CenterNet-Triplets application: surveillance camera illegal management detection

Xiangxiang Zhu¹, Li Yang¹, Dongping Zhang¹* and Ping Chen²*

¹Key Laboratory of Electromagnetic Wave Information Technology and Metrology of Zhejiang Province, China Jiliang University, Hangzhou, Zhejiang, CJLU, China
²Zhejiang Provincial Testing Institute of Electronic Information Products, Hangzhou, Zhejiang, China

*06a0303103@cjlu.edu.cn
*Corresponding author’s e-mail: 2581089700@qq.com

Abstract. There are two types of illegal business in city management: out-of-store management and illegal mobile management, we simply refer to as outmanagement and flowmanagement. The main way to deal with illegal business is manual management. However, this requires a lot of costs, and it is impossible for city management personnel to conduct 24-hour inspections. In order to improve management efficiency and save labor costs, this article uses city surveillance cameras, combined with deep learning object detection methods, to conduct real-time detection of illegal operations. Aiming at the situation where the amount of surveillance video data is small, a data enhancement method is proposed to improve the detection effect. CenterNet-Triplets is an object detection algorithm without anchor boxes. Its backbone network is Hourglass Network. The AP of CenterNet-Triplets on MS-COCO dataset is 47.0%, and it has a good detection speed. Experimental results show that CenterNet-Triplets is suitable as a method for detecting illegal operations in city management.

1. Introduction
With the development of society, in the process of urbanization, there are many problems in urban management. City management often requires a lot of labor costs. The application of artificial intelligence object technology to certain aspects of city management can save a lot of costs and meet the requirements of real-time detection, which is an application innovation that conforms to the concept of sustainable development. As shown in Figure 1 below illegal business in cities mainly include outmanagement and process flowmanagement, and unqualified outmanagement and flowmanagement will not only affect the appearance of the city, but also affect the urban traffic in severe cases, causing adverse effects. This article uses deep learning object detection technology to model the two types of illegal business, outmanagement and flowmanagement, and uses surveillance cameras to detect outmanagement and process flowmanagement in real time, and achieves a high accuracy rate.
2. Related work

Since the development of object detection, it has been applied in many fields, which can be divided into two categories: two-stages and one-stage.

2.1. Two-stages

The two-stage object detection algorithm is the first category, and the more famous algorithm is the R-CNN series. R-CNN [1] is one of the pioneering works of deep learning object detection, using selective search to predict the bounding box. Each candidate region of R-CNN must propagate in the forward direction, and many candidate regions overlap each other, resulting in duplication and slower speed. Fast-RCNN [2], like R-CNN, still obtains candidate boxes through selective search, but Fast-RCNN directly passes the input image through the pre-training model, maps the candidate box to the feature map to extract the region of interest, and then the regions of different sizes pass through the RoI-Pooling layer get feature vectors of the same size, and finally get the prediction of the category and bounding box through two fully connected layers. Faster-RCNN [3] proposes to replace selective search with RPN. RPN automatically learns and extracts good candidate regions through the network, which can reduce the number of candidate regions, increase speed and ensure accuracy. Mask-RCNN [4] adds a prediction branch based on Faster-RCNN, which can simultaneously detect objects and predict their length. R-FCN [5] replaces fully connected layers with position-sensitive score maps for better detection of objects. Cascade R-CNN [6] trained a series of detectors by increasing the IoU threshold, solving the problem of overfitting during training and quality mismatch during inference.

2.2. One-stage

The one-stage object detection algorithm is the second category. The one-stage algorithm includes the Anchor-boxes and Anchor-free series. The most representative algorithms based on Anchor-boxes are Yolov2-v4 series [7-9], SSD [10] series, RetinaNet [11] series and so on. The Yolo series regards the image as a grid of SxS, and then predicts the bounding boxes, and Yolov3 achieves higher real-time and mAP. Compared with YOLOv3, the improvement of YOLOv4 mainly includes the following four aspects. Firstly, data enhancement, including Mosaic data enhancement, cmBN, and SAT self-confrontation training. Secondly, YOLOv4 combines various new methods including CSPDarknet53, Mish activation function, and Dropblock [12]. Thirdly, YOLOv4 insert some performance-enhancing layers, such as SPP [13] module, FPN [14]+PAN structure. Finally, the anchor box mechanism of Yolov4's output layer is the same as Yolov3, the main improvement is the loss function CIOU_Loss during training, and the nms of the prediction box screening becomes DIOU_nms. The SSD extracts features by truncating the VGG network of the fully connected layer, generates anchor boxes on feature maps at multiple scales, predicts categories and bounding boxes, and the method is similar to Faster-RCNN. For feature maps with a relatively large width and height, the field is small and there
are many anchor frames, suitable for detecting small objects. For feature maps with a relatively small width and height, the field is large and there are less anchor frames, suitable for detecting large objects. SSD is still a multi-scale object detection network. RetinaNet pointed out that the reason for the poor performance of the one-stage algorithm is caused by the imbalance between the foreground and background categories. In order to solve this problem, this article proposes Focal Loss. It solves the problem of category imbalance by simply changing the loss function.

Anchor-free series of algorithms refers to the elimination of the anchor boxes in the object detection algorithm. Its main representative methods are YOLOv1 [15], CenterNet [16], CornerNet [17], CenterNet-Triplets [18], and FCOS [19]. The main idea of CenterNet is to return the attributes of other bounding boxes through the information of the center point, such as the distance of the center point and the four sides, posture, and direction. First, CenterNet will calculate the center point heat map, and then directly return the information that needs to be used through the network. This method is simple, fast, and efficient without any NMS post-processing operations, and can be trained directly end-to-end. However, using only the center point for regression obviously results in too little information obtained, which may not be sufficient to support the return of relatively effective information, which ultimately affects the detection performance. However, it may be due to the fact that its regression information is sufficient, which enhances the representation ability of various information makes it possible to improve the results. Compared to CenterNet's return of the boundary distance from the center point to obtain the bounding boxes, CornerNet does the opposite, directly using the upper left and lower right corner points to define the bound boxes, and a set of corner points to determine an object. CenterNet-Triplets has improved CornerNet. It integrates the center and corner information, which is equivalent to adding the center point information as one of the criteria based on CornerNet, showing superior performance.

3. Data augmentation

The data comes from manual collection and city monitoring, in which more images are collected manually and less in the city monitoring. Most of the manually collected images are 720*960 size and the actual monitoring scene images are 1920*1080 or larger. If the manually collected images are directly sent to the network training, it will affect the detection effect, so you need to use the texture method to manually collect images are used for data enhancement. As shown in Figure 2, the manually collected images are randomly pasted below the background monitoring image, where the background image selects a monitoring scene picture without outmanagement and flowmanagement, and uses multiple backgrounds, about 30 of which contains the monitoring background image for 2 nights.

![Figure 2. The process of data augmentation, paste the foreground image into the background image.](image-url)
4. Network structure

Experiments show that CenterNet-Triplets has higher accuracy and can meet the requirements of real-time detection. This is because in actual monitoring scenarios, it is not necessary to detect every frame. Although YOLOv4 has a faster speed, but the accuracy is slightly lost to CenterNet-Triplets, which has been reflected in the COCO dataset. YOLOv4 has a mAP of approximately 43.5% on the COCO dataset and CenterNet-Triplets is approximately 47.0%. According to the actual situation, CenterNet-Triplets was selected as the detection network. CenterNet-Triplets can be divided into three parts: image preprocessing, hourglass network Hourglass104 and network output layer. In addition, as can be seen from Figure 3 below, the network output layer can be divided into three branches, outputting the heatmaps, embeddings, and offsets of the upper left corner point, the lower right corner point, and the center point, respectively. During training, the input image size is 511*511, and features are extracted through Hourglass-104, and then divided into three branches of the network. The two branches through Cascade corner pooling are used to get the upper left corner and the lower right corner, respectively, and the center pooling branch. Get the center point. Cascade corner pooling and Center pooling have been described in CenterNet-Triplets, which will not be repeated here.

![CenterNet-Triplets network structure](image)

**Figure 3.** CenterNet-Triplets network structure.

5. Experiments

5.1. Dataset introduction

The data set in this paper comes from city video surveillance and manual collection. There are about 10,000 images from the city video surveillance. In this article, the scenes collected by the city video surveillance are not many, about the scenes captured by 200 cameras, and the data are mostly from the different states of these cameras at different times. What is described in Figure 1 above is the image collected by video surveillance. Another part of the data used in this article comes from manual collection, and the number of images collected by manual collection is about 80,000, but the size of the images is different. The manually collected images are shown in Figure 4 below.
Figure 4. Data of manual collection, the two images on the left are outmangement, and the two images on the right are flowmanagement and central merge branch.

5.2. Training parameter settings
The main configuration of the server is NVIDIA RTX 2080ti graphics cards and 128GB RAM. The program runs under the Ubuntu 18.04 system, and simultaneously calls the parallel computing architecture (CUDA), NVIDIA neural network library (Cudnn), and open source computer vision library (OPENCV). Both the test set and the verification set are city surveillance video images, the size is 1920*1080, and some scenes do not appear in the training set. In this experiment, the size of the training set is 87000 images, and the size of the test set and verification set is 1200 images. Training iterates a total of 100,000 times, and reduces the learning rate once every 50,000 and 80,000 times.

Table 1. CenterNet-Triplets hyperparameter setting table.

| Hyperparameter       | Value               |
|----------------------|---------------------|
| batch_size           | 24                  |
| learning_rate        | 0.0025              |
| max_iter             | 100000              |
| stepsze              | 50000,80000         |
| decay_rate           | 10                  |
| opt_algo             | Adam                |
| input_size           | [511, 511]          |
| output_sizes         | [128, 128]          |
| random_scale         | 0.6-1.4             |
| random_crop          | True                |
| random_color         | True                |
| top_k                | 100                 |
| nms_threshold        | 0.5                 |
| training set         | 87000               |
| testing set          | 1200                |
| val set              | 1200                |

5.3. Experimental results
The experiment is a comparison between CenterNet-Triplets and yolov4, Compare the original data with the completed data set, and all experiments use multiple scales. The experimental results show that the data-enhanced data set has a higher AP than the original dataset. Although yolov4 has a higher FPS, CenterNet-Triplets has a higher AP than Yolov4.
Table 2. Experimental results of the enhanced data set and the original dataset

| Method                        | Backbone      | Size       | Data Augmentation | AP  |
|-------------------------------|---------------|------------|-------------------|-----|
| CenterNet-Triplets (multi-scale) | Hourglass52   | 511*511    | yes               | 55.9% |
| CenterNet-Triplets (multi-scale) | Hourglass104  | 511*511    | yes               | 59.6% |
| CenterNet-Triplets (multi-scale) | Hourglass52   | 511*511    | no                | 42.5% |
| CenterNet-Triplets (multi-scale) | Hourglass104  | 511*511    | no                | 47.4% |
| Yolov4                        | CSPDarknet53  | 416*416    | yes               | 53.0% |
| Yolov4                        | CSPDarknet53  | 512*512    | yes               | 56.4% |
| Yolov4                        | CSPDarknet53  | 608*608    | yes               | 57.7% |
| Yolov4                        | CSPDarknet53  | 416*416    | no                | 41.2% |
| Yolov4                        | CSPDarknet53  | 512*512    | no                | 45.6% |

6. Conclusion
The experimental results show that the data enhancement method can improve the detection outmanagement and flowmagement in the city, and because the required speed is not too high, the target detection network CenterNet-Triplets with higher accuracy can be selected.

Acknowledgments
This work was supported by Key Research and Development Projects in Zhejiang Province (No. 2020C03104), Zhejiang Provincial NSF (No. LY19F030012) and Key Laboratory of Information Security of Zhejiang Province (No.KF201910).

References
[1] R. Girshick, J. Donahue, T. Darrell, and J. Malik. (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR, 1, 3, 4, 8
[2] R. Girshick. (2015) Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision, pages 1440–1448.
[3] S. Ren, K. He, R. Girshick, and J. Sun. (2015) Faster r-cnn: Towards real-time object detection with region proposal networks. arXiv preprint arXiv:1506.01497.
[4] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. (2017) Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pages 2961–2969.
[5] J. Dai, Y. Li, K. He, and J. Sun. (2016) R-fcn: Object detection via region-based fully convolutional networks. In: Advances in neural information processing systems, pages 379–387.
[6] Z. Cai and N. Vasconcelos. (2018) Cascade r-cnn: Delving into high quality object detection. In: Proceedings of the IEEE Conference on computer vision and pattern recognition, pages 6154–6162.
[7] Joseph Redmon and Ali Farhadi. (2017) YOLO9000: better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 7263–7271.
[8] Joseph Redmon and Ali Farhadi. (2018) YOLOv3: An incremental improvement. arXiv preprint arXiv:1804.02767. 2, 4, 7, 11.
[9] Bochkovskiy A, Wang C Y, Liao H Y M. YOLOv4: Optimal Speed and Accuracy of Object Detection[J]. arXiv preprint arXiv:2004.10934, 2020.
[10] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. (2016. 2, 11) SSD: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision (ECCV), pages 21–37.
[11] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Doll´ar. (2017) Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), pages 2980–2988. 2, 3, 11, 13.

[12] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. (2018. 3) DropBlock: A regularization method for convolutional networks. In Advances in Neural Information Processing Systems (NIPS), pages 10727–10737.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. (2015. 2) Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 37(9):1904–1916, 4, 7.

[14] Tsung-Yi Lin, Piotr Doll´ar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. (2017. 2) Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2117–2125.

[15] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. (2016) You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Sergey Ioffe and Christian Szegedy.

[16] Xingyi Zhou, Dequan Wang, and Philipp Kr¨ahenb¨uhl. (2019) “Objects as points,”. In arXiv preprint arXiv:1904.07850.

[17] Hei Law and Jia Deng. (2018. 2, 11) CornerNet: Detecting objects as paired keypoints. In Proceedings of the European Conference on Computer Vision (ECCV), pages 734–750.

[18] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. (2019. 2, 12) CenterNet: Keypoint triplets for object detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), pages 6569–6578.

[19] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. (2019. 2) FCOS: Fully convolutional one-stage object detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), pages 9627–9636.