EasyChair Preprint

№ 1171

Human body tracking method based on deep learning object detection

Zhifeng Yuan

June 12, 2019

Human Body Tracking Method Based on Deep Learning Object Detection

Yuan Zhifeng
College of Computer Science and Engineering
Northwest Normal University
Lanzhou.China
418910072@qq.com

ABSTRACT

Aiming at the problem of poor robustness of human detector based on artificial extraction feature, This paper proposes a visual tracking method based on deep learning object detection, which draws on the research results of target detection .The method utilizes the advantage of deep learning in feature representation, and uses the regression-based depth detection model YOLO to extract candidate targets. We re-clustered the data set for human targets, which improved the network performance of YOLO. For the extracted candidate frame position, the region is clipped. The HOG features of the candidate regions are extracted for target screening to achieve target tracking. Compared with pedestrian detection methods such as KCF and so on, this method reduces the miss detection rate and false detection rate, improves the robustness of the algorithm, and the detection speed meets the real-time requirements.

CCS CONCEPTS

Computing methodologies → Tracking.

KEYWORDS

YOLO, Target detection, Convolutional neural network, Target tracking

1 Introduction

Target tracking is an important research direction and research hotspot in the field of computer vision, and is widely used in military and life. Its research results are applied to precision guided weapons, unmanned aerial reconnaissance surveillance, human-computer interaction, video surveillance of pedestrians and vehicles[1-3].At this stage, the target tracking technology has made great progress, but there are still the possibility of tracking failure when faced with problems such as target appearance change, illumination change, occlusion, and similar targets. In general, tracking methods can be divided into two categories: production tracking and discriminant tracking [4]. The production tracking method needs to establish the prior model of the target first, and then searches the candidate region for the region that best matches the prior model and the smallest reconstruction error as the location of the target in the next frame. Typical examples are based on sparse coding [5], principal component method [6], dictionary learning [7] and other methods. The discriminant tracking method converts the tracking problem into a classification problem, separates the tracking target from the background, makes full use of the foreground and background information, and can better distinguish the two, so it has strong robustness. Common methods are models based on support vector machines [8], models based on naive Bayes [9].

In this paper, the deep convolutional neural network is used for candidate target extraction, and the extracted candidate targets are selected by using the fusion features of the target region's color histogram and localized HOG features to achieve tracking. This method takes advantage of the powerful advantages of deep learning in object detection and improves the robustness of the method.

Deep convolutional neural networks can autonomously learn the deep characteristics of the identified targets and refine their models. At present, the widely used convolutional neural network target recognition and detection methods can be divided into two categories. The first one is based on regional target recognition methods, such as Faster R-CNN [10], Mask R-CNN [11], etc. The method works well for the detection of small targets, but the detection speed is very slow; the other is the target recognition method based on regression, such as SDD [12], YOLO [13], etc. It uses an end-to-end target detection and identification method, the speed is much faster than the area-based target recognition method, and can basically meet the requirements of real-time.

2 Target Detection

The YOLO (You Only Look Once) algorithm is a regression-based target recognition method proposed by Redmon [13] in 2016, and has grown to the third generation by 2018. When detecting the target, the YOLO network only needs to perform a forward operation to complete the candidate frame selection, feature extraction, target classification, and target positioning multiple tasks, so the YOLO series algorithm detects quickly. The YOLO network can extract candidate regions directly from the image and predict the target location and classification probability through global image features. Turning the target detection problem into a regression problem enables end-to-end detection.

YOLO V3 still maintains the fast detection speed of the first two generations of YOLO, and at the same time greatly improves the accuracy of recognition, especially in the detection and recognition of small targets, the accuracy rate is greatly improved. YOLO V3 draws on the idea of residual neural network, introduces multiple residual network modules, and uses multi-scale prediction to improve the defects of YOLO for small target recognition, because it has high detection accuracy and good timeliness. This algorithm is one of the best algorithms for target detection.

2.1 Candidate Frame Selection

YOLO introduces the use of anchor boxes [14] in Faster R-CNN, which uses three different scales for prediction. Each scale has three anchor boxes, and the large scale features a small a priori box. Select the scale and the anchor box based on the target you want to identify, and modify the network structure using your own prepared prediction scale. The size of the candidate boxes is determined by clustering the size of the anchor boxes using the K-means clustering algorithm on the data set. The K-means algorithm usually uses Euclidean distance, Manhattan distance, Chebyshev distance to calculate the distance between two points. Using Euclidean distance will make large bounding boxes produce more errors than small bounding boxes. In order to improve the intersection of the candidate bound and the ground truth bound, we use the IOU to measure the clustering results.So the distance formula used in this article is:

$$d\left(box,\ centroid\right)=1-IOU\left(box,\ centroid\right)$$

In the formula: centroid represents the center of the cluster, box represents the sample, and IOU (box, centroid) represents the Intersection-over-Union of the cluster center box and the cluster box. IOU(Intersection-over-Union) indicates the accuracy of the prediction box. The formula is:

$$IOU(bb_{gt},bb_{dt})=\frac{bb_{gt}\bigcap bb_{dt}}{bb_{gt}\bigcup bb_{dt}}$$

Where bb_{gt} represents the real box and bb_{gt} represents the prediction box.

In this paper, we select K=9 and perform k-means cluster analysis on the training set respectively. Finally, the value of IOU gradually becomes gentle after 67.2.The size of the corresponding prediction box is set to the center of 9 clusters, and the results are (22,57), (43,109), (57,213), (99,276), (147,325), (177,205), (224,343), (349,363).

2.2 Network Structure

The YOLO V3's underlying network uses a number of well-formed 3*3 and 1*1 convolutional layers, and some residual network structures are used in subsequent multi-scale predictions. The final base network has 53 convolutional layers, so the author also calls them Darknet-53. Its structure is shown in Figure 1:

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Figure 1. Darknet-53

In neural networks, shallower convolutional layers can well characterize small-sized targets, while deeper convolutional layers can better describe large-scale targets. Therefore, according to the target size, different convolution layer features are selected to obtain more semantic information and predict the target [15]. Up sampling on a small scale can help the network learn more subtle features, and combine the output of the upper convolution layer to predict, which can better adapt to different size targets. Three different scales are used to predict the target in the YOLO V3 network. The process is shown in Figure 2:

1) Add some convolutional layers for prediction after the basic network, and the output feature map size is 13×13×1024. Scale 1 has good predictive power for large size targets.

2) The result of up sampling the convolutional layer of the penultimate layer in the scale 1 is added to the last feature map of size 26×26×512, and the prediction information is output again after multiple convolutions, and the feature map size is 26 × 26 × 768. Scale 2 uses a larger feature map for better prediction of smaller size targets.

3) The result of up sampling the convolutional layer of the second to last layer of scale 2 is added to the last feature image of size 52 x 52 x 256, and a feature map of 52 x 52 x 384 is output. Scale 3 has better predictive power for small size targets.

4) The YOLO layer is used for prediction for the three-scale feature map. The YOLO layer acts like a full join and consists of a convolutional layer of 3×3 and 1×1 convolution kernel sizes, enabling interaction between different feature map vectors. Finally, the output channels of the three scales are unified into a 75-dimensional output, and then classification and positional regression are performed here.

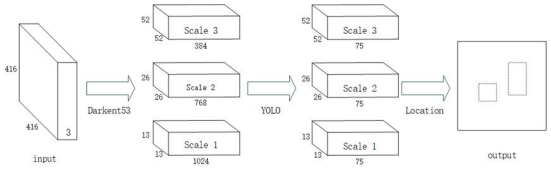


Figure 2. Multi-scale Prediction

3 Fusion Feature Extraction

After the target area is extracted using the YOLO neural network, features need to be extracted for the target area. In view of the different effects of light on imaging, this paper transforms the image from RGB space to HSV spatial image and calculates the color characteristic distribution to form the feature vector. The HSV spatial performance senses the saturation and brightness of the image, and the luminance component and the hue component are independent of each other and do not affect each other. Therefore, when the HSV spatial image is used, the correlation of the color can be removed, and the influence of the illumination on the image of the front and rear video frames is eliminated to some extent, and the illumination change is robust.

3.1 Network Structure

First, we perform HSV spatial transformation on the current frame image, analyze the contrast image component values, and finally detect the shadow interference caused by the illumination change through the formula and the set threshold. The interference is removed. Generally, the S and H values of the shadow and the background are not much different, but the V value changes significantly.

Converting the image to the HSV color space can eliminate the adverse effects of illumination and shadow on the target detection, and can reduce the initialization time and enhance the noise immunity of the tracking algorithm. The target area before the HSV space is converted is extracted from the candidate frame identified by the neural network. The candidate frame of the neural network is calculated by the intersection ratio when the position is returned, so there is a certain error in the candidate frame. In order to reduce the impact of the edge, a cosine window [16] is added before the feature is extracted. The specific calculation formula is as follows:

$$x_{ij} = (x_{ij}^{raw} - 0.5)\sin(\pi i / n)\sin(\pi j / n) \quad \forall i, j = 0, \ldots, n-1$$

3.2 Feature Calculation

Histogram of Oriented Gradient (HOG) is often used as a feature descriptor for object detection, which is used more in computer vision and image processing. The HOG feature characterizes the local appearance and shape of the image into the gradient magnitude and gradient direction of the pixel (x, y) by the gradient of the local image. The calculation formula is as follows:

$$G(x,y) = \sqrt{(H(x+1,y) - H(x-1,y))^2 + (H(x,y+1) - H(x,y-1))^2}$$

$$\alpha(x,y) = \tan^{-1}\left(\frac{H(x,y+1) - H(x,y-1)}{H(x+1,y) - H(x-1,y)}\right)$$

Figure 3(a) below is a test photograph in the INRIA Person data set, and Figure 3(b) on the right is the extracted HOG feature characteristic detection map.

(a) (b)

Figure 3. HOG Characteristic Detection Map of Candidate Regions

For the extracted HOG feature vector, in order to enhance its ability to express human features, each block is normalized using Euclidean distance. Its normalization formula is as follows:

$$v' = \frac{\mathrm{v}}{\sqrt{v_2^2 + \varepsilon^2}}$$

In the formula: 하 is a constant.

After obtaining the feature vector, the different candidate frames identified in the preceding and succeeding video frames are paired by calculating the cosine similarity of the vector. The specific calculation formula is as follows:

$$\text{similaity} = \cos(\theta) = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

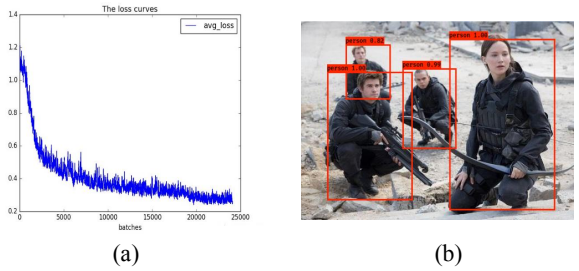
3 Experimental Results and Analysis

The experimental environment of this paper is: Intel i7-6700K, 16 GB of memory, Nvidia Geforce GTX1070，Ubuntu16.04 X64 operating system. In this experimental environment, the neural network detection speed reached 42 frame/s, meeting the real-time requirements.

The pedestrian detection database used in this paper is a mixed data set of PASCAL VOC and INRIA. The INRIA data set contains 614 positive sample images and 1237 pedestrians. The test data set contains 288 positive sample images and 589 pedestrians. Considering that there are fewer training samples in the INRIA dataset, in order to improve the generalization ability of the network, this paper extracts 6000 images of pedestrians in the PASCAL VOC dataset, and integrates with the INRIA dataset to expand the data volume in the dataset.

The YOLO V3 neural network was built using the Darknet network. The network training parameters were as follows:

batch=1, subdivision=1, the training image was reset to 416×416, the initial learning rate was 0.1, and it was set to 0.001 after 1000 iterations. The training loss is shown in Figure 4a. After training to 100 million times, the model area is stable. Figure 4b shows the recognition effect of the network.



(a) (b)

Figure 4. Training Results

We compares the performance of the YOLO V3 network with the classic R-CNN series and SSD series network from the aspects of Mean Average Precision(mAP) and detection frame number (FPS). The result is shown in Table 1. The table compares mAP metrics, FPS frames, and Batch Size batch sizes for different networks.

Table 1. Training Results

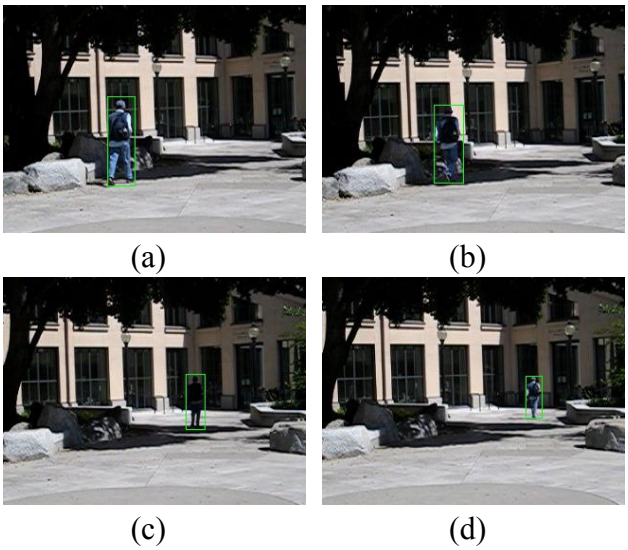
	mAP	FPS	Batch size
Fast R-CNN	70.0	0.5	1
Faster R-CNN	73.2	7	1
SSD300	74.3	46	1
SSD513	76.8	22	8
YOLO V3	80.0	54	8

We evaluated the algorithm using the OPE (One-Pass Evaluation) for human targets in the OTB dataset. We have implemented the YOLO V3 network and classic target tracking algorithm and the neural network target tracking algorithm based on R-CNN series and SSD series, And the result is shown in Table 2.

Table 2. Test comparison of algorithm on tracking sequence

	OPE_PRECISION	OPE_SUCCESS
OUR	0.732	0.716
CREST[16]	0.610	0.612
KCF[17]	0.516	0.495
ASLA[18]	0.470	0.410
FRAG[19]	0.309	0.227

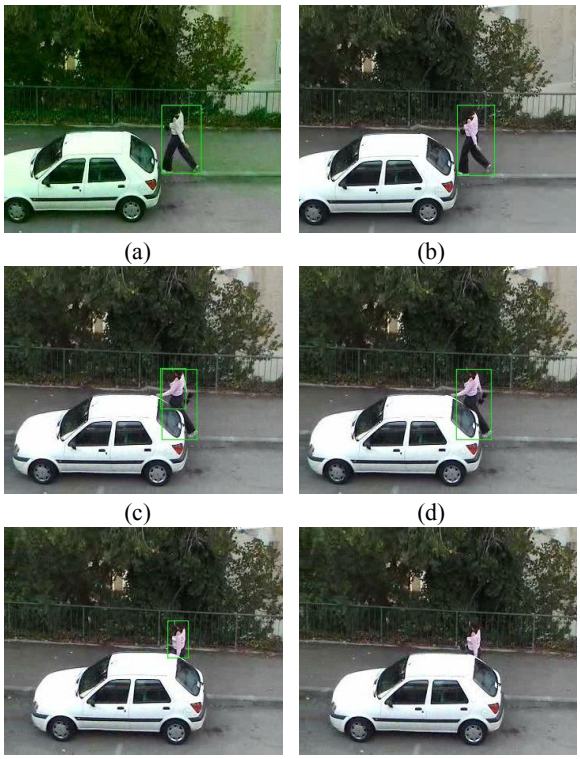
We selected two test datasets in OTB-100 to show the results of object tracking. In Figure 5, we showed the result of the "Human8". In this dataset, the target has illumination changes, scale changes, deformations, and the target is in the shadow.



(a) (b) (c) (d)

Figure 5. Human8 Dataset Tracking Results

In Figure 6, we showed the result of the "Woman". In the Figure 6a and Figure 6b, get a better tracking results when the image is under color distortion. In the Figure 6c, the Yolo V3 detects the wrong target, and in the Figure 6d, the algorithm selects the highest confidence result by similarity. In the Figure 6f, we lost the target, but in the next video frame, wo re-identification the target. In Figure 6h to Figure 6m, the focal length of the camera begins to change. Our algorithm satisfactorily completes the task of tracking.



(a) (b) (c) (d)

Figure 6. Woman Dataset Tracking Results

4 Summary and Discussions

In this study, an autonomous method is proposed to tracking Human body; this method is applicable to monitor systems. In order to achieve this goal, we first need to detect the location of the target in the video. So, we compared the performance of different neural networks, and the YOLO is trained according to the human body target. Then, we design algorithm to target tracking. The final result shows the potential and possibility of the proposed method.

ACKNOWLEDGMENTS

This research was supported by the Northwest Normal University research grant in 2018.

REFERENCES

[1] Sivanantham S, Paul N N, Iyer R S. Object tracking algorithm implementation for security applications[J]. Far East Journal of Electronics and Communications, 2016, 16(1): 1-13.

[2] Kwak S, Cho M, Laptev I, et al. Unsupervised object discovery and tracking in video collections [C]//IEEE International Conference on Computer Vision, 2015: 3173-3181.

[3] Luo Haibo, Xu Lingyun, Hui Bin, et al. Status and prospect of target tracking based on deep learning [J]. Infrared and Laser Engineering, 2017, 46(5): 0502002. (in Chinese)

[4] GUAN Hao, XUE XiangYang, AN ZhiYong. Advances on Application of Deep Learning for Video Object Tracking. Acta Automatica Sinica, 2016, 42(6): 834-847

[5] Mei X, Ling H. Robust visual tracking using l1 minimization [C]//IEEE International Conference on Computer Vision, 2010: 1436-1443.

[6] Ross D A, Lim J, Lin R S, et al. Incremental learning for robust visual tracking[J]. International Journal of Computer Vision, 2008, 77(1-3): 125-141.

[7] Wang N, Wang J, Yeung D Y. Online robust non-negative dictionary learning for visual tracking [C]//IEEE International Conference on Computer Vision, 2013: 657-664.

[8] Avidan S. Support vector tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2004, 26(8): 1064-1072

[9] Zhang K H, Zhang L, Yang M H. Real-time compressive tracking. In: Proceedings of 12th European Conference on Computer Vision. Florence, Italy: Springer, 2012. 864-877

[10] Girshick R. Fast R-CNN [C]// Proc of IEEE International Conference on Computer Vision. 2015: 1440-1448.

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, et al. Mask R-CNN[C] //IEEE Conference on Computer Vision and Pattern Recognition. 2018.

[12] Liu W, Anguelov D, Erhan D, et al. SSD: single shot multibox detector [C]// Proc of European Conference on Computer Vision. Springer, 2016: 21-37.

[13] Redmon J, Divvala S, Girshick R, et al. You only look once: unified, real-time object detection[J]. 2015: 779-788.

[14] Dai J, Li Y, He K, et al. R-FCN: object detection via region-based fully convolutional networks [C]// Neural Information Processing Systems. 2016: 379-387.

[15] Liu Hui,Peng Li,Wen Jiwei. Multi-Scale Aware Pedestrian Detection Algorithm Based on Improved Full Convolutional Network [J]. Laser & Optoelectronics Progress,2018,55(09):318-324.

[16] Song Y, Ma C, Gong L, et al. Crest: Convolutional residual learning for visual tracking[C]//Proceedings of the IEEE International Conference on Computer Vision. 2017: 2555-2564.

[17] Henriques, João F., et al. "High-speed tracking with kernelized correlation filters." IEEE transactions on pattern analysis and machine intelligence 37.3 (2014): 583-596.

[18] Jia X, Lu H, Yang M H. Visual tracking via adaptive structural local sparse appearance model[C]//2012 IEEE Conference on computer vision and pattern recognition. IEEE, 2012: 1822-1829.

[19] Adam A, Rivlin E, Shimshoni I. Robust fragments-based tracking using the integral histogram[C]//2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06). IEEE, 2006, 1: 798-805.