An Multi-feature Fusion Object Detection System for Mobile IoT Devices and Edge Computing

Xingyu Feng[1], Han Cao[1], and Qindong Sun[1,2(✉)]

[1] School of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, Shaanxi, China
sqd@xaut.edu.cn

[2] Shaanxi Key Laboratory of Network Computing and Security, Xi'an University of Technology, Xi'an 710048, Shaanxi, China

Abstract. With the increase of data scale and computing power, deep learning algorithm has made a prominent breakthrough in computer vision and other complex problems. However, its high complexity and large memory requirements make it very difficult to run in real time on the Internet of things terminal mobile devices. There is still delay the employing of cloud services cannot meet the real-time requirement. With the popularity of mobile terminal devices and the development of Internet of things, it is of great significance to design a real-time deep learning algorithm on IOT edge mobile devices with limited computing and memory resources. This paper proposes a new object detection method based on the current state-of-the-art object detection deep network model RetinaNet and traditional feature extraction method SIFT. RetinaNet is a one-stage detector with excellent detection speed and accuracy. We use RetinaNet as the object location method, then extract the CNN features and SIFT features of the fixed position image and combine them to train a new classifier. The object classification result will be based on the final classifier.

Keywords: Object detection · Deep learning · SIFT · IoT

1 Introduction

It is a great challenge to deploy applications based on deep learning technology on mobile devices. How to reduce the time complexity and memory requirements on the premise of ensuring the accuracy of the algorithm, so that it can run efficiently on mobile devices is an urgent problem to be solved.

Object detection aims at locating and classifying objects accurately from an image, which has practical application value, such as intelligent video surveillance [1], robot perception [2] and so on. Traditional object detection is roughly divided into three steps: (1) region selection, (2) feature extraction, and (3)

D. Wang et al. (Eds.): SPNCE 2020, LNICST 344, pp. 335–340, 2021.
https://doi.org/10.1007/978-3-030-66922-5_23

object classification. Feature extraction can be performed by Haar, HOG, LBP and SIFT methods. Compared with traditional methods, object detection based on deep learning has greatly improved the accuracy. At present, there are R-CNN [3], SSD [4], YOLOV1-YOLOV4 [5–8], and others.

The development of artificial intelligence has set off a wave of unmanned retail. The unrestricted placement of unmanned retail scenes poses challenges for object detection: (1) The same model shows different accuracy in different scenarios. (2) In this scenario, mobile devices need to be able to process images in real time. In order to solve this problem, this paper proposes an embedded object detection system for internet of things and mobile edge computing.

The structure of this paper is as follows, the next section introduces some work related to object detection. The Sect. 3 systematic elaborates the method proposed in this paper, and then shows the experimental results. Finally, we summarize the work done in this paper.

2 Related Work

At present, industry and academia have been committed to running deep learning applications on mobile devices. Srinivasan et al. [9] had trained lightweight object detection networks to help visually impaired users identify and locate household items through wearable devices. Blanco-Filgueira et al. [10] proposed an end-to-end solution for multi-object tracking based on deep learning for embedded and low-power IoT platforms.

Prior to 2012, object detection mainly uses SIFT, HOG, DPMs and other traditional features. In 2012, Krizhevsky et al. used 1.2 million data containing 1,000 different categories to train a deep CNN, which became the most advanced technology at the time. The methods based on CNN breaks through the bottleneck of traditional methods.

R-CNN [3] is an object detection method based on the candidate region method. Later, improved versions Fast R-CNN [11] and Faster R-CNN [12] appeared. OverFeat [13] is the first one-stage detectors based on deep learning and won the ILSVRC2013. After that, YOLO and SSD appeared one after another. At present, YOLO has four versions. Lin et al. [14] pointed out that one-stage detector detection accuracy is limited by class imbalance, thereby focal loss is proposed to solve this problem, and the network structure is named RetinaNet.

Therefore, this paper uses RetinaNet for object location, integrates CNN and SIFT features to classify, so that the final model has strong robustness to changes in the external environment.

3 Approach

The location of the object is obtained by using RetinaNet, and then the CNN features and SIFT features of the object image are fused, finally the category of object is obtained. The overall process of the framework is as follows (Fig. 1):

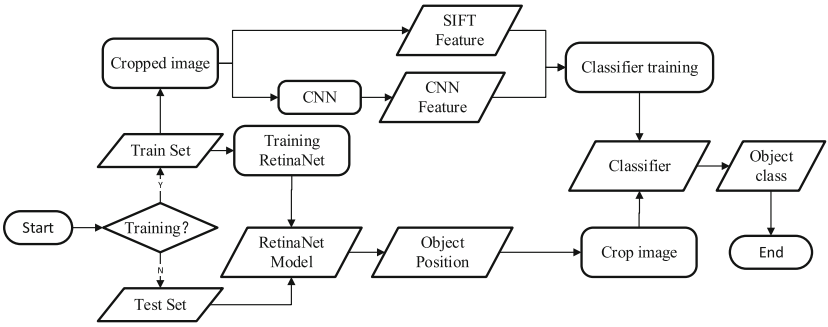

Fig. 1. The overall process of the object detection framework.

3.1 RetinaNet for Object Location

The main network of RetinaNet uses ResNet and FPN to generate feature pyramid model. After that, two sub-networks are connected: Classification Subnet and Box Regression Subnet, which are used for classifying and location respectively. The last layer of two sub-networks has the same outer structure but does not share parameters. The RetinaNet network structure is shown in Fig. 2.

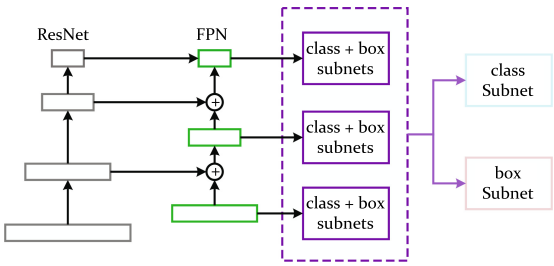

Fig. 2. Structure of RetinaNet.

Focal loss as a loss function to solve the problem that is class imbalance and distinguishing difficult instances:

$$FL(P_t) = -\alpha_t(1-p_t)^{\gamma}\log(p_t) \tag{1}$$

Where α is used to adjust the class imbalance, the range of value is [0, 1], $(1-p_t)^{\gamma}$ is the modulation factor, which can differentiate the easy example and hard example, γ is tunable focusing parameter and $\gamma \in [0, 5]$.

3.2 Fusion of SIFT Features and CNN Features to Classification

SIFT features have scale and illumination invariance. The extraction process takes grayscale images as input and scale space is used to detect image features

of different scales. According to the scale space theory, the scale space L of an image can be defined as:

$$L(x, y, \sigma) = G(x, y, \sigma) \times I(x, y) \tag{2}$$

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(x - x_i)^2 + (y - y_i)^2}{2\sigma^2}\right) \tag{3}$$

(x, y) is the spatial coordinate, σ is scale coordinate, and $G(x, y, \sigma)$ is a scale variable Gauss function. SIFT algorithm suggests that the feature detection on a certain scale can be obtained by subtracting the images in two adjacent Gaussian scale spaces to obtain the responsive image $D(x, y, \sigma)$. The local feature points are located in the spatial position and scale space by searching the local maximum value of the $D(x, y, \sigma)$. The mathematical description is:

$$G(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) \times I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma) \tag{4}$$

SIFT uses local extremum points in Gauss difference scale space as candidate feature points, determines the main direction of feature points to generate feature descriptors, and normalizes them to form SIFT feature vectors. Because the number of SIFT feature points extracted from each image is different, BOW algorithm is used to construct a uniform dimension feature vector for each image.

CNN neural network is usually used in image processing. It is composed of multiple convolution layers. Each convolution layer contains multiple convolution kernels. These convolution kernels are used to scan the entire image from left to right and from top to bottom to obtain output data called a feature map. This paper uses VGG19 to extract CNN features of the image. The subregion where the object is located is the input image. We use the output of the second complete acquisition layer as the CNN feature of the image, and input it with the SIFT feature into the classifier for training to build the classifier. The classification result is used as the target classification result of target detection.

4 Experimental Results and Analysis

The training set and test set used in this paper are the real-life pictures of Intelligent vending cabinet. We use RetinaNet's ResNet-101 model for training, and crop all object sub-images labeled in the training data set. Extract VGG19 features and SIFT features for fusion, and train the classifier.

The accuracy rate of RetinaNet is 97.93% where test set in the same environment as training set. This is an exciting result. The loss results in the training process of RetinaNet are shown in Fig. 3.

However, when we tested in different environments, we found that the performance of the model was not ideal. The following repeatability issues will occur:

1. Location failure. Sometimes has an object location failure that other items that have not appeared in the training set are displayed.
2. Misclassification. The object item is positioned accurately but the category is wrong.

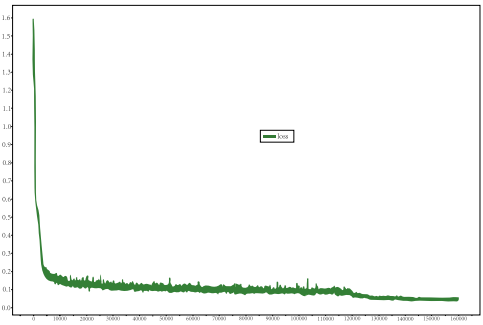

Fig. 3. Curve of loss.

It can be seen that changes in the external environment affect the accuracy of object detection. We also tried to expand the image with brightness, light, etc., and then trained with RetinaNet, but it did not achieve the desired results. Therefore, we try to train new classifiers by means of hybrid SIFT-CNN feature to improve detection accuracy. The output of RetinaNet is $[x_{min}, y_{min}, x_{max}, y_{max}, classes]$, x_{min}, y_{min} are the coordinates of the upper left corner of the object position, and x_{max}, y_{max} are the lower right corner, $classes$ is the confidence matrix, and the maximum value is the predicted class.

We will get the location of the object. It is used to crop the object, and extract the SIFT-CNN features of this area and input it into the classifier. For images whose SIFT features cannot be extracted, the category of the RetinaNet model will be the final category. Some areas with position errors input classifiers can obtain lower confidence, which can be filtered out by threshold. The classifier output can correct wrong label to a certain extent. The test results have been improved in changing scenarios, but there are still misclassifications. However, the speed has also declined. On TITAN XP, using RetinaNet alone, detecting an image takes an average of 180 ms, and our method will add about 23 ms to SIFT-CNN feature extraction and classification.

5 Conclusion and Future Work

In this study, we use RetinaNet to locate objects, then, the use CNN-SIFT hybrid feature of the detection area to predict the category. Although this corrects location errors and classification errors to a certain extent, improving the overall accuracy of object detection. It has not completely solved the accuracy degradation caused by external environmental change. In the future, we will study how to improve the robustness and accuracy of the model.

Acknowledgement. The research presented in this paper is supported in part by the National Natural Science Foundation (No.: 61571360), The Youth Innovation Team of Shaanxi Universities, the Innovation Project of Shaanxi Provincial Department of Education (No.: 17JF023) and the Project of Xi'an Technology Bureau (No.: GXYD14.12).

References

1. Wang, X.: Intelligent multi-camera video surveillance: a review. Pattern Recogn. Lett. **34**(1), 3–19 (2013)
2. Guo, L., Zhang, M., Wang, Y., et al.: Environmental perception of mobile robot, pp. 348–352. IEEE (2006)
3. Girshick, R., Donahue, J., Darrell, T., et al.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587. IEEE, Piscataway (2014)
4. Liu, W., et al.: SSD: single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
5. Redmon, J., Divvala, S., Girshick, R., et al.: You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788. IEEE, Piscataway (2016)
6. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7263–7271. IEEE, Piscataway (2017)
7. Redmon, J., Farhadi, A.: YOLOv3: an incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
8. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: YOLOv4: optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 (2020)
9. Li, P., Li, J., Huang, Z., et al.: Multi-key privacy-preserving deep learning in cloud computing. Future Gener. Comput. Syst. **74**, 76–85 (2017)
10. Blanco-Filgueira, B., García-Lesta, D., Fernández-Sanjurjo, M., et al.: Deep learning-based multiple object visual tracking on embedded system for IoT and mobile edge computing applications. IEEE Internet Things J. **6**(3), 5423–5431 (2019)
11. Girshick, R.: Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448. IEEE, Piscataway (2015)
12. Ren, S., He, K., Girshick, R., et al.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, pp. 91–99. The MIT Press, Cambridge (2015)
13. Sermanet, P., Eigen, D., Zhang, X., et al.: OverFeat: integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv:1312.6229 (2013)
14. Lin, T.Y., Goyal, P., Girshick, R., et al.: Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2980–2988. IEEE, Piscataway (2017)