Deep Window Detection in Street Scenes

Wenguang Ma and Wei Ma*
Faculty of Information Technology, Beijing University of Technology
Beijing, China
[e-mail: mawenguang@emails.bjut.edu.cn, mawei@bjut.edu.cn]
*Corresponding author: Wei Ma

Received Septebmer 15, 2019; revised December 5, 2019; accepted January 27, 2019;
published February 29, 2020

Abstract

Windows are key components of building facades. Detecting windows, crucial to 3D semantic reconstruction and scene parsing, is a challenging task in computer vision. Early methods try to solve window detection by using hand-crafted features and traditional classifiers. However, these methods are unable to handle the diversity of window instances in real scenes and suffer from heavy computational costs. Recently, convolutional neural networks based object detection algorithms attract much attention due to their good performances. Unfortunately, directly training them for challenging window detection cannot achieve satisfying results. In this paper, we propose an approach for window detection. It involves an improved Faster R-CNN architecture for window detection, featuring in a window region proposal network, an RoI feature fusion and a context enhancement module. Besides, a post optimization process is designed by the regular distribution of windows to refine detection results obtained by the improved deep architecture. Furthermore, we present a newly collected dataset which is the largest one for window detection in real street scenes to date. Experimental results on both existing datasets and the new dataset show that the proposed method has outstanding performance.

Keywords: Window dataset, window detection, regular distribution, context enhancement, convolutional neural network

A preliminary version of this paper was presented at APIC-IST 2019, and was selected as an outstanding paper. This research was supported by National Natural Science Foundation of China (61771026), and the Open Project Program of the National Laboratory of Pattern Recognition (NLPR).

http://doi.org/10.3837/tiis.2020.02.022 **ISSN : 1976-7277**

1. Introduction

Windows are important parts of building facades. The purpose of window detection is to obtain the locations of windows in input images. It is a fundamental task in computer vision and can help 3D reconstruction and visual SLAM in street scenes [1]. It also has many other applications, such as city modeling and autonomous city navigation.

Accurate detection of windows, however, is challenging due to the complexity in real scenes. Specifically speaking, windows in different styles of facades have various appearances. The opening or closing states of windows are uncertain. Decorations looking like windows are inevitable. Glass reflection causes large variations of window appearances. Occlusions, such as trees and vehicles, often appear in front of buildings.

In the past few years, many methods [2, 3, 4] have been proposed to detect windows. Most of them are based on hand-crafted features and traditional classifiers. In these methods, a sliding window is often used to extract multi-scale proposals, each indicating a possible window. Hand-crafted features, e.g. HOG [5], SIFT [6] and Haar wavelet [7] are extracted from each proposal region. Classifiers, such as AdaBoost [7] or SVM [8], are trained to determine the labels of proposals. These methods have many limitations. First, these hand-crafted features are inadequate to represent complex windows. Second, the sliding window always generates many redundant proposals, which substantially slows the detection process.

Recently, CNN-based object detection technologies [9-20] have shown its amazing power in various fields, such as vehicle detection [9] for transportation surveillance, face detection [10] for real-time video analysis and CT lesion detection [11] for AI medicine. Generally, these detection methods adopt a convolutional neural network to extract features of input images, which are then fed into two branches, for object classification and bounding box localization, respectively. Currently, there are many popular object detection algorithms. For example, R-CNN [12], Fast R-CNN [13] and Faster R-CNN [14] are two-stage detectors favoring high accuracy. YOLO [15], SSD [16] and RetinaNet [17] are one-stage detectors favoring high efficiency. However, direct using these algorithms for window detection is unable to obtain satisfactory results due to the various appearances of windows and the complexity of real scenes.

In this paper, we introduce an accurate and efficient window detection architecture which is inspired by the two-stage detector Faster R-CNN [14]. The proposed architecture mainly includes three novel modules: Window RPN, RoI feature fusion, and context enhancement module. In Window RPN, we design three extra anchors according to the size distribution of windows. With the original nine anchors used in Faster R-CNN, Window RPN, containing twelve anchors, achieves better matching between anchors and window ground truth boxes. In order to handle some occlusions and small windows, we present an RoI feature fusion module to take advantage of both the details in the low-level layers and context in the high-level layers. Due to the layout of windows takes a grid structure, the square receptive field achieved by feature extraction module (e.g. VGGNet [21] or ResNet [22]) may affect the detection of windows. We propose a novel Context Enhancement Module (CEM) that provides diverse receptive fields to tackle this problem. Finally, object classification branch and bounding box localization branch are adopted to detect windows. Furthermore, windows with confidence scores lower than a pre-defined threshold will be filtered out. Based on the regularity of windows in facades, such as the similarity and repeatability, we present a post optimization method to discover probably missed windows.

To our best knowledge, there is no specialized dataset for window detection in real street scenes to date, and our work fills the gap. We provide a new dataset, named Street Scene Window Detection (SSWD), which includes thousands of images containing windows. SSWD is carefully annotated with exhaustive bounding boxes. Some examples are shown in **Fig. 1**. The main contributions can be summarized as:

- A window detection dataset of street scenes is built and published at: https://github.com/wohaiyo/StreetSceneWindowDetectionDataset. The dataset, together with a small-scale dataset composed of around 100 pure facade images, will be used to train and evaluate the proposed method which has specialties as follows.
- We introduce an improved window detection network that features Window RPN and RoI feature fusion.
- We propose a novel context enhancement module to diversify the receptive fields of features in our detection network, which help achieve better results.
- Base on the regularity distribution of windows on the facade, we present an effective post optimization method to relocate missed windows.

The paper is organized as follows: we first review the previous works about window detection in Section 2. Then, we explain the details of the proposed SSWD dataset in Section 3. The proposed method for window detection is presented in Section 4. Experimental settings and results are provided in Section 5. Finally, Section 6 concludes the paper.

2. Related Work

In this section, we first review some early traditional methods on detecting windows. Then, some state-of-the-art object detection algorithms based on deep CNNs are introduced. We further analyze some existing datasets used for window detection.

2.1 Traditional window detection

Research on window detection has been active for a long time. Most of the early works [2, 3, 4] use hand-crafted features of windows to train a classifier and employ the sliding window to find out all possible positions. For example, [2] proposed a window detection system. During training, it extracted multi-scale Haar wavelet representation from marked regions in training images and learned an Adaboost driven cascaded decision tree. During inference, a sliding window was moved over a test image with pyramid scales. Similarly, [3] proposed a pipeline to detect windows of rectified images based on Haar-like features. However, the Haar-like features used in [7] were not robust since it cannot handle window detection in complex scenes. Moreover, the sliding window operation was time-consuming. On the other hand, another method [4] presented an idea for window detection that does not require a learning stage. This method achieved window detection by extended gradient projection with a facade color descriptor based on k-means clustering in CIE-Lab color space. It is difficult to use the method without learning for detecting windows in real street scenes. In particular, the texture of walls and some decorations in facades are generally complex, which makes it hard to use only gradient projection to locate windows.

2.2 Object detection in deep CNNs

With the quick improvement of deep convolutional neural networks, object detection is dominated by CNN-based detectors, which could be roughly divided into two categories:

two-stage approaches and one-stage approaches. The two-stage detection approaches, like R-CNN [12], Fast R-CNN [13] and Faster R-CNN [14], mainly consist of two stages. The first stage proposes a set of candidate regions, and the second one determines the accurate bounding boxes for the proposed regions and the corresponding class labels. Notably, Ren et al. [14] proposed region proposal network (RPN), a fully convolutional network that replaces traditional selective search strategy. Thus, the two-stage detectors could be trained end-to-end and generate high accuracy detection results. It ran slow (about 5 fps) due to the two-stage computational costs. The one-stage object detection approaches (e.g., YOLO [15], SSD [16], RetinaNet [17], CornerNet [18]) address the low-efficiency problem by using the feed-forward convolutional network to directly predict object locations and labels. SSD [16] spread out anchors with different scales to several convolutional layers and enforced each layer to focus on predicting object at a certain scale. Therefore, SSD achieved high accuracy in real-time. However, there exist serious class imbalance problems in one-stage object detector. [17] proposed focal loss which makes the network focus on the training of hard examples and prevents easy negative examples. To sum up, many CNN-based object detection algorithms have been proposed and verified in various tasks: face detection, pedestrian detection, and vehicle detection, etc. However, directly using these algorithms to detect windows cannot obtain satisfactory results. Our method combines the strength of the CNN detector and regular distribution property of windows to improve the accuracy of window detection.

2.3 Dataset about window detection

As for current data resources, to our best acknowledge, there is no dataset especially collected for window detection task to date. Although there exist a few datasets which contain window label, such as COCO [23] and ADE20K [24]. It is hard to use these datasets to detect windows in real scenes because of their rare window instances. On the other hand, facade datasets, such as CMP [25] and ECP [26], contain more windows in these datasets. However, all of the images in these datasets are rectified and viewed in a front-parallel direction. Using these datasets are also unable to achieve promising results. Here, we propose a new dataset, which is the first one for window detection in street scenes.

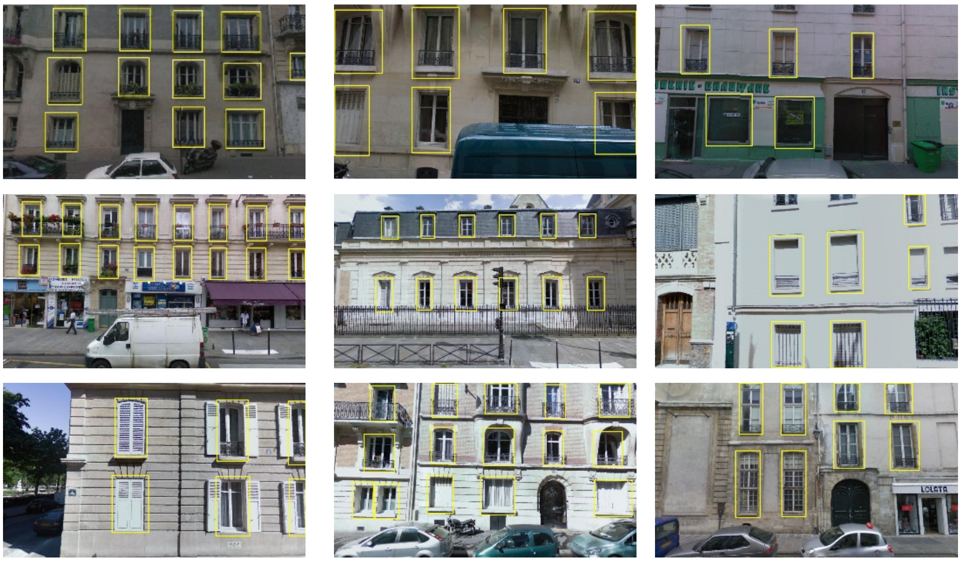


Fig. 1. Examples of our “Street Scene Window Detection (SSWD)” dataset set. Yellow bounding boxes are the annotated ground truth.

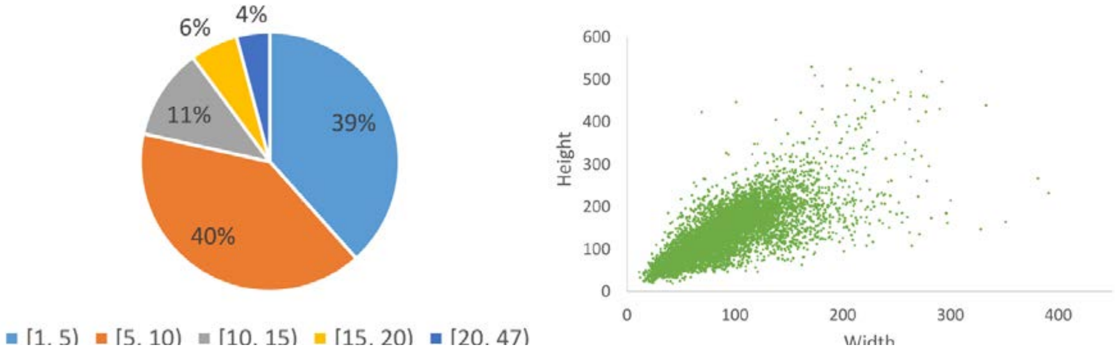
3. Street Scene Window Detection dataset

3.1 Image Annotation

The original street scene images of the SSWD dataset are selected from the Paris Street-View dataset [27] in which most of the images contain building elements, e.g. windows, balconies, etc. We collect more than one thousand images containing windows. All of the window instances are annotated by expert annotators and saved in a way similar to Pascal VOC [28] object detection dataset. We randomly divide Street Scene Window Detection dataset into a training set (1000 images), validation set (200 images), and test set (100 images).

3.2 Dataset Statistics

We discuss the SSWD dataset with more statistical details. First is the number of windows in each image. Our SSWD dataset has an average of 7.25 window instance per image. The interval statistics on the number of windows per image are shown in **Fig. 2 (a)**. It can be seen that the SSWD dataset not only has a wide range of window numbers but also has multiple instances in most of the images. The data distribution of window size is shown in **Fig. 2 (b)**, from which we can see that the width and height of the window instances have large ranges as real cases.



(a) Statistics on the number of windows per image

(b) The data distribution of window size

Fig. 2. Statistics of the SSWD dataset.

4. Window detection Architecture

In this section, we present our window detection architecture which mainly involves a window detection network (WD-Net) and a post optimization method. The overview of our window detection framework is shown in **Fig. 3**. Given a facade image, a backbone VGGNet [21] extracts features that are further enhanced by the Context Enhancement Module (CEM). The enhanced features are fed into Window RPN and RoI Feature Fusion (RFF). Specifically, for each proposal region from Window RPN, feature vectors of fixed length are extracted from VGGNet. RoI feature fusion fuses these multi-scale features adaptively. Each feature vector fused by RoI feature fusion is then fed into a sequence of fully connected (FC) layers that perform classification (Cls) and regression (Reg) for the corresponding proposal region. With the detection results provided by WD-Net, we further propose a post optimization method that includes Clustering and Voting to detect the missed windows.

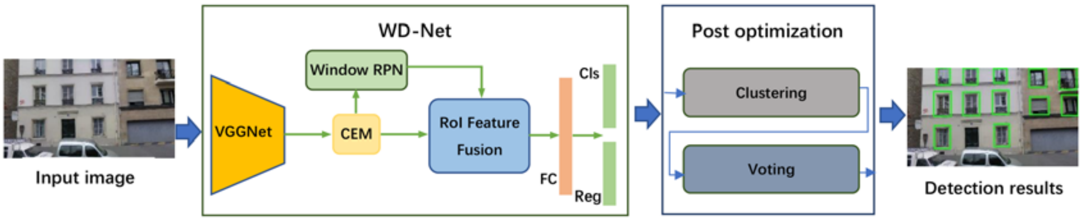


Fig. 3. Overall framework of the proposed window detection approach.

4.1 Window detection network

Our window detection network, called WD-Net, is inspired by the two-stage object detector Faster R-CNN [14]. Different from Faster R-CNN, our WD-Net has two novel architectural changes. The first one contains the window region proposal network (Window RPN) and RoI Feature Fusion (RFF). Window RPN is a specially designed module for detecting windows according to the actual sizes of window instances. RFF combines the details from the low-level layer and context from the high-level layer. The second one is the Context Enhancement Module (CEM). Multi-scale features generated from VGGNet are enhanced by the proposed CEM which diversifies the receptive fields. The details of window detection networks are shown in Fig. 4.

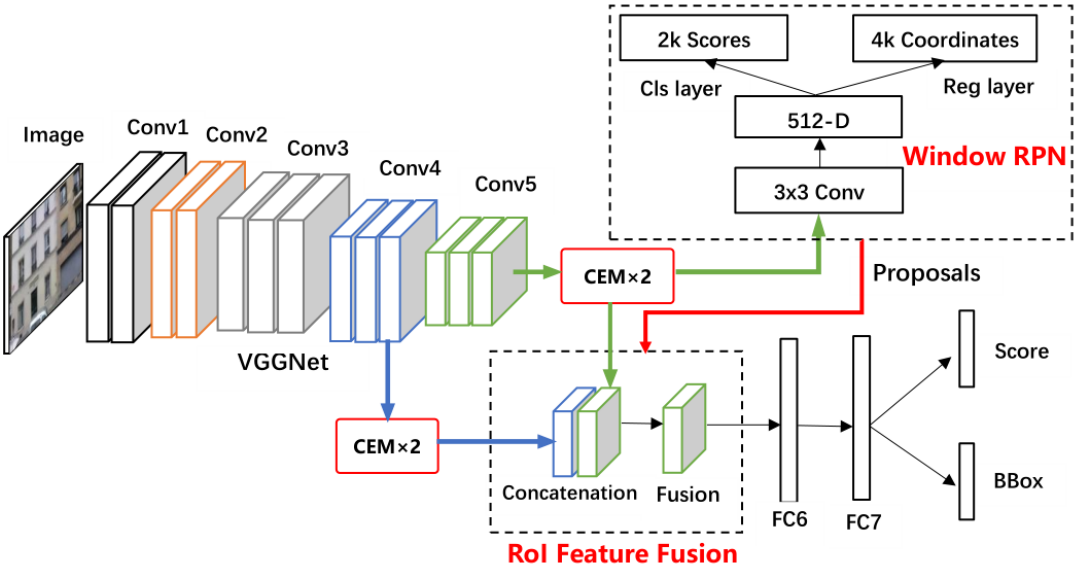


Fig. 4. The overall architecture of our window detection network. VGGNet is the backbone used to extract multi-scale features. The features are further enhanced by CEM. Window RPN focuses on proposal region generation through a classification and regression layer. RoI feature fusion first extracts fixed-length feature vectors from multi-scale features according to the proposals and then combines multi-scale features for fusion by concatenation. Each feature is fed into a sequence of fully connected layers that finally branches into predicting a score and offsets of a bounding box, respectively.

Window RPN

The anchor configuration is crucial for object detector. For two stage-detector, such as Faster R-CNN [14], RPN uses anchors to regress locations of foreground objects, followed by another regression branch to refine the proposal bounding boxes. One-stage detectors, such as RetinaNet [17], uses anchors to regress the bounding boxes of objects directly. For window detection, we design a special anchor configuration for the window detector. The RPN is a proposal region generator in Faster R-CNN, which is class-agnostic for all of the objects in images. For window detection as ours, the original RPN cannot provide satisfactory results due to the challenges in window detection. Therefore, a special window region proposal network module is proposed.

Following Faster R-CNN, our Window RPN is built on the top layer of the feature map (conv5_3 in VGGNet). It is followed by an intermediate 3×3 convolutional layer and two siblings 1×1 convolutional layer for bounding box regression and classification, respectively. In particular, the stride of the output feature is 16. Faster R-CNN adopts 9 anchor boxes with 3 aspect ratios of 1:1, 1:2 and 2:1, with box areas of 128^2, 256^2, 512^2, the area of each anchor is shown in **Table 1**. In window detection, we obtain the size distribution of width and height from **Fig. 2 (b)**. We sample three proposal sizes ([40, 60], [70, 110], [110, 180]) which occur most frequently in SSWD dataset. By combining the original anchors with the three new anchors, our new anchor strategy is formed. Our new anchors are listed in **Table 1** and the visual differences are shown in **Fig. 5**.

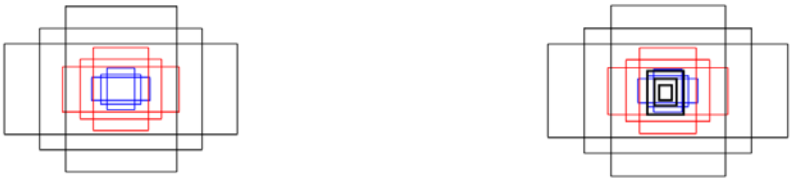


Fig. 5. Comparison of two anchor strategies. The left is used in Faster R-CNN, the right is used in our WD-Net, the black rectangles in the right image are the special anchor designed for window detection.

Table 1. Size of anchors in Faster R-CNN and our new anchors

Method	Ratio=0.5	Ratio=1.0	Ratio=2.0
Faster R-CNN	184×96	128×128	88×176
	368×192	256×256	176×352
	736×384	512×512	352×704
Our new anchors	40×60	70×110	110×180

RoI Feature Fusion

Windows in the wild always have various scales and may be seriously occluded by trees or cars. Faster R-CNN adopts the top layer feature map (conv5_3) which is not robust enough to detect windows. We address this problem by RoI Feature Fusion (RFF). RFF tackles scale variations of windows by incorporating low-level features and occlusions by enhancing the context of high-level features.

Through a backbone network, the feature map of each layer is extracted. The low-level feature map often has detail information, such as edges and textures, which lack a semantic context. On the contrary, the high-level feature map represents rich semantics but the detail information is rare. In order to aggregate the information of different level layers, RFF adopts conv4_3 and conv5_3 from the backbone instead of using conv5_3 alone. Conv4_3 provides more detail information for detecting some tiny windows and Conv5_3 provides abundant semantic context which is useful to recognize windows. Some regions of interest (RoI) are extracted from conv4_3 and conv5_3 according to the proposals from Window RPN. We resize each RoI with a fixed spatial size of 14×14 by RoIAlign [29] operation, followed by a max-pooling layer whose kernel size is 2 and stride is 2. After that, two feature maps with 7×7 resolution are concatenated. The feature dimension is reduced by a 1×1 convolution. Then the new feature is fed into bounding box regression and classification module to determine the accurate coordinates and class label of the proposal region.

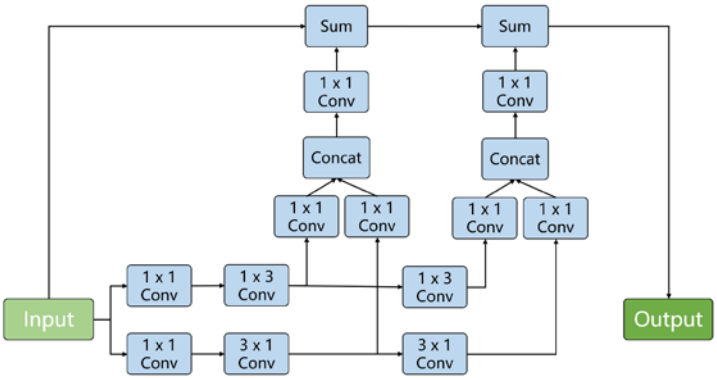


Fig. 6. The structure of our Context Enhancement Module (CEM) adopts 1D convolution kernels to construct the rectangular receptive fields.

Context Enhancement Module

As we all know, the shapes of windows are almost rectangular and the layouts of windows are latticed. Nevertheless, the receptive fields obtained by the backbone are usually square which may hurt the detection results of windows. In order to diversify the receptive fields of features, we design a novel Context Enhancement Module (CEM) based on 1D convolutions. The details of the proposed CEM are shown in **Fig. 6**. We first reduce the channel number to one half of the previous layer by a 1×1 convolution layer. Then, we use 1×k and k×1 (k=3) to provide a rectangular receptive field. Through another two 1×1 convolution layer, the feature maps from two branches are concatenated together. Meanwhile, the intermediate features are further processed by another 1×k and k×1 (k=3) convolution to enhance the diversity of receptive fields. Finally, with the same concatenation, features are fused by element-addition. In particular, as our WD-Net, we use a cascaded CEM to enlarge the receptive field of conv4_3 and conv5_3 (see **Fig. 4**). Through the 1D convolutional kernels, some extreme window instances will be detected.

Loss Function

For training Window RPN, each anchor is assigned with a binary class label. An anchor is assigned with a positive label if the anchor has the highest Intersection-over-Union (IoU) overlap with a ground truth box or its IoU overlap with any ground truth box is higher than 0.7. An anchor is assigned with a negative label if its IoU overlap with any ground truth box is lower than 0.3. We use the multi-task loss like the one in [14] to train our WD-Net, which is defined as follows:

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i,p_i^*) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i,t_i^*) \tag{1}$$

where i is the index of an anchor. L_{cls} is the soft-max loss function for classifying the windows and backgrounds. p_i and p_i^* are the predicted probability of anchor i and the ground truth label, respectively. L_{reg} is the smooth L1 loss for regressing bounding boxes of windows. p_i^* is 1 only for the positive anchor. Otherwise, it is 0. The two term L_{cls} and L_{reg} are normalized by N_{cls} and N_{reg}, and controlled by the hyper-parameter λ. We set $\lambda = 1$ in experiments.

The smooth L1 loss function used in regression of window bounding boxes is defined as follow:

$$L_{reg}(t,t^*) = \sum_{i\in\{x,y,w,h\}} smooth_{L_1}(t_i - t_i^*) \tag{2}$$

$$smooth_{L_1}(x) = \begin{cases} 0.5x^2, & if|x| < 1 \\ |x| - 0.5, & otherwise \end{cases} \tag{3}$$

where, t and t^* are predicted and ground truth bounding boxes, respectively. i is the index of (x,y,w,h) which denotes the center coordinates, width and height of a bounding box. The smooth L1 loss used in object detection is more robust and less sensitive to outliers.

4.2 Regular distribution based post optimization method (Post Processing)

The window detection network outputs many candidate bounding boxes and the corresponding confidence scores. Some candidate boxes might be filtered out for the reason that its confidence score is lower than a pre-defined threshold. To address this problem, we present a post optimization method which utilizes the regularity of windows distribution.

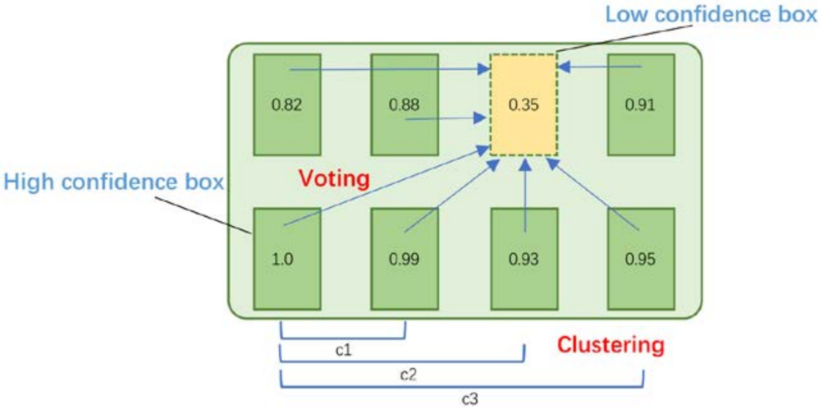


Fig. 7. Process of post optimization. The value in each box represents the confidence score as a window. c1, c2, c3 represent clusters of spacing distances. These blue arrows indicate the voting process.

We first obtain a set of window bounding boxes with higher confidence scores through WD-Net. Then the spacing distances of windows bounding boxes are calculated along with the horizontal and vertical directions, respectively. We also cluster these distances according to a threshold. Finally, we obtain distances between low confidence bounding boxes and high confidence bounding boxes. If a distance belongs to a cluster, the number of votes from high confidence bounding box increases. If a low confidence bounding box gets more than half of the votes, the bounding box is outputted as other high confidence bounding boxes. The process of post optimization method is illustrated in **Fig. 7**. By integrating the post optimization process, our window detection method is able to perform well by using both the powerful CNN detectors and the regular distribution of windows.

5. Experiments

In this section, we first provide some settings and metrics used in our experiments. Then, the experimental results with analyses are presented on both the ECP dataset and the proposed SSWD dataset. Finally, we verify the post optimization method.

5.1 Experimental settings

Training details: We perform experiments on the ECP dataset and the proposed SSWD dataset. In our WD-Net, we set the shorter edge of the input image to 600 pixels and the longer edge to no more than 1000 pixels. We adopt the VGGNet pre-trained on ImageNet [30] as the backbone of the model. Window RPN is used to propose candidate regions, and 256 anchors are sampled per image with a 1:3 ratio of positive to negative anchors. RoIAlign is adopted in all experiments. We use the Momentum optimizer to optimize the training loss with 0.9 momentum and 0.0001 weight decay. The new parameters are initialized by Xaiver [31]. We only use standard horizontal flipping for data augmentations. Our model is trained on a single 1080Ti GPU with an initial learning rate of 0.001.

Inference: During testing, the input images are first resized as the training stage. The max number of detection is 100 and the confidence score is 0.05 per image. Non-maximum suppression (NMS) with a threshold of 0.3 is applied to all predictions.

Metrics: To evaluate the detection results, we use the typical precision rate, recall rate and mean Average Precision(mAP):

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

where, TP, FP, FN denote the true positive, false positive and false negative, respectively. The standard Intersection over union (IoU) criterion is employed to evaluate the overlapping area of bounding boxes.

$$mAP = \int_0^1 P(R)dR \tag{6}$$

where the P and R represent the precision rate and recall rate in the above. The mAP solves the single-point value limitations of precision rate and recall rate. It is able to reflect the global performance of detection methods.

5.2 Experiments on ECP Dataset

ECP [26] is a pixel-wise classification dataset which contains 104 rectified images of facades of Haussmannian style buildings. The original dataset is used to parse facade into semantic elements, such as windows, doors, balconies, etc. To experiment on this dataset, we use the windows mask to create ground truth bounding boxes for the detection task. We randomly divide the ECP dataset into a training set (74 images) and test set (30 images) to evaluate the performance of window detectors. The sizes of windows in the ECP dataset are generally small. We modify the original anchor configuration to 9 anchor boxes with 3 aspect ratios of 1:1, 1:2 and 2:1, and with box areas of 32^2, 64^2 and 128^2.

In **Table 2**, we present the detection results on the ECP dataset. There are four detection algorithms: original Faster R-CNN, our WD-Net without CEM, our WD-Net and our full method. Compared with Faster R-CNN, our WD-Net achieves better mAP, In addition, since we adopt CEM on WD-Net, there is a significant improvement compared with baseline. The full method achieves the best mAP among the four methods, in which the pyramid features can handle multi-size windows, CEM provides diverse receptive fields to better capture extreme window instances, and the regularity-based post optimization helps discover missed windows.

Table 2. Detection results on the ECP dataset. Bold fonts indicate the best mAP.

Method	mAP (%)
Faster R-CNN [14]	91.196
WD-Net – CEM	91.431
WD-Net	91.678
WD-Net + Post	**91.680**

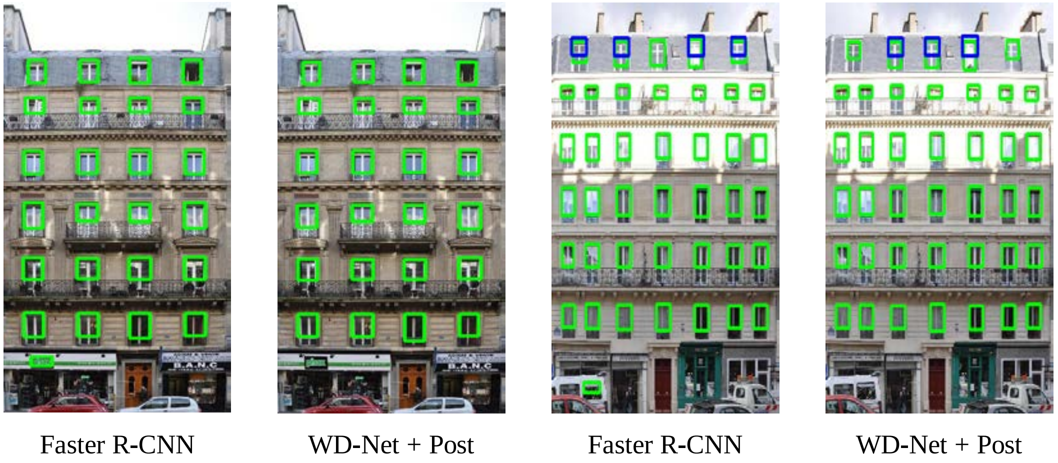

Faster R-CNN WD-Net + Post Faster R-CNN WD-Net + Post

Fig. 8. Qualitative results on the ECP dataset. Green boxes are predicted windows, and blue boxes are the missed ones.

We show some visual experimental results on the ECP dataset in **Fig. 8**. The detection results (bounding box in green) are directly outputted by each method and the ground truth bounding boxes are in blue. Here, we compare the original Faster R-CNN with our full method. We can see that Faster R-CNN and our method have nearly the same performances on the ECP dataset. Because all of the images in the ECP dataset are rectified and viewed in a frontal direction which makes the window detection task relatively easy. In the first column and second column, we can see that Faster R-CNN mistakes a part of the shop as a window. Furthermore, in the third and fourth column, the missed windows of our method are less than those of Faster R-CNN. This demonstrates that our new architecture with multiple scales features enhanced by context has advantages on window detection.

5.3 Experiments on SSWD Dataset

Table 3 summarizes detection results on the SSWD dataset of different methods. Faster R-CNN is a baseline setting. Then we incrementally add our improvements: Window RPN and RoI feature fusion, CEM and Post. Compared with Faster R-CNN, adding our new architecture improves the mAP. It shows that the Window RPN and RoI feature fusion are useful for detecting windows. The result is further improved by CEM. It is demonstrated that our CEM is helpful to capture various window shapes and distributions for better accuracy by multi-scale features with the diverse receptive fields. Finally, with our post optimization, some missed windows are detected by using the regular distribution of windows.

Table 3. Detection results on the SSWD dataset. Bold fonts indicate the best mAP.

Method	mAP (%)
Faster R-CNN [14]	92.542
WD-Net – CEM	92.599
WD-Net	92.862
WD-Net + Post	**93.086**

We showcase some detection results on the SSWD dataset in Fig. 9. The predicted detection results (bounding box in green) are directly outputted by each method and the ground truth bounding boxes are in blue. The first column of Fig. 9 shows that the detection results in the case of dense small windows, our method achieves better performances thanks to the multi-scale feature integration. The second column is the images with heavy occlusions. With the diverse receptive field obtained by our CEM, our WD-Net has advantages in these extreme window instances. As shown in the third column, window detection architecture can detect more windows which is missed by Faster R-CNN.

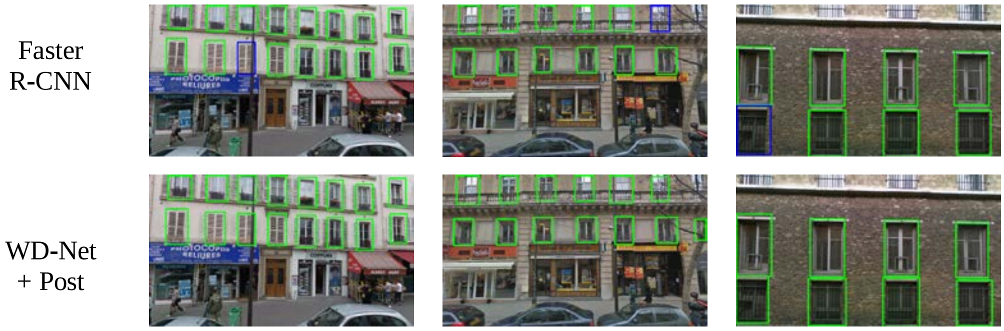


Fig. 9. Qualitative results of different methods on the SSWD dataset. Green boxes are predicted windows, and blue boxes are the missed ones.

5.4 Experiments on Post Optimization Method

In **Fig. 10**, we demonstrate the performance of our post optimization method. Meanwhile, we apply our post processing after Faster R-CNN and our WD-Net. The threshold of high confidence score is set 0.05, and the low confidence score is set 0.01. The blue bounding boxes in the left column of **Fig. 10** are the lost boxes by the two detection algorithms. We can also learn the confidence score at the right images that is lower than the standard threshold 0.05. Those low confidence boxes (blue box) are output as high confidence ones (corresponding green boxes in the right column), thanks to the post optimization method which uses the distribution regularity of windows to recheck these proposals.

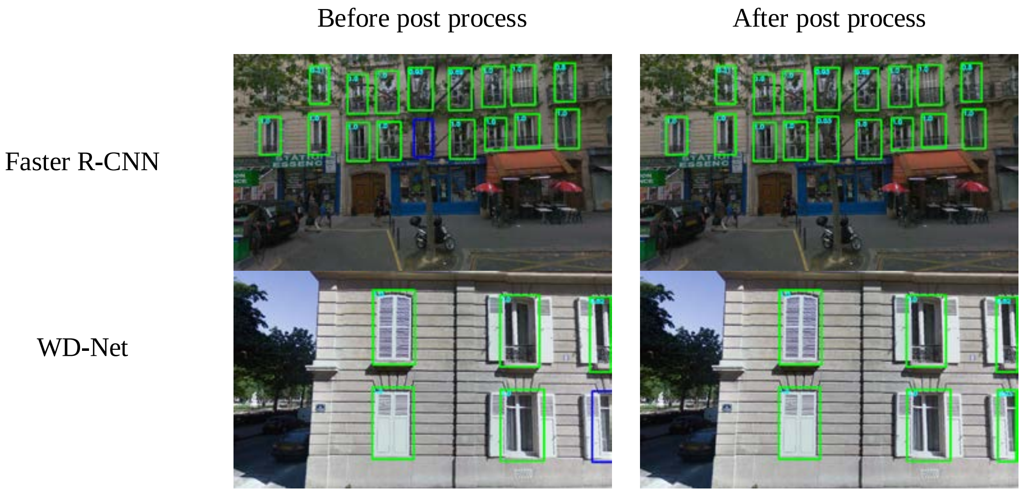


Fig. 10. Qualitative results of different methods on the SSWD dataset. Green boxes are predicted windows, and blue boxes denote the missing ones. The number in each box denotes the confidence score.

6. Conclusions

In this paper, we proposed a Street Scene Window Detection dataset and an effective window detection architecture. As far as we know, SSWD is the largest dataset specially built for window detection task which might be helpful to the research community. Our window detection architecture mainly contains WD-Net and a post optimization method. WD-Net features a new anchor strategy designed by the width and height distribution of the dataset, an RoI feature fusion module that fuses multi-scale features and a context enhancement module that can diversify the receptive fields of features. The post optimization, relying on the regular distributions of windows of buildings can further detect missed windows. Experiments on our SSWD dataset and ECP dataset show that our method obtains state-of-the-art performance.

References

[1] Andrea Cohen, Johannes L. Schönberger, Pablo Speciale, Torsten Sattler, Jan-Michael Frahm and Marc Pollefeys, “Indoor-outdoor 3d reconstruction alignment,” in *Proc. of the European Conference on Computer Vision*, pp.285-300, October, 2016. Article (CrossRef Link)

[2] Haider Ali, Christin Seifert, Nitin Jindal, Lucas Paletta and Gerhard Paar, “Window detection in facades,” in *Proc. of the 14th International Conference on Image Analysis and Processing*, pp.837-842, September, 2007. Article (CrossRef Link)

[3] Marcel Neuhausen and Markus König, “Improved Window Detection in Facade Images,” in *Proc.*

of the Advances in Informatics and Computing in Civil and Construction Engineering, pp.537-543, January, 2019. Article (CrossRef Link)

[4] Michal Recky and Franz Leberl, "Michal Windows detection using k-means in CIE-Lab color space," in *Proc. of the 20th International Conference on Pattern Recognition*, pp.356-359., August, 2010. Article (CrossRef Link)

[5] Navneet Dalal and Bill Triggs, "Histograms of oriented gradients for human detection," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 886-893, June, 2005. Article (CrossRef Link)

[6] David G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110, 2004. Article (CrossRef Link)

[7] Paul Viola and Michael Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.511-518, December, 2001. Article (CrossRef Link)

[8] Christopher JC. Burges, "A tutorial on support vector machines for pattern recognition," *Data mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121-167, 1998. Article (CrossRef Link)

[9] Jiaquan Shen, Ningzhong Liu, Han Sun, Xiaoli Tao and Qiangyi Li, "Vehicle detection in aerial images based on hyper feature map in deep convolutional network," *KSII Transactions on Internet and Information Systems*, vol. 13, no. 4, pp. 1989-2011, 2019. Article (CrossRef Link)

[10] Haoxiang Li, Zhe Lin, Xiaohui Shen, Jonathan Brandt and Gang Hua, "A convolutional neural network cascade for face detection," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.5325-5334, June, 2015. Article (CrossRef Link)

[11] Martin Zlocha, Qi Dou and Ben Glocker, "Improving retinaNet for CT lesion detection with dense masks from weak RECIST labels," in *Proc. of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp.402-410, October, 2019. Article (CrossRef Link)

[12] Ross Girshick, Jeff Donahue, Trevor Darrell and Jitendra Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.580-587, June, 2014. Article (CrossRef Link)

[13] Ross Girshick, "Fast r-cnn," in *Proc. of the IEEE International Conference on Computer Vision*, pp.1440-1448, December, 2015. Article (CrossRef Link)

[14] Shaoqing Ren, Kaiming He, Ross Girshick and Jian Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Proc. of the Advances in Neural Information Processing Systems*, pp.91-99, December, 2015. Article (CrossRef Link)

[15] Joseph Redmon, Santosh Divvala, Ross Girshick and Ali Farhadi, "You only look once: Unified, real-time object detection," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.779-788, June, 2016. Article (CrossRef Link)

[16] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu and Alexander C. Berg, "Ssd: Single shot multibox detector," in *Proc. of the European Conference on Computer Vision*, pp.21-37, October, 2016. Article (CrossRef Link)

[17] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He and Piotr Dollár, "Focal loss for dense object detection," in *Proc. of the IEEE International Conference on Computer Vision*, pp.2980-2988, October, 2017. Article (CrossRef Link)

[18] Hei Law and Jia Deng, "Cornernet: Detecting objects as paired keypoints," in *Proc. of the European Conference on Computer Vision*, pp.734-750, 2019. Article (CrossRef Link)

[19] Seohee Park, Myunggeun Ji and Junchul Chun, "2D human pose estimation based on object detection using RGB-D information," *KSII Transactions on Internet and Information Systems*, vol. 12, no. 2, pp. 800-816, 2018. Article (CrossRef Link)

[20] Md Abu Layek, TaeChoong Chung and Eui-Nam Huh, "Remote distance measurement from a single image by automatic detection and perspective correction," *KSII Transactions on Internet and Information Systems*, vol. 13, no. 8, pp. 3981-4004, 2019. Article (CrossRef Link)

[21] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *Computer Vision and Pattern Recognition*, 2014. Article (CrossRef Link)

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun, “Deep residual learning for image recognition,” in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.770-778, June, 2016. Article (CrossRef Link)

[23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár and C. Lawrence Zitnick, “Microsoft coco: Common objects in context,” in *Proc. of the European Conference on Computer Vision*, pp.740-755, September, 2014. Article (CrossRef Link)

[24] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso and Antonio Torralba, “Scene parsing through ade20k dataset,” in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.633-641, June, 2017. Article (CrossRef Link)

[25] Radim Tyleček and Radim Šára, “Spatial pattern templates for recognition of objects with regular structure,” in *Proc. of the German Conference on Pattern Recognition*, pp.364-374, September, 2013. Article (CrossRef Link)

[26] O. Teboul, “Ecole centrale paris facades database,” (Web Link).

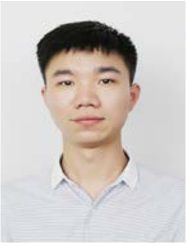
[27] Carl Doersch, Saurabh Singh, Abhinav Gupta, Josef Sivic and Alexei Efros, “What makes paris look like paris?,” *ACM Transaction on Graphics*, vol. 31, no. 4, pp.2-5, 2012. Article (CrossRef Link)

[28] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn and Andrew Zisserman, “The pascal visual object classes challenge: A retrospective,” *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98-136, 2015. Article (CrossRef Link)

[29] Kaiming He, Georgia Gkioxari, Piotr Dollár and Ross Girshick, “Mask r-cnn,” in *Proc. of the IEEE International Conference on Computer Vision*, pp.2961-2969, October, 2017. Article (CrossRef Link)

[30] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li and Li Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.248-255, June, 2009. Article (CrossRef Link)

[31] Xavier Glorot and Yoshua Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *Proc. of the thirteenth International Conference on Artificial Intelligence and Statistics*, pp.249-256, May, 2010. Article (CrossRef Link)



Wenguang Ma was born in 1996. He received the B.S. degree in Computer Science and Technology from Beijing University of Technology, Beijing, China in 2018. He is currently pursuing the M.S. degree in Beijing University of Technology. His research interests include Computer Vision and Machine Learning.

Wei Ma received her Ph.D. degree in Computer Science from Peking University, in 2009. She is currently an Associate Professor at the Faculty of Information Technology, Beijing University of Technology, China. Her research interests include Image Processing, Computer Vision and e-Heritage.