Research on Traffic Information Detection of the Visually Impaired Based on Improved YOLOv3

Dongying Gao*, Tongyue Gao, Shaoshan Shi, ZhongChao Zhang, Zhao Ding
School of Mechatronics Engineering and Automation, Shanghai University, Shanghai, China

*Corresponding author: dawngdy@shu.edu.cn

Abstract. Among the target detection algorithms, YOLOv3 algorithm has fast detection speed and high accuracy, but it is difficult to directly deploy to small embedded devices because of its high requirements for the computing power of the device. In response to this problem, this paper combines the characteristics of EfficientNet-lite network and YOLOv3, and proposes an improved model of YOLOv3 combined with EfficientNet-lite network. This model takes advantage of the small size and high efficiency of the EfficientNet-lite network to reduce the size of the model, so that it can be applied to wearable devices to help blind people traveling to detect environmental information. Experimental results show that the model greatly reduces its size and its dependence on equipment performance under the premise of a small decrease in detection accuracy.

Keywords: Image detection, YOLOv3, EfficientNet, Ancillary equipment, Traffic sign detection.

1. Introduction

Eyes are an important organ for us to observe and perceive the world. According to statistics, worldwide, it is estimated that about 1.3 billion people have some form of visual impairment. In terms of distance vision, 188.5 million people have mild visual impairment, 217 million people have moderate to severe visual impairment, and 36 million people are blind [1]. In terms of myopia, 826 million people suffer from myopia [2]. Nowadays, there are many kinds of blind guide methods and equipment. Such as guide dogs, electronic guide dogs, guide sticks, guide glasses, and other wearable guide devices for the blind. The existing blind guide devices do not really remove the obstacles to the perception of the environment by the blind.

In recent years, deep learning has been rapidly developed and applied in various fields, such as image detection. Target detection based on deep learning can be divided into multi-stage processing to achieve target detection and single model to achieve target detection. Among them, the multi-stage target detection algorithm has RCNN (Regions with CNN features), but the training process and prediction process of the RCNN model are very complicated. Single model for target detection include SSD (Single Shot multibox Detector), YOLO, etc. Through analysis, we can find that the performance of the model is very much related to the depth of the model. The depth of the network model has expanded from a dozen layers to hundreds of layers. The improvement of model accuracy and the
expansion of depth also means the increase of model parameters. For a mobile device, the structure of these networks is too large, and ordinary mobile devices cannot meet its computing requirements. In order to adapt to mobile devices, many scholars trim the network model in depth and width to achieve the purpose of reducing the amount of calculation. For example, Yolo-tiny, a lightweight version of the YOLO (You Only Look Once) series [3][4][5]. However, some researchers use special structural designs for lightweight models, such as MobileNet[6][7], ShuffleNet[8] RFB (Respective Field Block), etc. By injecting more prior knowledge, the lightweight model can also achieve higher recognition accuracy. EfficientNet-lite network is an efficient and portable network. It uses a series of fixed scale scaling factors to scale the network dimensions. Through this special scaling method, the model is more efficient. This paper takes advantage of the small size and high efficiency of EfficientNet-lite network, and proposes an improved model based on YOLOv3.

2. Related work
In this paper, the improved algorithm model of YOLOv3 combined with the EfficientNet-lite network is used to realize the recognition of the surrounding environment and realize the purpose of broadcasting the surrounding road condition information for the blind. Figure 1 is a system sequence diagram that describes the detection process.

![System sequence diagram](image)

**Figure 1.** System sequence diagram.

2.1. YOLO model basic framework
YOLO's design concept follows an end-to-end training method and real-time detection. YOLO divides the input image into SxS grids. If the center of an object falls within a certain grid cell, the grid cell is responsible for detecting the object. In the YOLO algorithm, each grid cell predicts B target frames, and each target frame contains 5 parameters. There are K kinds of targets to be tested, so the algorithm finally outputs a prediction result vector of length SxSxBx5+K. The five parameters are the center coordinates (x, y) of the predicted target bounding box, the width w and the height h, and a confidence value [9]. The flow chart of the YOLO model is shown in Figure 2.
2.2. Loss function

The loss function of YOLOv3 is mainly composed of three parts, and the formula of the loss function is as follows:

\[
L(\sigma, O, c, t, i, g) = \lambda_1 L_{\text{conf}}(\sigma, c) + \lambda_2 L_{\text{loc}}(i, g) + \lambda_3 L_{\text{loc}}(i, g)
\]  

(1)

Where, \( \lambda_1, \lambda_2, \lambda_3 \) are balance coefficients.

2.2.1. The target positioning offset loss.

\[
L_{\text{loc}}(i, g) = \frac{\sum \text{d} x \sum \text{d} y \sum \text{d} w \sum \text{d} h (i - g)^2}{N_{\text{POS}}}
\]

(2)
Where, the predicted value is: $t_x, t_y, t_w, t_h$, and the predicted value $t_x, t_y$ is activated by the Sigmoid function.

$$\begin{align*}
\hat{i}_i^x &= \text{Sigmoid}(i_i^x) \\
\hat{i}_i^y &= \text{Sigmoid}(i_i^y)
\end{align*}$$

(3)

$$\begin{align*}
\hat{i}_i^w &= t_w \\
\hat{i}_i^h &= t_h
\end{align*}$$

(4)

Where, $\hat{g}_i^x, \hat{g}_i^y$ are the coordinates of the ground truth mapping on the preset feature layer, minus the coordinate of the upper left corner of the cell where it is located, is its offset. $\hat{g}_i^w$ is the offset of the ground truth box relative to the matching anchor it takes. $N_{POS}$ is the number of positive samples.

2.2.2. The confidence loss.  

The target score predicted by each bounding box uses logistic regression. Logistic regression generally uses binary cross-entropy loss,

$$L_{\text{conf}}(o, c) = -\frac{1}{N} \sum_{i=1}^{N} \left[ o_i \ln(\hat{c}_i) + (1-o_i) \ln(1-\hat{c}_i) \right]$$

(7)

$$\hat{c}_i = \text{Sigmoid}(c_i), \quad o_i \in (0, 1)$$

(8)

Where, $c$ is the predicted value, $\hat{c}_i$ is the prediction confidence of $c$ obtained by the Sigmoid function, and $N$ is the number of positive and negative samples.

2.2.3. The target classification loss.

$$L_{\text{cls}}(o, c) = \frac{\sum_{i=1}^{N_{POS}} o_i \ln(\hat{c}_{ij}) + (1-o_i) \ln(1-\hat{c}_{ij})}{N_{POS}}, \quad o_{ij} \in \{0, 1\}$$

(9)

$$\hat{C}_{ij} = \text{Sigmoid}(C_{ij})$$

(10)

Where, $C_{ij}$ is the predicted value, $\hat{C}_{ij}$ is the target probability obtained by $C_{ij}$ through the Sigmoid function, and $N_{POS}$ is the number of positive samples.

2.3. Bounding box prediction

In the early stage of training, its position prediction is unstable. The position prediction formula is:
\[
\begin{aligned}
  x &= (t_x \times w_o) + x_o \\
  y &= (t_y \times h_o) + y_o 
\end{aligned}
\] (11)

Where, \( x \) and \( y \) are the center of the prediction frame. \( x_o \) and \( y_o \) are the coordinates of the center point of the anchor. \( w_o \) and \( h_o \) are the width and height of the a priori box. \( t_x \) and \( t_y \) are the parameters to be learned. The width and height of the bounding box are used as the displacement of the cluster center, and the Sigmoid function is used to predict the center coordinates of the bounding box relative to the filter application position.

2.4. EfficientNet network structure analysis

In deep learning research, sometimes it is necessary to expand the network. The method of expanding the network is usually to change the size of the input image, deepen the depth of the network, or widen the width of the network. A lot of model expansion studies are to improve one aspect of the network or to combine the two aspects. EfficientNet is a network that uses compound model expansion ideas. It uses a series of fixed scale scaling factors to uniformly scale the network dimensions. Moreover, it is a lightweight neural network. While ensuring a certain image classification effect, it is lighter than the MobileNet neural network [10].

3. Experiment

This experiment uses Chinese Traffic Sign Detection Benchmark (CSUST) and DFG Traffic Sign Data Set. Table 1 shows some parameters of the two datasets.

| Dataset     | Category | Training set | Testing set |
|-------------|----------|--------------|-------------|
| CCTSDB      | 3        | 14625        | 1099        |
| DFGTSD      | 20       | 5254         | 1703        |

3.1. CSUST dataset

The CSUST dataset was produced by the team of Zhang Jianming from Changsha University of Science and Technology. The data set contains a total of 15,734 images, of which 15,724 are labeled images, and the training set contains 14,625. There are three types of marked data: indication signs, prohibition signs, and warning signs [11].

3.2. DFG Traffic Sign Data Set

The DFG Traffic Sign Data Set contains 200 traffic sign categories, each category has at least 20 instances, including 6758 images with a resolution of 1920x1080 and 199 images with a resolution of 720x576[12].

![Types of traffic signs in DFGTSD dataset.](image)
3.3. Analysis of results
In the experiment, the PASCAL VOC standard was used to evaluate the model, that is, to calculate the mAP when the IoU of the predicted frame and the real frame is equal to 0.5. Table 2 shows the detection accuracy results of the four models on the VOC 2007 testing set. It can be seen from the experimental results that the algorithm model in this paper is more lightweight. The model size is 3.1MB, which is much smaller than Tiny YOLOv2 and Tiny YOLOv3, and it is also reduced by 0.9MB compared to YOLO Nano. Compared with YOLO Nano, this model has reduced computational cost, but its mAP value has also decreased. Compared with Tiny YOLOv3, the detection accuracy of this model is increased by 2.7%, and compared with YOLO Nano, it is reduced by 8%. Therefore, compared with Tiny YOLOv3, the detection accuracy of the algorithm proposed in this paper has been improved. Compared with YOLO Nano, it has not achieved the expected effect, and there is still room for improvement.

| Method               | Model Size | FLOPS    | mAP (VOC 2007) |
|----------------------|------------|----------|----------------|
| Tiny YOLOv2[4]       | 60.5MB     | 6.97BFlops | 57.1%          |
| Tiny YOLOv3[5]       | 33.4MB     | 5.52BFlops | 58.4%          |
| YOLO Nano[13]        | 4.0MB      | 4.51BFlops | 69.1%          |
| Method of this paper | 3.1MB      | 4.16BFlops | 61.1%          |

Table 3 shows the comparison of mAP values between YOLOv3, Tiny YOLOv3 and the method in this paper on the CCTSDB and DFGTSD datasets. It can be seen from the experimental results that the method in this paper is more lightweight due to the introduction of the EfficientNet-lite network, which is more conducive to the application of small embedded devices. Compared with Tiny YOLOv3, the mAP on the dataset CCTSDB decreased by 21.8%, and the mAP on the dataset DFGTSD decreased by 10.8%. Therefore, there is a certain gap between the method in this paper and Tiny YOLOv3 in the mAP index, and the detection accuracy has not reached the expected effect, and there is a lot of room for improvement.

| dataset   | Method          | mAP  |
|-----------|-----------------|------|
| CCTSDB    | YOLOv3          | 92.9%|
|           | Tiny YOLOv3     | 82.1%|
|           | Method of this paper | 60.3%|
| DFGTSD    | YOLOv3          | 90.3%|
|           | Tiny YOLOv3     | 66.7%|
|           | Method of this paper | 55.9%|

4. Conclusion
This paper proposes an improved YOLOv3 model combined with the EfficientNet-lite network. The model reduces the size of the model by optimizing parameters, so that the model size is reduced to 3.1MB, which is 0.8MB lower than YOLO Nano. In the VOC 2007 test set, the detection accuracy of the improved model is higher than that of Tiny YOLOv3.

The model proposed in this paper requires less processing performance of the device, which is more conducive to deploying the model in wearable devices for the blind. The size of the model is reduced, resulting in a decrease in its detection accuracy. On the CCTSDB and DFGTSD datasets, the detection accuracy of this model is lower than that of Tiny YOLOv3. Therefore, we will continue to research and optimize the model to minimize the loss of detection accuracy caused by the reduction of model size.
References

[1] R.R.A. Bourne, S.R. Flaxman, T. Braithwaite, M.V. Cicinelli, A. Das, J.B. Jonas, et al. Magnitude, temporal trends, and projections of the global prevalence of blindness and distance and near vision impairment: a systematic review and meta-analysis. Lancet Glob Health, 5 (2017) 888–897.

[2] T.R. Fricke, N. Tahhan, S. Resnikoff, E. Papas, A. Burnett, M.H. Suit, T. Naduvilath, K.S. Naidoo, Global prevalence of presbyopia and vision Impairment from uncorrected presbyopia: systematic review, meta-analysis, and modelling, Ophthalmology, 125 (2018) 1492–1499.

[3] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, IEEE Conference on computer vision and pattern recognition (CVPR), 2016, pp. 779–788.

[4] J. Redmon, A. Farhadi, YOLO9000: better, faster, stronger, IEEE Conference on computer vision and pattern recognition (CVPR), 2017, pp. 7263–7271.

[5] J. Redmon, A. Farhadi, Yolov3: An incremental improvement, arXiv preprint arXiv:1804.02767, 2018.

[6] A.G. Howard, M.L. Zhu, B. Chen, D. Kalenichenko, W.J. Wang, T. Weyand, M. Andreetto, H. Adam, G. Inc, Mobilenets: Efficient convolutional neural networks for mobile vision applications, arXiv preprint arXiv:1704.04861, 2017.

[7] M. Sandler, A. Howard, M.L. Zhu, A. Zhmoginov, L.-C. Chen, Mobilenetv2: Inverted residuals and linear bottlenecks, arXiv preprint arXiv:1801.04381, 2019.

[8] X.Y. Zhang, X.Y. Zhou, M.X. Lin, J. Sun, M. Inc, ShuffleNet: An extremely efficient convolutional neural network for mobile devices, arXiv preprint arXiv:1707.01083, 2017.

[9] R. Shi, T.X. Li, Y. Yamaguchi, An attribution-based pruning method for real-time mango detection with YOLO network. Computers and Electronics in Agriculture, 169 (2020) 105214.

[10] M.X. Tan, Q.V. Le, EfficientNet: rethinking model scaling for convolutional neural networks, arXiv preprint arXiv:1905.11946, 2020.

[11] J.M. Zhang, Z.P. Xie, J. Sun, X. Zou, J. Wang, A cascaded R-CNN with multiscale attention and imbalanced samples for traffic sign detection, IEEE Access, 8 (2020) 29742–29754.

[12] D. Tabernik, D. Skočaj, Deep Learning for Large-Scale Traffic-Sign Detection and Recognition, arXiv preprint arXiv:1904.00649, 2019.

[13] A. Wong, M. Fauquier, M.J. Shafiee, F. Li, B. Chwyl, J. Chung, YOLO nano: a highly compact you only look once convolutional neural network for object detection, arXiv preprint arXiv:1910.01271, 2019.