Study on Improved YOLO_v3-based Algorithm for Identifying Open Windows on Building Facades

Guangmin Sun, Pengfei Lin*, Yu Li
Faculty of Information Technology, Beijing University of Technology, Beijing China

*Corresponding author’s email: Linpf_725@163.com

Abstract. In order to ensure the security of building facades in key areas and improve the security detection efficiency of security personnel on building facades, this paper proposes an improved YOLO v3-based algorithm for open window detection recognition in building facade images, which extracts open window features from images to make predictions on full images by convolutional neural network. Firstly, due to the absence of publicly available window datasets in the network and the high number of window types that exist in reality, a self-constructed window dataset containing 13573 images of open windows is used to train and test the window detection model. The data set is then clustered by the K-Means clustering algorithm to select an Anchor Box more suitable for window detection, which draws on the ShuffleNet idea to strengthen the feature extraction method, and then optimizes the network structure of YOLO v3. Finally, a block detection mechanism is introduced to effectively enhance the network's ability to detect small dense targets; experimental results show that the method improves the accuracy and speed of window detection and reduces the workload of security personnel in key areas to manually check for windows on both sides of the street. Fire detection, missing and falling bricks on the floor, and overhead throw detection are of great importance.

Keywords. YOLO v3, Window opening detection, Floor security

1. Introduction
With the rapid development of national economic construction and the increasing frequency of international and domestic exchanges, all kinds of large-scale activities and conferences continue to increase[1]. Especially during large events, the concentration of people and materials, the slightest negligence will easily cause accidents. The facade of the building often hides unknown risks, and whether the use of existing video surveillance, real-time monitoring of the building facade and alarm is particularly important. The monitoring and timely handling of dangerous behaviors, such as window throwing, has become an urgent problem for security work. In order to improve the situation, it is necessary to detect windows on the building facade in key areas to reduce the probability of dangerous behaviors caused by open windows.

In 2018, Wang Bo et al. proposed an image recognition-based window opening alarm detection method [2], which obtains RGB images through surveillance video, color space conversion of the images to obtain HIS images, and determines the window opening alarm location based on the
luminance component image data in the HIS images. The method relies on traditional image processing methods to obtain window opening features from images, and environmental factors such as light and luminance have a large impact on the results of window opening detection.

With the booming development of artificial intelligence, more and more deep learning algorithms based on GPU acceleration services are rapidly trained and applied in practice. Currently, deep learning-based target detection algorithms have been applied to various fields, with deep neural networks that can extract more complex features, with strong model expression and high detection accuracy. It mainly consists of two categories: the two-stage region-based target detection algorithms R-CNN [3-5] series and Faster R-CNN [5], which have accurate detection speed but cannot reach real-time; and the one-stage regression-based target detection algorithms SSD [6] series and YOLO [7-9] series, which have fast detection speed but inaccurate localization.

Open window detection also belongs to a class of target detection problems. In this paper, a window opening detection dataset is made using building facade surveillance video data. By borrowing the ShuffleNet idea, we improve the YOLO V3 model to predict the location of open windows. The dimensions of the target box are clustered to obtain nine Anchor Boxes suitable for window opening detection. The experiments show that the improved YOLO V3-based window opening detection algorithm has better detection accuracy and efficiency.

2. Algorithm

At present, most of the windows in high-rise buildings use sliding windows, casement windows and shot windows, but due to the location of the camera is installed at a distance from the building, the window in the opening, generally a small percentage of the picture, so how to detect the opening of the window in a timely manner such as a small target is the key to the problem. In this paper, we choose the popular YOLO v3 target detection algorithm, which is fast and simple, with high accuracy and low false detection rate, to meet the practical requirements of window detection. In order to strengthen the deficiencies of the YOLO v3 network in feature extraction and to alleviate the gradient disappearance problem of the feature extraction network, this paper introduces the ShuffleNet [10] module on top of the original network. It not only enhances the original network in feature transfer, maintaining accuracy while reducing the computational load of the model. In addition, considering the inadequacy of the original network in detecting small targets, a block detection mechanism is introduced, which greatly improves the robustness of the algorithm to small targets such as open windows.

2.1 YOLO

Joseph Redmon et al. proposed a new end-to-end object detection network in 2016, named YOLO [3]. The YOLO family of algorithms classifies the object detection problem as a regression problem, and a convolutional neural network can be used to predict the probability of an object's bounding box and category directly from the input image, thus enabling end-to-end detection. Through continuous improvements, Joseph Redmon et al. introduced the YOLO v2 [4] and YOLO v3 [5] target detection algorithms with more robust accuracy and speed in 2017 and 2018, respectively. The basic idea of the YOLO v3 algorithm is: the input image is extracted by feature extraction network, the input image is divided into 13*13 grids, and if the coordinates of the center of the real box of an object fall into a certain grid, then this grid is responsible for detecting the object. Since each grid point predicts a fixed number of bounding boxes, only the bounding box with the largest intersection ratio with the real box is used to predict the object. It achieves real time.

2.2 ShuffleNet

In this paper, ShuffleNet basic unit is introduced in the original Darknet-53 feature extraction network. The structure of ShuffleNet basic unit is shown in Fig.1 below. The idea of grouped convolution is adopted to group the different feature maps of the input layer, and then different convolution kernels are used to convolve each group, which reduces the computational amount of convolution and enhances the transfer of features.
Figure 1. Structure of the ShuffleNet base unit

The ShuffleNet unit is similar to the residual structure in the ResNet [13] network, and the basic idea is to replace the original network dense 1*1 convolution with a 1*1 group convolution, adding only a channel shuffle operation after the first 1*1 convolution. Also, the ReLU activation function is not used after the 3*3 depthwise convolution. The 3*3 average pooling with a step of 2 is used for the original input, and the depthwise convolution is taken with a step of 2 to ensure that the two channels have the same shape, and then the resulting feature map is concat-connected to the output to greatly reduce the computation and parameter size.

2.3 Improvements to the YOLO v3 feature extraction network

YOLO v3 uses Darknet-53 for its network structure, which is basically fully convolutional on one hand, and introduces the residual structure of ResNet [13] on the other hand, which makes it much less difficult to train the deep network and improves the accuracy significantly. However, Darknet-53 is only the feature extraction layer, that is, only the convolutional layer in front of the average pooling layer is used to extract features.

In order to strengthen the deficiencies of the YOLO v3 network in feature extraction and to alleviate the problem of gradient disappearance in the feature extraction network, this paper introduces the ShuffleNet [10] module on the basis of the original network, which makes use of cross-layer links to fully perform feature multiplexing between multiple layers, and uses the idea of packet convolution to greatly reduce the model parameters. The improved YOLO v3 feature extraction network is shown in Fig.2 below.
Figure 2. Structure of the ShuffleNet base unit

Among them, the DB L module represents the 1*1 ordinary convolutional layer, Batch Normalization layer, and Relu activation function; the G-DBL module represents the 1*1 packet convolutional layer, BN layer, and Relu activation function; and the DW-DB module represents the 3*3 deep convolutional layer and BN layer. Secondly, although the BN layer plays an active role in training the network, the network increases operations during forward inference, which affects the performance of the model and takes up more memory or explicit memory space. In this paper, the BN layer and the convolutional layer are fused to improve the speed of forward inference of the model.

2.4 Anchor Boxes clustering

In this paper, we mainly study the problem of open window detection. Windows are used as small targets in the whole image and have different sizes. Only through clustering algorithm can we find out the Anchor Boxes suitable for open window detection in this paper. To some extent, we can improve the speed and accuracy of small target detection.

In this paper, the k-means clustering algorithm is used to get all the target boxes in the dataset to get all the widths and heights. Random selection gets 9 clusters centered on 9 points, and continuously calculates the distance from other points to the points in the cluster and adjusts the cluster and center to which each point belongs until the 9 centers do not change anymore. The values x, y of these 9 center points are the 9 appropriate Anchor Boxes (the width and height of the target box) for the entire data. This paper does not change the number of 9 different sized anchor boxes generated by the original YOLO v3 network on the COCO dataset. For the input floor images, nine anchor box sizes suitable for this paper's detection were generated, namely 11*14, 11*22, 16*23, 18*28, 19*32, 22*23, 22*16, 23*29, and 28*33.

2.5 Streamlining the loss function

In terms of the loss function, YOLO v3 takes the form of mean square error for the regression prediction part and binary cross entropy for the classification part and confidence level. The target detection identification in this paper is only for the class of target objects such as open windows, so the classification loss is directly deleted in the calculation of the function [11]. As a result, the streamlined loss function is changed from the original three localization errors, classification error, and confidence level error to the localization error form and confidence level error.

The final loss function is:

\[
\text{Loss} = \lambda_{\text{coord}} \sum_{i=0}^{1} \sum_{j=0}^{\text{window}} \left( (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right) + \lambda_{\text{coord}} \sum_{i=0}^{1} \sum_{j=0}^{\text{window}} \left( \sqrt{w_i} - \sqrt{\hat{w}_j} \right)^2 + \sum_{c \in \text{class}} \left( p_i(c) - \hat{p}_j(c) \right)^2
\]

where (1) is the localization error and (2) is the confidence error. The horizontal and vertical coordinates of the real and predicted target coordinate frames are \((\hat{x}_i, \hat{y}_i)\) and \((x_i, y_i)\), respectively, the width and height of the real and predicted targets are \((\hat{w}_i, \hat{h}_i)\) and \((w_i, h_i)\), respectively, and the confidence levels of the real and predicted targets are \(p_i(c)\) and \(\hat{p}_j(c)\), \(\text{window}^i\) represents the predicted value of the jth bounding box where the object falls into grid i. The parameter \(\lambda_{\text{coord}}\) is the weight set by Loss, s represents the grid cell, and B represents each bounding box.

2.6 Block testing mechanism

Since the YOLO v3 model has a limit on the size of the input image, in YOLO's network, the convolutional layer is connected to two fully connected layers at the end, and the fully connected layer requires a fixed size vector as input, so it also requires the original image to have a fixed size 448*448. 448 was chosen because the image size satisfies a multiple of 32, and in DarkNet network, performing five steps of 2 of downsampling, at the very bottom, the feature map size needs to satisfy an odd
number of 13*13 to ensure that the center point falls in the only box. If it is an even number, the center point falls in the central 4 boxes, leading to ambiguity.

The floor image needs to be detected in a larger size, simply inputting the large image into the YOLO v3 model, the model will uniformly Resize the large image to a smaller size of 448*448, weakening the small target information such as windows in the image, making the loss of the more obvious open window features of the original image, which is not conducive to a better detection effect.

In this paper, a blocking detection mechanism is introduced, the main steps of which are: taking the upper left corner of the original image as a starting point, blocking the original image with a fixed window with fixed size and fixed sliding step, the fixed size of the window is 640*480, which is close to the input size of the original YOLO v3 network; the blocked image is then input into the network for detection, which can be processed by parallel acceleration, which not only improves the detection effect, but also keeps the detection efficiency basically unchanged. Finally, the blocking detection result is mapped to the original image.

3. Creation Of The Data Set

3.1 Image acquisition
In deep learning, the quality and quantity of samples will be a decisive factor for the model and will directly affect the accuracy of window opening recognition detection and the generalization ability of the model. Some large datasets openly available on the Internet for deep learning applications do not provide the open window dataset detected in this paper. In order to solve this problem, I built a window detection dataset OpenWindow, which is mainly obtained from video surveillance and web crawler. In this paper, several floor videos of actual scenes are selected, including floor videos under different lighting, backgrounds and angles. It includes five kinds of windows, such as sliding windows, lift-up windows, casement windows, shot windows, and top-hung windows, which are widely used in office buildings, residential buildings, and teaching buildings.

3.2 Image pre-processing
For the video data, I mainly use the inter-frame video interception method. Firstly, we read in the local floor video, and then we read each frame of the video, and save it as a .jpg format every F frame, so that we can get a picture of the slight change of the window opening on the floor in different states. In addition, due to the inconsistent resolution of the video surveillance cameras, the obtained images are normalized to the size of 1920*1080 on the basis of ensuring that the image details are not lost.

For the images obtained on the network, the existence of different formats and resolutions may cause the target detection network to be unable to read the images during training. So only the color image data in .jpg format that is the same as the one that can be processed in this paper is retained. And the images are normalized to the size of 448*448.

3.3 Image annotation
In this study, an improved YOLO v3 algorithm model is used to label the open window images using the image labeling tool Labeling. Since the open window detection algorithm designed in this paper identifies targets mainly for one type of windows, the label category is set to open_window. The labeled image generates an XML file, which mainly records the number of targets, target category, four coordinates of the target bounding box and other information. Finally for the 1920*1080 image, the original image is cropped into a small image block with a size of 640*640 with the open window as the center, and the labeled data set is made into a PASCAL-VOC format to save the data information.

4. Experimental Results And Analysis

4.1 Experimental environment and parameter configuration
The environment and parameter configuration of this experiment are shown in Table I below.
TABLE I. EXPERIMENTAL ENVIRONMENT CONFIGURATION

| System   | Windows10 |
|----------|------------|
| CPU      | Intel(R) Core(TM) i7-9400M CPU @2.90GHz |
| GPU      | NVIDIA GeForce GTX2070 |
| RAM      | 32G |
| SSD      | 500G |
| Library  | OPENCV3.4, CUDNN, CUDA10.2 |

4.2 Parameter setting and training method of the model

In this paper, an improved YOLO v3 model is used with a self-built OpenWindow dataset with 10014 training set images and 3559 test set images. The inputs of the model are 416*416, the initial learning rate is set to 0.001, the batch size is set to 128, the maximum number of iterations is determined according to the model evaluation metrics, and a multiscale training strategy is used to enhance the robustness of the model for images with different input sizes.

The validation set loss curves for the improved YOLO v3 algorithm used in this paper are shown in Fig.3.

4.3 Comparative experiments and analysis of results

Algorithm metrics based on open window detection. This paper adopts the criteria proposed in the literature [12], and the algorithm evaluation metrics used in this paper include Accuracy, Precision, and Recall.

Accuracy (Accuracy) is calculated by the formula:

\[
R_{\text{Accuracy}} = \frac{N_{TP} + N_{TN}}{N_{TP} + N_{TN} + N_{FP} + N_{FN}} (3)
\]

Precision is calculated by the formula:

\[
R_{\text{Precision}} = \frac{N_{TP}}{N_{TP} + N_{FP}} (4)
\]

Recall is calculated by the formula:

\[
R_{\text{Recall}} = \frac{N_{TP}}{N_{TP} + N_{FN}} (5)
\]

where TP (True Positive) represents predicting positive classes as positive and FN (False Negative) represents predicting positive classes as negative. FP (False Positive) represents predicting negative classes as positive, and TN (True Negative) represents predicting negative classes as negative.

Table II shows the comparison of the evaluation metrics of the improved YOLO v3 algorithm on the OpenWindow dataset before and after improvement. It is clear from Table 2 that the improved YOLO v3 algorithm has improved the accuracy, precision, and recall of OpenWindow target detection on the OpenWindow dataset compared to the original unimproved YOLO V3 algorithm.
TABLE II. COMPARISON OF EVALUATION METRICS OF YOLO V3 ALGORITHM BEFORE AND AFTER IMPROVEMENT

| Algorithm  | Accuracy | Precision | Recall |
|------------|----------|-----------|--------|
| YOLOV3     | 0.91     | 98.49%    | 0.98   |
| Improved   | 0.92     | 99.49%    | 0.99   |

Based on the above experimental comparison, the OpenWindow dataset created in this paper was used to train and test different target detection algorithms in order to reflect the contribution of the improved YOLO v3 target detection algorithm in this paper. The detection algorithm mainly includes two-stage target detection algorithm Faster R-CNN, one-stage target detection algorithm SSD, YOLO v3, and improved YOLO v3. The algorithm is mainly evaluated for the two metrics of average accuracy and detection speed. The specific comparison results are shown in Table III below.

TABLE III. COMPARISON OF THE RESULTS OF DIFFERENT TARGET DETECTION ALGORITHMS

| Algorithm  | Precision | Speed   |
|------------|-----------|---------|
| Faster R-CNN | 96.54% | 1.02s   |
| SSD        | 84.76%   | 0.0287s |
| YOLO v3    | 98.49%   | 0.0245s |
| Improved   | 99.49%   | 0.0145s |

In order to verify the effectiveness of the improved YOLO v3 target detection algorithm in detecting windows on building facades in different scenarios, this paper selects five building floor videos obtained in real scenarios and performs experimental detection on a sub-basis. A comparison of the window detection effects of the YOLO v3 algorithm before and after the improvement is shown in Figure 4 below.

This paper experimentally shows that when there is an entire glass curtain wall on the façade of the modern building in the video and the feature of whether the window is open or not is not obvious, the algorithm all accurately frames the location of the open window in the image, which shows that the algorithm can strongly suppress false and misreporting, has strong robustness and model generalization ability. From the comparison in the figure below, it can be seen that the improved YOLO v3 algorithm has superior performance in the presence of occlusion, image rotation, and whether or not it misses detection.

**Figure 4.** Detection effect of different algorithms before and after improvement

The results of the improved YOLO v3 algorithm for window detection on the same building façade image are shown in Fig.5.
Figure 5. Results of open window detection by the improved YOLO V3 algorithm for images of the same building façade

The algorithm has better detection results under different illumination conditions and in the detection of dense small targets. It shows that the algorithm has a high performance in detecting window openings under different lighting conditions.

The improved YOLO V3 algorithm's window opening detection results for windows of different structures (sliding, casement, and shot windows) on the building façade images are shown in Fig. 6 below. As can be seen from the figure below, the algorithm has a good detection effect on the window opening behavior of windows with different structures, which also verifies the strong model generalization ability of the algorithm.

Figure 6. Effect of the improved YOLO V3 algorithm on the detection of open windows for different structures

5. Conclusion

Existing window-opening detection algorithms use traditional image processing methods for detection, due to the large effect produced by illumination and the poor real-time performance of the algorithm. Therefore, this paper proposes an improved YOLO v3-based windowing detection algorithm, which draws on the ShuffleNet idea to enhance the feature extraction method, reduce the network parameters, and improve the speed of target detection. And a scenario-rich dataset containing various windowing targets is constructed for training and testing. Experiments show that the algorithm in this paper outperforms other detection algorithms in all aspects, and the algorithm is robust to lighting and other environments, and can meet the real-time requirements. The good effect of this algorithm for open window detection is verified. In summary, the algorithm has important use value and application prospects.

6. Acknowledgement

This study was supported by the National Natural Science Foundation of China [11527801, 41706201].

7. References

[1] Z. He 2016 "Talk about how to do a better job of fire security at major events". Fire science and technology, vol. 35, no. 9, pp. 1327–1329.
[2] B. Wang, X. Sun, W. Zhao, J. Yang, C. Yang, and Y. Liu 2018 "An image recognition based window opening alarm detection system and method". Beijing, China, August.
[3] R. Girshick, J. Donahue, T. Darrell and J. Malik 2014 "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 2014 pp. 580-587.
[4] R. Girshick 2015 "Fast R-CNN," in 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, pp. 1440-1448.

[5] S. Ren, K. He, R. Girshick and J. Sun 2017 "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks" in IEEE Transactions on Pattern Analysis & Machine Intelligence, vol. 39, no. 06, pp. 1137-1149.

[6] W. Liu, D. Anguelov, D. Erhan, S. Christian, R. Scott, C. Fu and B. Alexander 2016 "SSD: Single shot multi Box detector," in 2016 European Conference on Computer Vision, Ithaca, USA, pp. 21-37.

[7] J. Redmon, S. Divvala, R. Girshick and A. Farhadi 2016 "You Only Look Once: Unified, Real-Time Object Detection," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, pp. 779-788.

[8] J. Redmon and A. Farhadi 2017 "YOLO9000: Better, Faster, Stronger," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, pp. 6517-6525.

[9] J. Redmon and A. Farhadi 2018 "YOLOv3: An Incremental Improvement," in 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1-6.

[10] X. Zhang, X. Zhou, M. Lin and J. Sun,"ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices," in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 2018 pp. 6848-6856.

[11] W. Zhang, J. Wei 2020 "Improved YOLO v3 Fire Detection Algorithm Embedded with DenseNet Architecture and Cavity Convolution Module," Journal of Tianjin University (Natural Sciences and Engineering Technology), vol. 53, no. 9, pp. 976–983

[12] P. Dollar, C. Wojek, B. Schiele and P. Perona 2009 "Pedestrian detection: A benchmark," in 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops), Miami, FL, pp. 304-311.

[13] K. He, X. Zhang, S. Ren and J. Sun 2016 "Deep Residual Learning for Image Recognition," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, pp. 770-778.