing [34, 23, 22] treat facade parsing as a semantic segmentation

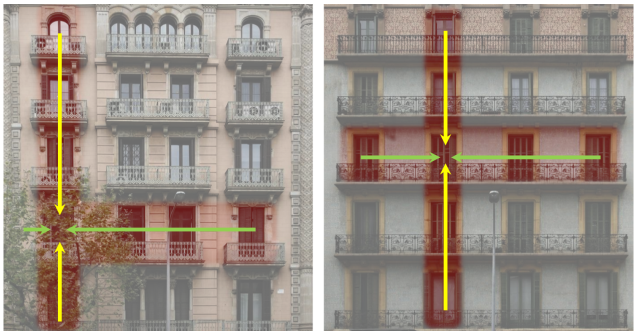

Figure 2. Facade layouts present strong regularity, as the architectural elements are well-aligned both vertically and horizontally. The spatial correlation between facade elements in the two directions provides valuable context for facade element detection.

problem and employ popular CNNs to achieve better performance. DeepFacade [23, 22] illustrates the importance of facade structural priors and introduces the shape symmetry of facade elements as a constraint, aiming to produce more regular segmentation results. Bounding boxes are considered as auxiliary data to refine the shape of segmentation regions in their work. However, the element symmetry and the facade layout regularity, which are crucial for obtaining complete and reasonable facade parsing results, are ignored.

Though a pixel collection can flexibly describe freeform object shapes in the semantic segmentation framework, we argue that dense semantic region masks are not the most appropriate representation for facade parsing. First, objects on a facade usually appear as symmetrical quadrilaterals in a rectified street-view image. However, it is non-trivial to exploit this geometric property efficiently in pixel-wise segmentation approaches. Second, facade segmentation usually results in a labeled mask image where each pixel is assigned a single category. However, facade components are not always disjoint. Overlapping frequently happens among various categories such as windows and blinds. Figure 1 shows a typical case where the balconies overlap with the bottom region of their nearby windows. The dense single-category assignment makes the rendering and modeling of the overlapping regions much more complicated, even resulting in the structure loss of the nesting regions. In contrast, we propose a detection-based framework to decompose facade images while supporting overlapping facade elements and involving the global layout context to generate more regular facade arrangements.

The element layout usually presents a strong regularity and shows a grid-like element arrangement. A facade element is usually significantly correlated with facade objects in the same row or column. Figure 2 illustrates our motivation. The window that is partially occluded by vegetation can be accurately localized based on its related horizontal and vertical element groups. Based on this observation, we leverage the spatial regularity of facade layout in our element detection framework. We propose an Element-Arrangement Context Network (EACNet) to exploit the arrangement regularity among facade elements arranged in the same row or column. We conducted extensive experiments to evaluate the effectiveness of our method. Our EACNet achieves top performance on the Graz50 [31] and ECP [38] datasets. Even on the challenging CMP [39] dataset, our EACNet effectively captures element-arrangement spatial context and significantly facilitates the facade parsing task.

2. Related Work

We discuss related work on traditional facade parsing and CNN-based facade segmentation. We also discuss several typical object detection approaches. In addition, we discuss attention mechanisms and several related general self-attention schemes.

Facade parsing and modeling have been extensively studied in computer graphics and computer vision. There are two mainstreams of traditional methods: utilizing grammar-based recognition or following conventional image segmentation pipeline. Grammar-based approaches model facades according to a set of parametric grammars, based on which the procedural modeling procedure can utilize image analysis techniques to derive a hierarchical facade subdivision from an image [1, 5, 16]. Similar ideas can be found in other methods that target general procedural modeling for structural objects [11, 36, 45]. Another stream of facade layout generation methods is segmenting the input images. Several approaches incorporate traditional machine learning to fit the procedural modeling pipeline [37, 29, 8]. Some others utilize architectural principles to optimize facade segmentation [43, 4, 3, 25, 17], aiming to produce more regular segmentation regions.

With the rapid development of deep learning, CNN-based semantic segmentation frameworks have been adopted for facade parsing. Directly applying the fully-convolutional networks for semantic segmentation into facade segmentation [34, 7] generates pixel-wise label prediction. Subsequently, object symmetry is taken into account to refine the segmented region boundaries in DeepFacade [23] that uses a loss function to penalize segmentation regions that are not horizontally and vertically symmetric. Its extension work [22] adds another loss term that forces the window regions to match the rectangular shapes obtained by a pre-trained auxiliary Mask R-CNN [12]. While they focus on improving the regularity of single element shapes, our approach naturally ensures the single shape regularity and exploits global layout regularity with a well-designed attention scheme.

Object detection pipelines directly output rectangular boxes for objects in an image. Many two-stage region de-

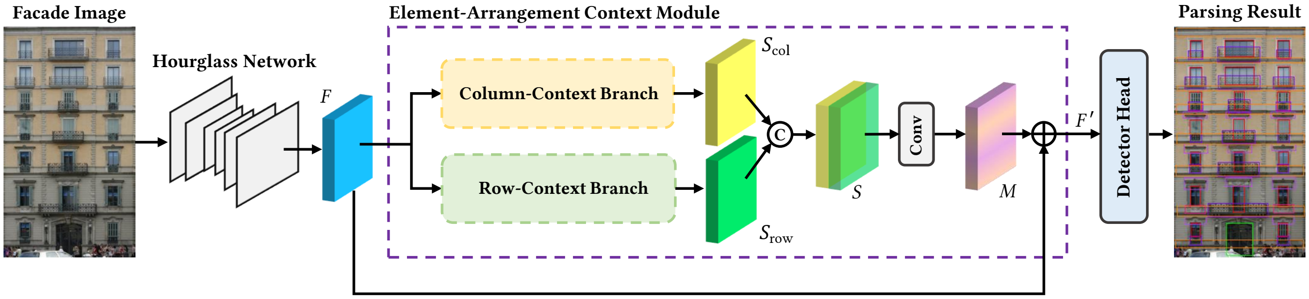

Figure 3. Overview of the proposed EACNet. After extracting feature maps from the input image using an hourglass network, we aggregate the spatial context between facade elements by the proposed element-arrangement context module. Two rectilinear context branches are designed to capture the vertical and horizontal correlations between elements, and the contexts in two directions are aggregated to enhance the local features. Finally, a detector head is attached to obtain the final facade parsing results from the aggregated feature maps. "©" denotes feature concatenation, and "$\oplus$" denotes element-wise addition.

tection networks have been proposed [10, 9]. More recently, keypoint estimation is utilized to locate objects for one-stage detection. CornerNet [19] detects objects by localizing a pair of key points and groups them by using associative embedding [27]. CenterNet [49] treats object center as a single shape-agnostic anchor, detecting an object by extracting a center point, and thus needs not any keypoint grouping steps. Based on the one-stage detection framework, our EACNet is designed specifically for facade parsing by incorporating spatial facade layout regularity.

Self-attention was first introduced in the pioneering work [40] to enhance the representation capability of neural networks and now is widely used for various tasks. However, self-attention suffers from quadratic computation and memory cost, which is particularly challenging for images. Recently, many efforts have been made to investigate sparse and memory-efficient forms, including hierarchical attention [44], clustering-based sparse attention [32], attention to sparse keypoints only [33], attention to image patches instead of pixels [6] and attention with linear complexity [41]. These methods can greatly reduce additional computation and memory costs and make self-attention more efficient. For computer vision tasks, SENet [14] models channel-wise relationships in an attention mechanism. PSANet [48] learns two global attention maps to aggregate contextual information for each position in the feature maps adaptively. The non-local Network [42] generates a huge attention map by calculating the correlation matrix between each spatial point in the feature maps. However, the computation and memory costs for obtaining the attention maps in these methods are significantly high. CCNet [15] develops a criss-cross attention module that captures contextual information in criss-cross paths instead of the whole image and then employs a recurrent operation to harvest full-image dependencies. Inspired by CCNet [15], we further decompose the criss-cross correlation into two independent uni-directional attention branches that only capture long-range dependence between elements aligned in the same row and column separately, considering the spatial regularity of facade elements. This separation explicitly brings structural priors for the spatial correlation between pixels and makes our network more efficient and precise by considering the column-wise and row-wise distinction.

3. Our Approach

In this section, we first introduce the framework of our Element-Arrangement Context Network (EACNet). Then we describe the proposed Element-Arrangement Context Module (EACM) in detail, including the row and column context branches that capture spatial context to enforce the arrangement regularity of the facade elements.

3.1. Network Architecture

Figure 3 shows the overview of the proposed EACNet. Given an input facade image, an hourglass convolutional neural network [28] is employed as the backbone that downsamples the input image by 4 times and extracts feature maps F with spatial size $H \times W$ from the input image. The feature maps $F \in \mathbb{R}^{C \times H \times W}$ are then fed into an element-arrangement context module that learns the correlations between a position on the facade and all different positions in the same row and the same column. The long-range dependencies in the two axis-aligned directions are crucial for localizing facade objects because they show strong repetitiveness and alignment regularity in structure. The two branches in EACM produce feature maps $S_{\mathrm{col}} \in \mathbb{R}^{C \times H \times W}$ and $S_{\mathrm{row}} \in \mathbb{R}^{C \times H \times W}$ that collect spatial context in a single column and row, respectively. The feature maps S_{col} and S_{row} are concatenated and fed to a convolutional layer that acts as feature adaptation. The produced feature maps M are added to the image features F to enhance the representation of each position. The enhanced features F' are fed into a detector head to predict the bounding boxes that represent the parsing results.

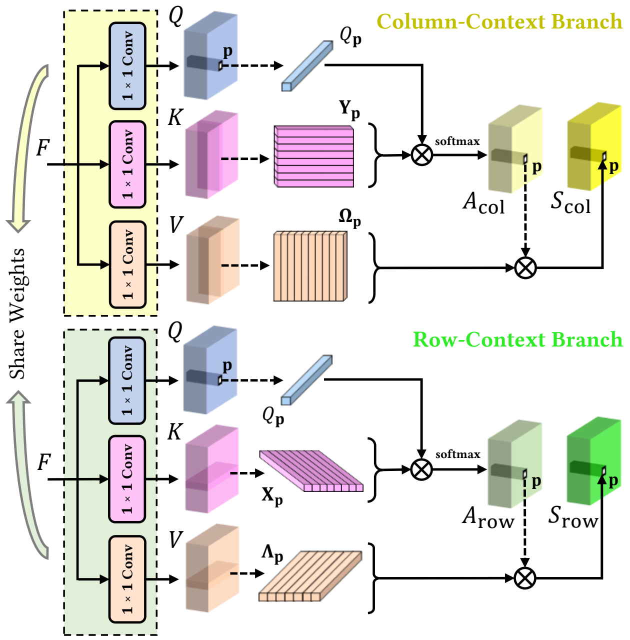

Figure 4. The column-context branch and row-context branch of the proposed element-arrangement context module. They both take feature maps produced by the backbone network to harvest spatial contextual information with shared weights of convolutional layers."⊗" denotes matrix multiplication.

3.2. Element-Arrangement Context Module

In the facade parsing task, it is crucial to exploit the priors of facade structure to promote the parsing quality. To incorporate man-made rules into an end-to-end CNN, existing CNN-based facade parsing methods either restrict the object shape by using symmetric constraint [23, 22] or using atrous convolution [2] to capture nonlocal context. But they seldom take advantage of the holistic facade structure efficiently. In contrast, we propose a novel element-arrangement context module to exploit the element-arrangement regularity and appearance similarity.

It is a fact that the facade elements share strong repetitiveness and alignments in structure. To explicitly leverage the arrangement regularity of the facade elements, we propose an EACM that guides the network focus on the facade elements aligned in the same row and the same column. As shown in Figure 3, the proposed EACM contains two branches, the column-context branch and the row-context branch, which collect element-arrangement spatial context in two directions. They are similar in structure but correspond to different element arrangement directions. To the best of our knowledge, this is the first attempt to employ self-attention to incorporate the facade layout structural regularity into a facade parsing network. Next, we introduce the details of these two context branches.

Following the self-attention mechanism, as illustrated in Figure 4, we first apply three parallel convolutional layers with 1×1 filters on features F to obtain query features Q, key features K, and value features V, with shape $C \times H \times W$. The two branches of EACM both use Q, K, and V to generate contextual features (*i.e.* they share the weights of the convolutional layers). For a point $\mathbf{p} = (i, j)$, the column branch calculates the correlations between $\mathbf{p}$ and other positions in the j-th column, and the row branch calculates the correlations between $\mathbf{p}$ and positions in the i-th row. For a query vector $Q_\mathbf{p} \in \mathbb{R}^{C \times 1}$ in features Q, we extract key vectors from features K along the i-th row and j-th column separately, which gives two sets of feature vectors:

$$\begin{aligned} \mathbf{X_p} &= \left\{K_{(i,1)}, K_{(i,2)}, \dots, K_{(i,j)}, \dots, K_{(i,W)}\right\} \\ \mathbf{Y_p} &= \left\{K_{(1,j)}, K_{(2,j)}, \dots, K_{(i,j)}, \dots, K_{(H,j)}\right\} \end{aligned} \tag{1}$$

The cardinal number of the obtained vector sets $\mathbf{X_p}$ and $\mathbf{Y_p}$ are W and H respectively. In the column branch, the correlations between $\mathbf{p}$ and its corresponding column-path positions can be calculated and collected in a vector ${}^{\text{c}}A_\mathbf{p} \in \mathbb{R}^{H \times 1}$ located in attention maps A_col, which is defined as

$${}^{\text{c}}A_\mathbf{p}^{(k)} = \frac{\exp\left(Q_\mathbf{p}^T \mathbf{Y}_\mathbf{p}^{(k)}\right)}{\sum_{t=1}^{|\mathbf{Y_p}|} \exp\left(Q_\mathbf{p}^T \mathbf{Y}_\mathbf{p}^{(t)}\right)}, \tag{2}$$

where ${}^{\text{c}}A_\mathbf{p}^{(k)}$ is the k-th element of vector ${}^{\text{c}}A_\mathbf{p}$, and $\mathbf{Y}_\mathbf{p}^{(k)}$ is the k-th feature vector of set $\mathbf{Y_p}$.

In the row branch, similar to the calculation of A_col, we calculate the attention maps A_row, where the vector located at point $\mathbf{p}$ is defined as

$${}^{\text{r}}A_\mathbf{p}^{(k)} = \frac{\exp\left(Q_\mathbf{p}^T \mathbf{X}_\mathbf{p}^{(k)}\right)}{\sum_{t=1}^{|\mathbf{X_p}|} \exp\left(Q_\mathbf{p}^T \mathbf{X}_\mathbf{p}^{(t)}\right)}, \tag{3}$$

where ${}^{\text{r}}A_\mathbf{p}^{(k)}$ is the k-th element of the vector ${}^{\text{r}}A_\mathbf{p} \in \mathbb{R}^{W \times 1}$, and $\mathbf{X}_\mathbf{p}^{(k)}$ is the k-th feature vector of set $\mathbf{X_p}$.

After obtaining attention maps A_row and A_col that measure correlations in row and column paths, the module extracts values in row-column paths from features V over the spatial dimension for further context aggregation. For a point $\mathbf{p} = (i, j)$, two sets $\mathbf{\Lambda_p}$ and $\mathbf{\Omega_p}$ can be obtained, both of which consist of C vectors. The c-th element of $\mathbf{\Lambda_p}$ and $\mathbf{\Omega_p}$ are defined as

$$\begin{aligned} \mathbf{\Lambda}_\mathbf{p}^{(c)} &= (V_{ci0}, V_{ci1}, \dots, V_{ciW})^T, \\ \mathbf{\Omega}_\mathbf{p}^{(c)} &= (V_{c0j}, V_{c1j} \dots, V_{cHj})^T, \end{aligned} \tag{4}$$

where V_{cij} denotes the value located at (i, j) of the c-th channel of the feature maps V.

The elements of correlation vectors ${}^{\text{c}}A_\mathbf{p}$ and ${}^{\text{r}}A_\mathbf{p}$ are separately used as the weights of vectors $\mathbf{\Omega}_\mathbf{p}^{(c)}$ and $\mathbf{\Lambda}_\mathbf{p}^{(c)}$ for

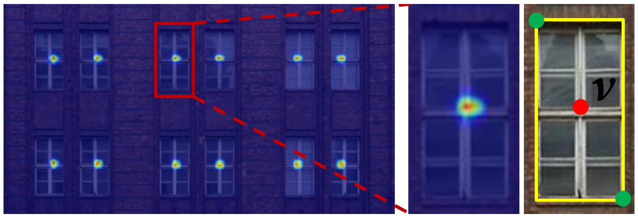

Figure 5. Facade elements typically have highly symmetric rectangular shapes. We encoder a facade object by a center point and its size parameters and predict heatmaps for the center point.

conducting spatial context aggregation at position $\mathbf{p}$, which generates ${}^{\mathrm{c}}S_{\mathbf{p}} \in \mathbb{R}^{C\times 1}$ and ${}^{\mathrm{r}}S_{\mathbf{p}} \in \mathbb{R}^{C\times 1}$ as follows:

$$\begin{aligned} {}^{\mathrm{c}}S_{\mathbf{p}}^{(c)} &= {}^{\mathrm{c}}A_{\mathbf{p}}^{T}\mathbf{\Omega}_{\mathbf{p}}^{(c)}, \\ {}^{\mathrm{r}}S_{\mathbf{p}}^{(c)} &= {}^{\mathrm{r}}A_{\mathbf{p}}^{T}\mathbf{\Lambda}_{\mathbf{p}}^{(c)}. \end{aligned} \tag{5}$$

Collecting spatial context at different positions finally gives contextual features S_{row} and S_{col}, both with shape $C \times H \times W$. Then, they are used to enhance the local feature maps. To exploit the element-arrangement regularity and appearance similarity of both row and column as a guidance, S_{col} and S_{row} are concatenated and fused together to produce integrated contextual features $M \in \mathbb{R}^{C\times H\times W}$. The contextual information is then added to features F to produce updated feature maps F' as

$$F' = \omega(S) + F, \tag{6}$$

where ω is a projection function implemented by a convolutional layer with 1×1 kernel size. $S \in \mathbb{R}^{2C\times H\times W}$ is produced by concatenating S_{col} and S_{row} together.

3.3. Detector Head

The enhanced feature maps F' are fed into a detector head to obtain the final facade parsing results. As shown in Figure 5, it is effective to represent a symmetric facade element by a center and its width and height. We employ CenterNet [49] which models an object bounding box with a center point and object size in our EACNet. The detector head consists of three branches. Each branch applies convolutional layers on F' to generate a set of heatmaps for center location prediction for each element category, local offset prediction, and object size prediction, respectively. The center location prediction branch generates $\hat{E} \in \mathbb{R}^{C'\times H\times W}$, where C' is the number of categories of the facade elements. The value of $\hat{E}_{cij}$ at location $\mathbf{p} = (i, j)$ is the score for class c in the predicted heatmaps. The local offset prediction branch generates $\hat{O} \in \mathbb{R}^{2\times H\times W}$, which is used as a slight adjustment of the center location. The object size prediction branch generates $\hat{U} \in \mathbb{R}^{2\times H\times W}$, which gives the height and width of an object. To obtain the coordinates of center points, a 3×3 max-pooling layer is applied on $\hat{E}$ for peak extraction.

During training, we use a pixel-wise logistic regression with focal loss [20] for center location prediction:

$$L_{\mathrm{p}} = -\frac{1}{N}\sum_{c,i,j}\begin{cases}(1-\hat{E}_{cij})^{\alpha}\log(\hat{E}_{cij}), & \text{if } E_{cij} = 1\\ (1-E_{cij})^{\beta}(\hat{E}_{cij})^{\alpha}\log(1-\hat{E}_{cij}), & \text{otherwise}\end{cases} \tag{7}$$

where E is the ground truth and N is the number of facade objects. Both α and β are hyper-parameters that control the contribution of each point. We set $\alpha = 2$ and $\beta = 4$ in all experiments. The local offset and object size are both trained with L1 distance as loss function. The total loss function is

$$L = L_{\mathrm{p}} + \frac{\lambda}{N}\sum_{k}\mathcal{L}_1(\hat{O}_k, O_k) + \frac{\mu}{N}\sum_{k}\mathcal{L}_1(\hat{U}_k, U_k), \tag{8}$$

where O_k and U_k are the location offset and the object size of the k-th element, respectively. The scale factors λ and μ are used for weight adjustment.

4. Experiment Results and Discussions

In this section, we first introduce four facade datasets and present the corresponding evaluation metrics and the training details. Then, we compare our facade parsing method with existing segmentation-based facade parsing works. A series of ablation experiments are also conducted to demonstrate the effectiveness of the proposed EACM.

4.1. Datasets and Evaluation Metrics

Four public facade datasets are used in our experiments, including ECP [38], CMP [39], Graz50 [31], and eTRIMS [22]. The first three contain rectified facade images with their semantic label masks. Images in the eTRIMS dataset are not rectified.

The ECP facade dataset [38] consists of 104 well-rectified building facade images. All the images contain facades from Paris and share similar architectural styles. The pixel annotations contain eight classes, including *window, wall, balcony, door, shop, sky, chimney,* and *roof.* Since there are some categories not belonging to facade elements, we choose *window, balcony,door,* and *shop* for evaluation. Since the ECP dataset does not have bounding box annotation, we perform contour fitting on the provided semantic masks to generate the bounding box for each element. For overlapping elements, we adjust their bounding box sizes to match the corresponding regions. We follow [22] to divide the dataset, using 80 images for training and 24 for testing.

The Graz50 facade dataset [31] contains 50 facade images with multiple building styles. Similar to the ECP dataset, the Graz50 dataset only contains rectangular areas labeled as ground truth semantic masks. Contour fitting is also applied to obtain suitable bounding box annotation. The provided data contains two facade element classes, *window*, and *door*, which are both used in our experiments. We

Method	Pixel Accuracy (%)			IoU (%)	
	Window	Door	Avg.	Window	Door
Koziński *et al.* [18]	82	50	66.0	–	–
Koziński *et al.* [17]	84	60	72.0	–	–
Cohen *et al.* [3]	85	64	74.5	–	–
Rahmani *et al.* [29]	79.3	79.1	79.2	–	–
DeepFacade-V1 [23]	87.7	88.2	87.9	–	–
Rahmani *et al.* [30]	83.7	**93.8**	88.8	–	–
DeepFacade-V2 [22]	88.8	89.1	88.9	71.3	56.5
Ours	**89.9**	87.8	**88.9**	**80.9**	**73.8**

Table 1. Quantitative comparison with state-of-the-art facade parsing methods on the Graz50 dataset.

follow the dataset division strategy used in [22], using 30 images for training and 20 images for testing.

The CMP facade dataset [39] contains 606 rectified images of facades with diverse architectural styles. The dataset is split into two parts that consist of 378 and 228 images. The latter part contains more irregular and non-planar facades that often have substantial occlusion from vegetation, making the CMP dataset very challenging. The annotation of this dataset is a set of rectangles with class labels and allows overlapping and nesting. The dataset includes 12 specified classes. In our experiment, we use six categories that belong to facade elements, including *sill, balcony, door, molding, window*, and *cornice*. We use 484 images that are randomly selected from two subsets for training. The remaining 122 facade images are used for testing.

4.2. Training Settings

The Hourglass backbone [28] used in our EACNet is initialized using the weights of a model pre-trained on the COCO dataset [21]. The remaining part of the network is initialized randomly. In all experiments, the network is trained on a single GPU, using an Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and the initial learning rate as 0.0001, 0.0002, 0.0002, and 0.0004 for CMP, ECP, eTRIMS, and Graz50 datasets, respectively. The scale factors λ and μ in Eq. 8 are set to 1 and 0.1 respectively. We use batch size 4 for all three datasets. For ECP, eTRIMS, and Graz50, the network is trained for 120, 100, and 80 epochs respectively. For CMP, the network is trained for 200 epochs, and we drop the learning rate by 90% at epochs 140. We use random horizontal flipping, random scaling in the range of $[0.6, 1.3]$, and color jittering for data augmentation. We randomly crop large images or pad small images into a fixed size for training. For the ECP, CMP, and eTRIMS dataset, we train on an input resolution of 512×512. Because the images in Graz50 have lower resolutions, which vary between 200 and 500 pixels in height and width, we use 256×256 input resolution for this dataset.

Method	Window	Balcony	Door	Shop	Avg.
	Pixel Accuracy (%)				
Cohen *et al.* [4]	85	91	79	94	87.3
ATLAS [25]	78	87	71	95	82.8
Cohen *et al.* [3]	87	92	79	96	88.5
Rahmani *et al.* [29]	80.4	86.4	79.5	95.2	85.4
DeepFacade-V1 [23]	93.0	95.0	90.9	95.6	93.6
Rahmani *et al.* [30]	78.6	89.2	89.2	**96.3**	88.3
DeepFacade-V2 [22]	**97.6**	**96.2**	92.3	96.0	**95.5**
Ours	94.4	95.9	**95.3**	92.0	94.4
	IoU (%)				
DeepFacade-V2 [22]	80.3	85.2	63.1	80.3	77.2
Ours	**89.8**	**88.0**	**64.3**	**86.1**	**82.1**
Δ	9.5	2.8	1.2	5.8	4.9

Table 2. Quantitative comparison with state-of-the-art facade parsing methods on the ECP dataset.

4.3. Quantitative Evaluation

We quantitatively evaluate our method by comparing it with state-of-the-art facade parsing approaches on the Graz50 and ECP datasets. As stated in the recent work DeepFacade-V2 [22], most of the existing works merely use simple pixel accuracy metric for evaluation. However, high pixel accuracy does not always imply superior performance because of the class imbalance. Following the previous work [22], we mainly use IoU metric for evaluation and also report pixel accuracy results as a reference.

Table 1 shows the performance of our method and other state-of-the-art methods on the Graz50 dataset. As it shows, our method achieves the highest average pixel accuracy. Compared with the state-of-the-art method DeepFacade-V2 [22], our method gives better IoU results by a large margin. In Table 2, we provide the quantitative comparison on the ECP dataset. It shows that our method outperforms the state-of-the-art method in IoU of all classes and provides comparable pixel accuracy results with DeepFacade-V2 [22]. The "Δ" in the last row denotes the performance gain brought by our EACNet compared with DeepFacade-V2, showing the superiority of our approach.

Table 1 and Table 2 show that our method provides much higher IoU on each facade element category, especially those highly aligned and repetitive in structure. In particular, for the *'window'* category which is the most frequent element on facades, compared with the best result of the previous method, our method improves the IoU by about 10% on both Graz50 and ECP datasets. It demonstrates that our EACNet effectively leverages the layout regularity of building facades and exploits long-range dependencies between facade elements.

4.4. Qualitative Evaluation

To better demonstrate the superiority of our facade parsing framework, we show some facade parsing results

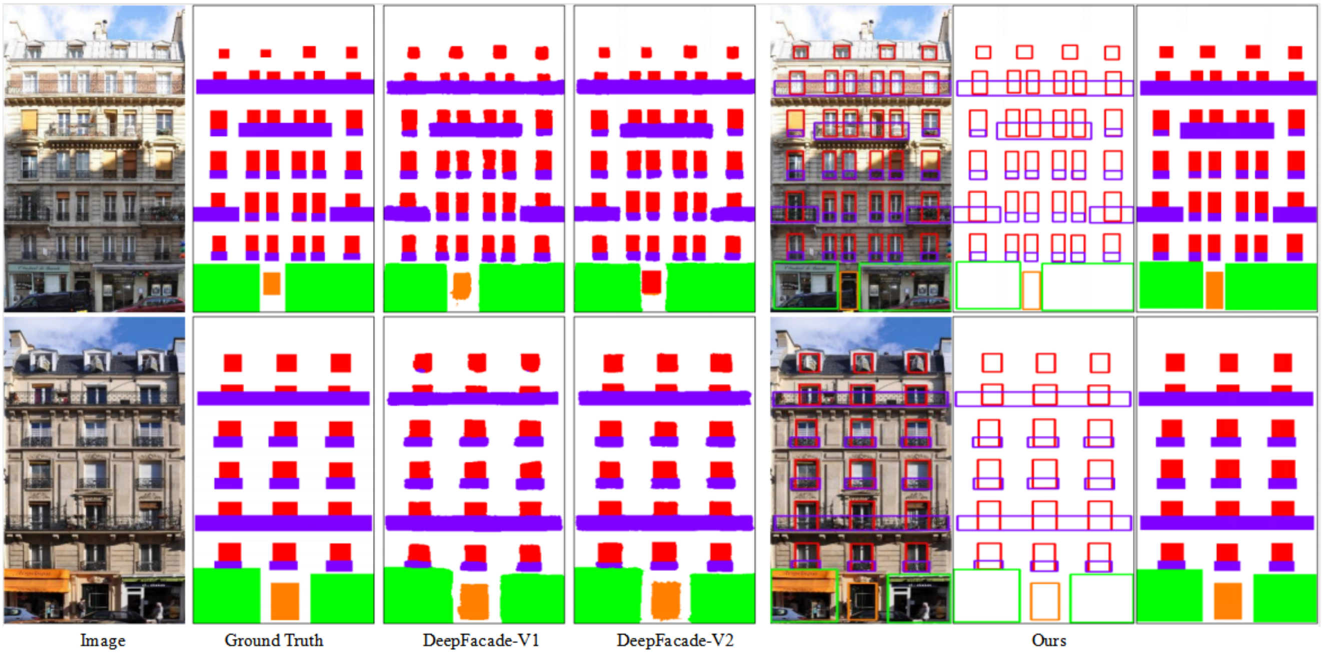

Figure 6. Qualitative comparisons of our method and state-of-the-art facade parsing methods DeepFacade-V1 [23] and DeepFacade-V2 [22] on the ECP dataset. The last three columns are the results parsed by the proposed EACNet, from left to right are the sample images with labeled bounding boxes, the visualization of parameterized rectangular regions, and the rendered semantic maps, respectively.

in Figure 6 of our method and state-of-the-art methods DeepFacade-V1/V2 on the ECP dataset. DeepFacade-V1 tends to produce rough region boundaries for facade elements. DeepFacade-V2 produces more rectangular regions but mistakenly classifies the door as a window in the first row. In contrast, our method produces more regular regions for various facade element categories. Moreover, the parameterized parsing results allow overlapping and nesting, which is more applicable than dense pixel-wise masks to applications such as facade modeling. In particular, though the area where windows and balconies overlap has a complex texture, our parsing framework is able to produce complete regions for *'window'* and *'balcony'* objects.

4.5. Results on Unrectified Facade Images

While the results on the ECP, CMP, and Graz50 datasets well demonstrate the effectiveness of our EACNet, we also show the flexibility of our EACNet on parsing unrectified facade images. On unrectified facade images, elements are not perfectly rectangular, for which our EACNet is not applicable directly. However, there are many well-established rectification approaches for facade images. We take the TILT approach [47] which estimates the homography matrix for image rectification based on low-rank texture features. Given an image region that contains windows, TILT [47] estimates a homography matrix and applies the projection transformation on the entire image to produce a rectified facade image.

Method	Window	Door	**Avg.**
DeepFacade-V2 [22]	71.1	77.9	74.5
Ours-Unrectified	65.2	68.8	67.0
Ours-Rectified	**85.2**	**79.4**	**82.3**

Table 3. Comparison of IoU on the eTRIMS Dataset [22]

We conduct evaluations on the 8-class eTRIMS datasets [22], which contain 60 facade images from different perspectives. We use 48 images for training and 12 images for testing. The eTRIMS dataset consists of 8 classes including *window, wall, door, sky, pavement, vegetation, car*, and *road*. Putting aside the categories not belonging to facade elements, we choose the *'window'* and *'door'* categories for evaluation. Since only semantic masks on the unrectified views are provided in the eTRIMS dataset, we manually label the bounding boxes for evaluation.

We train two models of our EACNet on the rectified images and unrectified images respectively and compare the results with DeepFacade-V2 [22] in Table 3. 'Ours-Unrectified' model is directly trained with the 2D bounding boxes of facade elements on unrectified perspectives. It is reasonable that this model can not achieve a higher IoU with the ground-truth segmentation masks that are not rectangular. 'Ours-Rectified' is the model trained with 2D bounding boxes that are well-fitted to the element regions on the rec-

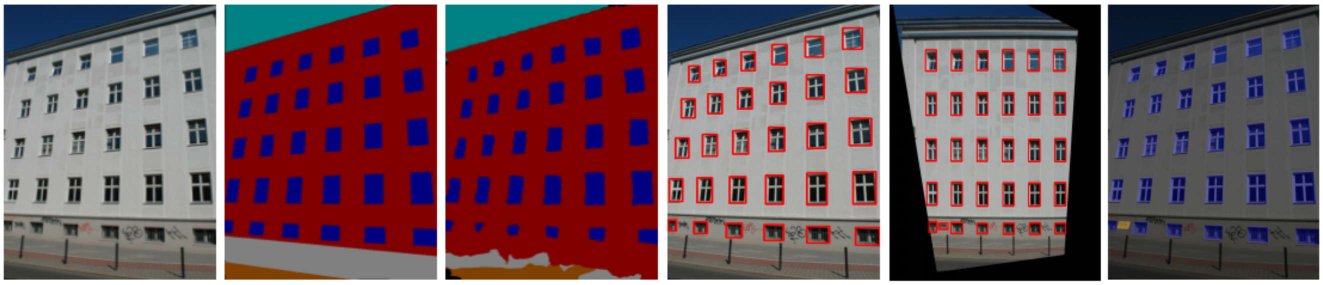

Figure 7. Qualitative comparison for unrectified facade images. From left to right are the input image, ground truth mask, semantic segmentation result of DeepFacade-V1 [23], our detection results without rectification, our detection result on the rectified image, and the semantic mask obtained by transforming the detection results on the rectified image to the original view.

tified images. We test our model on the testing set of the rectified images and obtain the bounding boxes detected on the rectified images. Then we apply the inverse projection transformation on the bounding boxes to produce the semantic mask on the unrectified images. We calculate the IoU with the ground truth. As Table 3 shows, our method outperforms the segmentation-based method DeepFacade-V2 [22] by a large margin with image rectification.

Figure 7 shows an example of facade parsing for unrectified images using different methods. Though the segmentation-based method is flexible to represent non-rectangular regions under perspective projection, it fails to generate accurate and regular region boundaries for facade elements. Due to the restriction of rectangular shapes of the detection framework, directly applying our EACNet on the unrectified images successfully detects all elements but fails to generate well-fitted region boundaries. In comparison, with a well-established rectification step, our EACNet can produce accurate and structured region boundaries for facade elements.

4.6. Ablation Study

4.6.1 Different Attention Mechanisms

As described in Sec. 3.2, our EACM is designed to collect row-column spatial contextual information and leverage the arrangement regularity of the facade elements. The recurrent criss-cross attention (RCCA) module of CCNet [15] collects spatial context in criss-cross paths, which is similar but different from our EACM. We compare our EACM with RCCA on Graz50, ECP, and CMP datasets. We replace EACM with RCCA in our framework and use the same training settings for comparison. We test two models, $\text{RCCA}^{\mathbf{1}}$ and $\text{RCCA}^{\mathbf{2}}$ which employ one-loop and two-loops of RCCAs, respectively. As Table 4 shows, our EACM brings performance gain on all three datasets, while the RCCA only achieves slight improvement on the CMP dataset. RCCA even performs worse than the baseline network on Graz50 and ECP. One reason is that RCCA collects contextual information from all the pixels on the criss-cross paths and applies softmax on them, subsequently cannot efficiently utilize the element-arrangement regularity on each direction separately. As a result, for the Graz50 and ECP datasets that contain facades with neatly arranged facade elements, RCCA does not work well. For facades with complex layouts and more categories in CMP, the dense full-image contextual information harvested by RCCA can be helpful. In contrast, our EACM effectively exploits the layout regularity in horizontal and vertical directions separately and outperforms RCCA on various scenarios.

Dataset	Method	AP	AP^{50}	AP^{75}
Graz50	Baseline	65.8	94.1	**85.2**
	+RCCA^{1}	62.3	94.1	79.7
	+RCCA^{2}	63.8	94.7	83.1
	+EACM	**68.2**	**96.8**	84.2
ECP	Baseline	79.3	99.4	93.6
	+RCCA^{1}	78.1	99.4	93.0
	+RCCA^{2}	78.4	99.4	94.1
	+EACM	**80.1**	**99.4**	**95.2**
CMP	Baseline	39.7	67.9	41.0
	+RCCA^{1}	39.7	68.4	40.7
	+RCCA^{2}	39.8	68.3	41.2
	+EACM	**40.2**	**68.4**	**42.3**

Table 4. The performance of our method on Graz50, ECP, and CMP datasets. We show comparison of the proposed EACM and the most related method, recurrent criss-cross attention module [15]. $\text{RCCA}^{\mathbf{1}}$ and $\text{RCCA}^{\mathbf{2}}$ correspond to RCCAs with the number of recurrent 1 and 2 separately.

4.6.2 Effect of EACM

In Table 5, we show the quantitative performance of our method with different configurations on the CMP dataset. '+EACM' means adding an EACM between the hourglass backbone and the detector head. 'Flip Test' means combining horizontally flipped images during inference, which is

+EACM	Flip Test	AP	AP^{50}	AP^{75}	Sill	Balcony	Door	Molding	Window	Cornice
		38.6	67.1	38.9	38.6	33.9	33.8	25.3	59.2	40.8
	✓	39.7	67.9	41.0	40.3	34.9	34.5	25.5	60.8	**42.2**
✓		39.2	67.5	40.1	39.9	35.0	34.4	24.9	60.4	40.6
	✓	**40.2**	**68.4**	**42.3**	**40.9**	**35.8**	**34.6**	**26.0**	**62.0**	41.6

Table 5. The effect of our element-arrangement context module. The last six columns are per-class AP results of facade element categories of the CMP dataset. We show results with/without flip test-time augmentation.

widely used in recent detection networks [19, 49]. The results show that our EACM consistently improves the three AP metrics and per-class AP of important facade element classes. In particular, our EACM significantly improves the parsing accuracy of the *window* category (from 59.2 to 60.4 without flip-test and from 60.8 to 62.0 with flip-test). It is mainly because that windows show strong regularity and repetitiveness and our EACM effectively exploits the arrangement regularity and appearance similarity among window elements. Doors do not strictly follow the arrangement regularity with other elements. Nevertheless, the slight improvement for the *'door'* category also indicates that our EACM is also helpful for shape regularity since it collects local spatial context for each position on a door.

To validate the effectiveness of the proposed EACM on leveraging the layout regularity, we visualize the attention maps in Figure 8. We can see that EACM focuses on element regions aligned in the same row or column, which proves that our method effectively exploits the spatial arrangement regularity and appearance similarity. In addition, we further investigate the effect of EACM by exploring two different strategies for fusing the contextual features produced by two context branches. Besides concatenating features S_{row} and S_{col} to produce S, another possible fusion strategy is element-wise addition. The precision-recall curves under different IoU thresholds are shown in Figure 9. The results indicate that the two configurations of EACM both make a performance improvement and concatenation fusion achieves the best performance.

4.6.3 Different Backbones

To further demonstrate the effectiveness of our EACM on various networks, we combine our EACM with different backbone networks, including ResNet-101 [13], DLA-34 [46], and Hourglass [28]. Table 6 shows our quantitative results on the CMP dataset. We report the average precision over all the IoU thresholds (AP), AP at IoU threshold 0.5 (AP^{50}), and 0.75 (AP^{75}). One can see that adding our EACM brings performance gains on all the three backbone networks, demonstrating that our EACM facilitates the facade parsing task by exploiting the spatial arrangement regularity and appearance similarity of facade elements.

5. Conclusion

In this paper, we presented an Element-Arrangement Context Network (EACNet) for facade parsing. Our EACNet parses a facade image into axis-aligned element regions as parameterized bounding boxes. The proposed Element-Arrangement Context Module (EACM) collects spatial column-context and row-context simultaneously, effectively leveraging the spatial arrangement regularity and appearance similarity. Experiments on four public datasets show that our facade parsing framework outperforms the existing facade parsing methods. The significant performance improvements demonstrate that the proposed EACM guides the network focus on facade elements aligned in the same row and column to utilize layout regularity.

Acknowledgement

This work was supported by Tencent Corporation.

References

[1] F. Bao, M. Schwarz, and P. Wonka. Procedural facade variations from a single layout. *ACM Transactions on Graphics*, 32(1), 2013. 2

[2] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2018. 4

[3] A. Cohen, M. R. Oswald, Y. Liu, and M. Pollefeys. Symmetry-aware facade parsing with occlusions. In *International Conference on 3D Vision*, pages 393–401, 2017. 2, 6

[4] A. Cohen, A. G. Schwing, and M. Pollefeys. Efficient structured parsing of facades using dynamic programming. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3206–3213, 2014. 1, 2, 6

[5] M. Dang, D. Ceylan, B. Neubert, and M. Pauly. SAFE: structure-aware facade editing. *Computer Graphics Forum*, 33(2):83–93, 2014. 2

[6] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 3

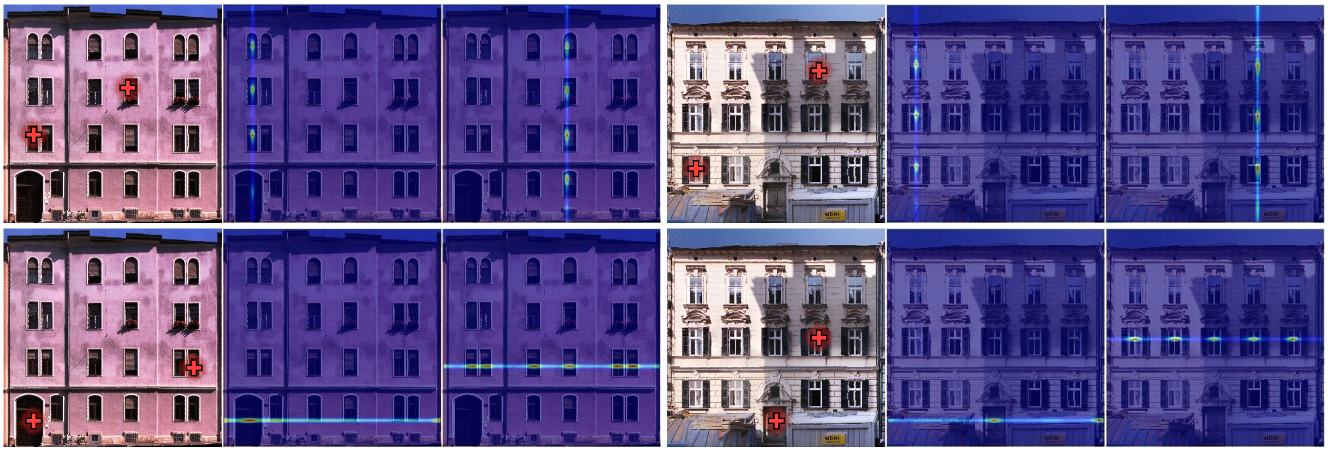

Figure 8. Visualization of attention maps A_{col} and A_{row} in our EACM. The query points are marked in red.

Backbone	+EACM	AP		AP^{50}		AP^{75}	
		w/o Flip	Flip	w/o Flip	Flip	w/o Flip	Flip
Hourglass		38.6	39.7	67.1	67.9	38.9	41.0
	✓	**39.2**	**40.2**	**67.5**	**68.4**	**40.1**	**42.3**
DLA-34		32.4	33.7	62.8	63.7	29.8	31.2
	✓	33.5	34.6	63.8	65.1	31.7	33.0
ResNet-101		29.9	31.0	60.8	61.9	26.7	27.7
	✓	30.9	31.9	62.4	63.8	26.8	28.2

Table 6. Comparison of different backbone networks on CMP dataset. '+EACM' means adding our EACM module between the backbone network and the detector head. 'Flip' means using test-time flip augmentation. 'w/o Flip' means no flip test-time augmentation.

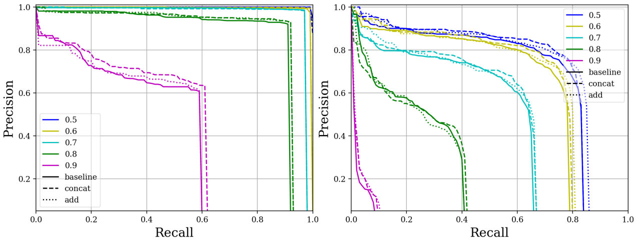

Figure 9. The effect of using different fusion strategies in EACM. We show precision-recall curves under different IoU thresholds. Left: results on ECP. Right: results on CMP. The solid lines correspond to the baseline. Dashed lines and dotted lines are results of concatenation and addition fusion respectively.

[7] J. Femiani, W. Reyaz Para, N. Mitra, and P. Wonka. Facade segmentation in the wild. In *arXiv preprint arXiv:1805.08634*, 2018. 2

[8] R. Gaddle, V. Jampani, R. Marlet, and P. V. Gehler. Efficient 2D and 3D facade segmentation using auto-context. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(5):1273–1280, 2018. 2

[9] R. Girshick. Fast R-CNN. In *IEEE International Conference on Computer Vision*, pages 1440–1448, 2015. 3

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014. 3

[11] F. Han and S.-C. Zhu. Bottom-up/top-down image parsing by attribute graph grammar. In *IEEE International Conference on Computer Vision*, volume 2, pages 1778–1785, 2005. 2

[12] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *IEEE International Conference on Computer Vision*, pages 2980–2988, 2017. 2

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 9

[14] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 3

[15] Z. Huang, X. Wang, Y. Wei, L. Huang, H. Shi, W. Liu, and T. S. Huang. Ccnet: Criss-cross attention for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 3, 8

[16] M. Ilčík, P. Musialski, T. Auzinger, and M. Wimmer. Layer-based procedural design of façades. *Computer Graphics Forum*, 34(2):205–216, 2015. 2

[17] M. Koziński, R. Gadde, S. Zagoruyko, G. Obozinski, and R. Marlet. A mrf shape prior for facade parsing with occlu-

sions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2820–2828, 2015. 2, 6

[18] M. Koziński, G. Obozinski, and R. Marlet. Beyond procedural facade parsing: Bidirectional alignment via linear programming. In *Asian Conference on Computer Vision*, pages 79–94, 2015. 6

[19] H. Law and J. Deng. Cornernet: Detecting objects as paired keypoints. In *European Conference on Computer Vision*, pages 734–750, 2018. 3, 9

[20] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *IEEE International Conference on Computer Vision*, pages 2999–3007, 2017. 5

[21] 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*, pages 740–755, 2014. 6

[22] H. Liu, Y. Xu, J. Zhang, J. Zhu, Y. Li, and S. C. H. Hoi. DeepFacade: A deep learning approach to facade parsing with symmetric loss. *IEEE Transactions on Multimedia*, 22(12):3153–3165, 2020. 1, 2, 4, 5, 6, 7, 8

[23] H. Liu, J. Zhang, and S. C. H. Hoi. DeepFacade: A deep learning approach to facade parsing. In *International Joint Conference on Artificial Intelligence*, pages 2301–2307, 2017. 1, 2, 4, 6, 7, 8

[24] A. Martinović, M. Mathias, J. Weissenberg, and L. Van Gool. A three-layered approach to facade parsing. In *European Conference on Computer Vision*, pages 416–429, 2012. 1

[25] M. Mathias, A. Martinović, and L. Van Gool. ATLAS: A three-layered approach to facade parsing. *International Journal of Computer Vision*, 118(1):22–48, 2016. 1, 2, 6

[26] P. Müller, G. Zeng, P. Wonka, and L. Van Gool. Image-based procedural modeling of facades. *ACM Transactions on Graphics*, 26(3):85–es, 2007. 1

[27] A. Newell, Z. Huang, and J. Deng. Associative embedding: End-to-end learning for joint detection and grouping. In *Advances in Neural Information Processing Systems*, pages 2277–2287, 2017. 3

[28] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*, pages 483–499, 2016. 3, 6, 9

[29] K. Rahmani, H. Huang, and H. Mayer. Facade segmentation with a structured random forest. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-1/W1:175–181, 2017. 2, 6

[30] K. Rahmani, H. Huang, and H. Mayer. High quality facade segmentation base on structured random forest, region proposal network and rectangular fitting. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-2:223–230, 2018. 6

[31] H. Riemenschneider, U. Krispel, W. Thaller, M. Donoser, S. Havemann, D. Fellner, and H. Bischof. Irregular lattices for complex shape grammar facade parsing. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1640–1647, 2012. 2, 5

[32] A. Roy, M. Saffar, A. Vaswani, and D. Grangier. Efficient content-based sparse attention with routing transformers. *Transactions of the Association for Computational Linguistics*, 9:53–68, 2021. 3

[33] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich. Superglue: Learning feature matching with graph neural networks. In *IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020. 3

[34] M. Schmitz and H. Mayer. A convolutional network for semantic facade segmentation and interpolation. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pages 709–715, 2016. 1, 2

[35] C.-H. Shen, S.-S. Huang, H. Fu, and S.-M. Hu. Adaptive partitioning of urban facades. *ACM Transactions on Graphics*, 30(6):1–10, 2011. 1

[36] J. O. Talton, Y. Lou, S. Lesser, J. Duke, R. Měch, and V. Koltun. Metropolis procedural modeling. *ACM Transactions on Graphics*, 30(2), 2011. 2

[37] O. Teboul, I. Kokkinos, L. Simon, P. Koutsourakis, and N. Paragios. Shape grammar parsing via reinforcement learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2273–2280, 2011. 1, 2

[38] O. Teboul, L. Simon, P. Koutsourakis, and N. Paragios. Segmentation of building facades using procedural shape priors. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3105–3112, 2010. 1, 2, 5

[39] R. Tyleček and R. Šára. Spatial pattern templates for recognition of objects with regular structure. In *German Conference on Pattern Recognition*, pages 364–374, 2013. 2, 5, 6

[40] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 3

[41] S. Wang, B. Z. Li, M. Khabsa, H. Fang, and H. Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020. 3

[42] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018. 3

[43] C. Yang, T. Han, L. Quan, and C.-L. Tai. Parsing façade with rank-one approximation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 1, 2

[44] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489, 2016. 3

[45] Y.-T. Yeh, K. Breeden, L. Yang, M. Fisher, and P. Hanrahan. Synthesis of tiled patterns using factor graphs. *ACM Transactions on Graphics*, 32(1), 2013. 2

[46] F. Yu, D. Wang, E. Shelhamer, and T. Darrell. Deep layer aggregation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2403–2412. 9

[47] Z. Zhang, A. Ganesh, X. Liang, and Y. Ma. Tilt: Transform invariant low-rank textures. *International journal of computer vision*, 99(1):1–24, 2012. 7

[48] H. Zhao, Y. Zhang, S. Liu, J. Shi, C. C. Loy, D. Lin, and J. Jia. Psanet: Point-wise spatial attention network for scene parsing. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 267–283, 2018. 3

[49] X. Zhou, D. Wang, and P. Krähenbühl. Objects as points. In *arXiv preprint arXiv:1904.07850*, 2019. 3, 5, 9