2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC) | 978-1-7281-8028-1/20/$31.00 ©2021 IEEE | DOI: 10.1109/IAEAC50856.2021.9390613

A video streaming vehicle detection algorithm based on YOLOv4

Xizhi Hu[1], Zheng Wei[1], Wenchao Zhou[1]

1.School of Mechanical and Automotive Engineering, South China University of Technology, Guangzhou 510640, China huxizhi@scut.edu.cn, weizheng_602@163.com , zhouxiaoxi1994@163.com

***Abstract*—To achieve safe driving of vehicles, it is necessary to perceive information about the vehicle's surroundings, and computer vision is one of the key technologies to solve this problem. The YOLO series and SSD, RetinaNet algorithm are representative of one-stage target detection algorithms, which have high accuracy and high speed. YOLOv4 is the latest algorithm of YOLO series, which has improved the speed and accuracy of vehicle target detection than before, but there is still a distance from the real real-time in vehicle detection. This paper proposes an improved YOLOv4-based video stream vehicle target detection algorithm to solve the problem in the detection speed which is not fast enough. This paper first introduces the YOLOv4 algorithm theoretically, then proposes an algorithmic process to speed up the detection speed, and finally conducts practical road experiments. From the experimental results, the algorithm of this paper can improve the detection speed of the algorithm without losing accuracy, which can provide a basis for decision making for safe vehicle driving.**

Keywords—*YOLOv4; Video Streaming, Vehicle detection; algorithm fusion*

I. Introduction

With the rapid development of China's economy and productivity, China's per capita car ownership is on the rise. While cars bring us the convenience of daily life, they also bring us potential safety hazards. To reduce such safety hazards, it is necessary for vehicles to sense their surroundings and thus make corresponding responses to different environments. Computer vision technology is a key technology in the field of intelligent driving, which enables vehicles to sense other obstacles in front of the vehicle such as vehicles, pedestrians, etc. Through the fusion with other sensors such as millimeter wave radar, LIDAR and other data can make the vehicle well sense the surrounding environment, thus ensuring the safety of vehicle driving. In the current field of vehicle target detection, the mainstream detection methods are divided into traditional image processing-based detection methods and deep learning-based detection methods. Traditional image processing-based detection methods require manual design features and a series of feature extraction. This method is easy to understand and has fast computational speed, but the algorithm's generalization performance is poor, and the robustness is not as good as the deep learning-based methods; deep learning-based methods can achieve end-to-end training and detection because they mostly use convolutional neural networks for learning and feature extraction, which have good generalization, robustness, and detection effects. The target detection based on convolutional neural network is divided into one-stage target detection algorithm and two-stage target detection algorithm according to whether the candidate frame of the possible target needs to be extracted in advance, two-stage representative algorithms are R-CNN, fast R-CNN, faster R-CNN, R-FCN, etc. one-stage representative algorithms are YOLO series, SSD, RetinaNet, etc. Considering the requirements of real time and accuracy, YOLOv4 algorithm is chosen in this paper to achieve the detection of vehicle targets.

II. Related Theories

The YOLOv4 algorithm mainly includes three parts: Backbone(the backbone feature extraction network), Neck(the feature pyramid)and YOLO Head. When the resolution of the input image is 608*608, its network structure is shown in Figure 1. Backbone uses CSPDarkNet53 to extract the model features; Neck part includes SPP structure and PANet structure, which are the feature pyramid structure of the model; the role of YOLO Head is to use the features extracted from the previous part to obtain the final prediction results, including the class of the target, the Bbox(Bounding box) and the Confidence.

The CSPDarknet53 network is an improvement on the Darknet53 network. There are 53 convolutional layers in the Darknet53 network, and excluding the last fully connected layer, there are 52 convolutions in the main network. When the input image comes into the network, the image first passes through a convolution kernel with 32 filters, and then passes through 5 sets of residual units Residual_body (in these 5 residual units, each unit consists of a single convolutional layer with a set of repeatedly executed convolutional layers with repetitions of 1, 2, 8, 8, 4; in each repeatedly executed convolutional layer, first performs a 1×1 convolution operation and then a 3×3 convolution operation). A residual network is one in which the input of the previous convolutional layer without convolutional operation is fused with the data that has undergone convolutional operation before being used as the input of the next layer. This operation can solve the problem of gradient disappearance in deep learning, so that the network can continue to deepen and strengthen the feature extraction ability of the model. In the residual unit, the first individual convolution layer are all convolution operations with stride of 2. Therefore, the structure with input of (608,608,32) is obtained as (19,19,1024) after 5 residual units.

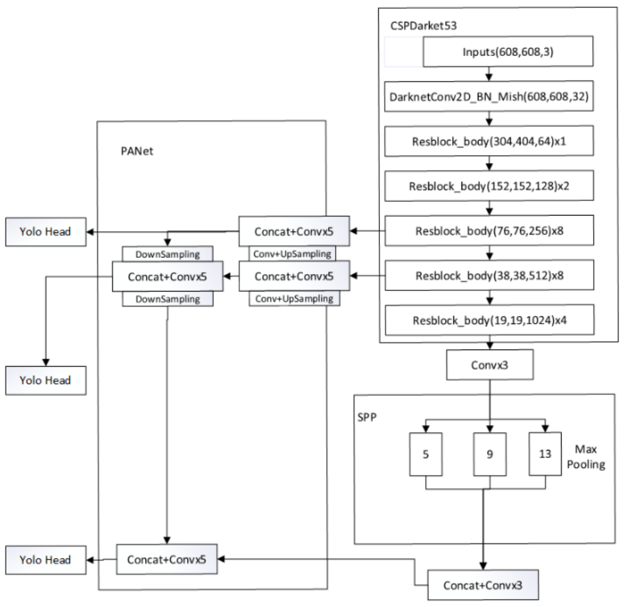

CSPDarket53
Inputs(608,608,3)
DarknetConv2D_BN_Mish(608,608,32)
Resblock_body(304,404,64)x1
Resblock_body(152,152,128)x2
Resblock_body(76,76,256)x8
Resblock_body(38,38,512)x8
Resblock_body(19,19,1024)x4
Convx3
SPP
5
9
13
Max Pooling
Concat+Convx3
PANet
Yolo Head
Concat+Convx5
DownSampling
Conv+UpSampling
Yolo Head
Yolo Head

Fig. 1. Network structure diagram of YOLOv4

CSPDarknet53 has made 2 main improvements on the basis of Darknet53. One is the use of the Mish activation function instead of the LeakyReLU function in the DarknetConv2D module, which is given as follows.

$$Mish = x \times \tanh(\ln(1 + e^{x})) \tag{1}$$

Its difference with the LeakyReLU activation function is shown in Figure 2

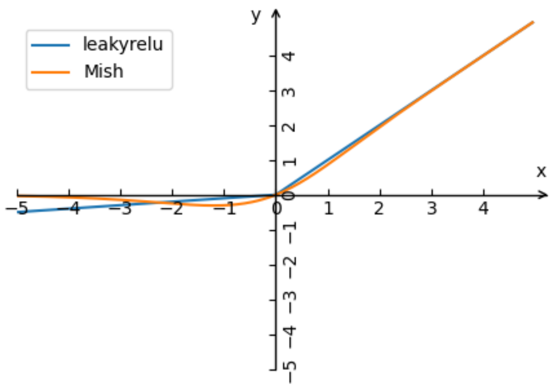


Fig. 2. Comparison of Mish with LeakyReLU

The advantage of the Mish function is that it is unbounded, avoids saturation due to capping, and allows a small negative gradient flow, thus ensuring information flow; the gradient of Mish is much smoother, allowing better information to penetrate deeper into the neural network. Secondly, CSPNet is used in the structure of Resblock_body, the structure of CSPNet is to split the original residual block into left and right parts: the main part continues the stack of the original residual block, and the other part is like a residual edge, which is directly connected to the end after a small amount of processing, so it can be considered that there is a large residual edge in CSPNet, and its structure diagram is shown in Figure 3 .

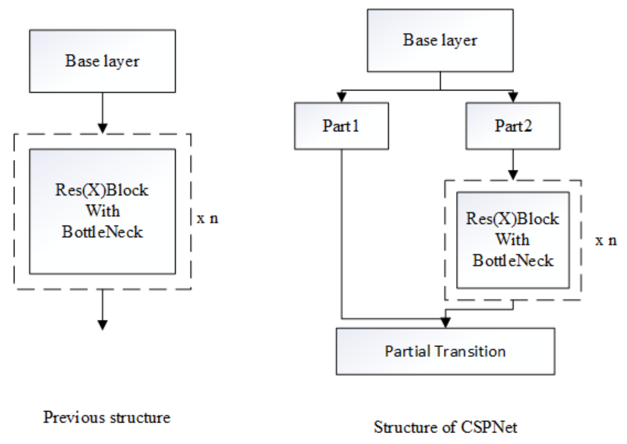


Fig. 3. Comparison of CSPNet structure with the previous structure

The SPP structure is sandwiched in the convolution of the last feature layer of CSPDarknet53.After three convolutions of the last feature layer of CSPDarknet53, the maximum pooling kernel sizes of 13×13,9×9, 5×5, and 1×1 are utilized for four different scales, respectively. SPP can effectively increase the perceptual field of the model and separate the most significant contextual features. The role of PANet structure is to iteratively extract the features, it means to complete the feature extraction from bottom to top and top to bottom in the feature pyramid, as shown in Figure 1, and this process is done by several up sampling and down sampling jobs.

In terms of feature utilization, YOLOv4 extracts a total of three feature layers, and the shapes of the three feature layers are (76,76,256), (38,38,512), and (19,19,1024) as shown in Figure 1, respectively. The feature layers form three output layers after PANet's splicing, convolution, upsampling, and downsampling, and their shapes are (76,76.), (38,38,75), (19,19,75), the last dimension is 75 because this graph is obtained based on the VOC dataset training, it contains 20 classes, the output also contains the center coordinates x, y of the prediction box as well as the height h, width w, confidence, the algorithm has 3 preselected boxes for each feature layer, so the dimension is 3 × 25. After getting the final after obtaining the final prediction structure, NMS (Non-Maximum Suppression) is performed to obtain the final detection results based on its score.

YOLOv4 also has improvements in training techniques with its mosaic data enhancement method, label smoothing, learning rate cosine annealing decay, and CIOU. These techniques help to improve the feature extraction ability of the model and the accuracy of the algorithm.

III. Algorithm Fusion Process

In this paper, the improved algorithm firstly hashes the difference value for each frame of video input to get the hash value, then extracts the fingerprint with the previous frame to calculate their hamming distance, and compares the hamming distance with the threshold value to determine the similarity between this frame and the previous frame. Finally, the algorithm is selected according to the similarity size, so it improves the average computation speed of the whole algorithm by this method.

A. Difference Hash Algorithm

The difference hash algorithm (DAH) is a hash algorithm mainly used for fast detection of image difference, and it has the advantages of both fast detection and good detection compared with the average hash algorithm and perceptual hash algorithm which are also used for detecting image difference. Its workflow is as follows.

1） Reduce the input image to a size of 9×8, hiding the details of the image, and the processed image has a total of 72 pixel points

2） Convert the reduced image into a 64-level grayscale map, and convert the original RGB three-channel three-dimensional comparison into a one-channel one-dimensional comparison

3） Calculate the difference value. The difference value is calculated by comparing each pixel in each row with the next pixel in the same row, if the intensity of the previous pixel is greater than the intensity of the next pixel, then the difference value is 1, otherwise it is 0. After processing, an 8x8 difference value matrix is obtained.

4） converted to dHash values. Each value in each row of the 8x8 matrix obtained in step 3 is viewed as 1 bit, and each 8 bits form a hexadecimal value, and finally a hash string of length 8. This string is the hash value we want.

The Hamming distance is the number of different bits on the corresponding bit between two strings and is calculated as follows.

$$d(x,y) = \sum x[i] \oplus y[i] \tag{2}$$

By calculating the dHash value of the current frame and the dHash value of the previous image for the calculation of Hamming distance, we can get the similarity of the two frames, generally take the threshold value of 5. That means when the Hamming distance is not greater than 5, we can consider that the two frames are similar to each other.

B. Camshift tracking algorithm

Camshift algorithm is an adaptive adjustment of Mean-Shift algorithm. It is characterized by the ability to adjust the size of its own search box and fast speed, its algorithmic idea is to do Mean-Shift algorithm processing of each frame of the video, and the results of the previous frame as the initial value of the current frame to iterate, so it can achieve the effect of tracking. Its algorithm flow chart is shown in Figure 4, and its specific flow is as follows.

1） Build the color histogram of H component based on the HSV space image converted from RGB space image, and then reverse project it to obtain the color probability density distribution。

2） Let the pixel （x,y） be located in the tracking frame, then $I(x,y)$ denotes the probability estimate corresponding to the pixel in the color probability estimation map. The zero-order moment M_{00} and the first-order moments M_{10},M_{01} of the tracking window are calculated as follows.

$$M_{00} = \sum\sum I(x,y) \tag{3}$$

$$M_{10} = \sum\sum xI(x,y) \tag{4}$$

$$M_{01} = \sum\sum yI(x,y) \tag{5}$$

3） Calculate the center position of the tracking frame as well as the height h and the length l：

$$x_c = \frac{M_{10}}{M_{00}}，\ y_c = \frac{M_{01}}{M_{00}} \tag{6}$$

$$h = \sqrt{\frac{M_{00}}{256}}, l = 1.2 \cdot h \tag{7}$$

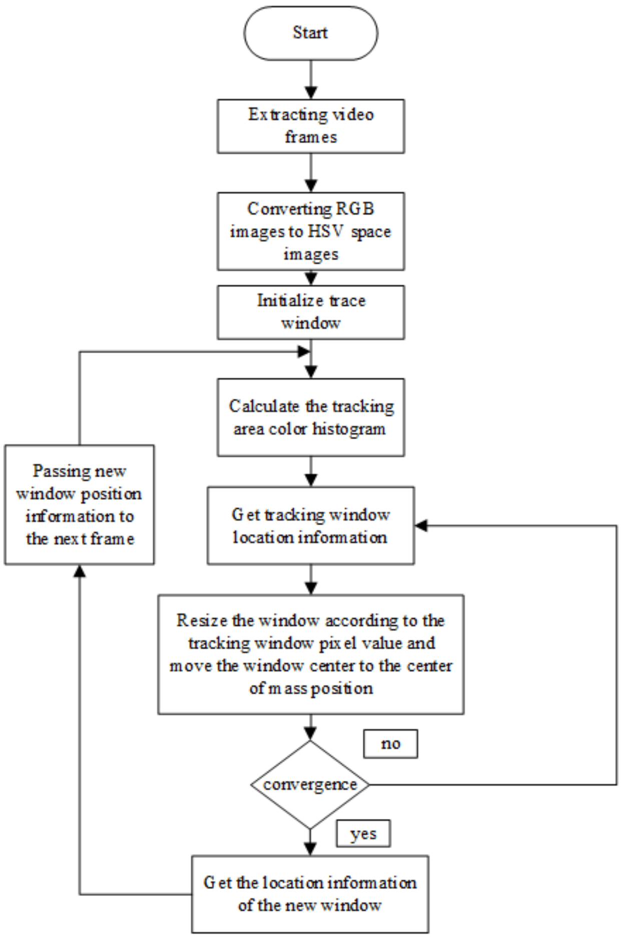

Start
Extracting video frames
Converting RGB images to HSV space images
Initialize trace window
Calculate the tracking area color histogram
Get tracking window location information
Resize the window according to the tracking window pixel value and move the window center to the center of mass position
convergence
no
yes
Get the location information of the new window
Passing new window position information to the next frame

Fig. 4. Camshift algorithm flow chart

C. YOLOv4 algorithm based on migration learning

Transfer learning is a kind of migration of trained model parameters to new models to help train new models. YOLOv4 is officially trained using the VOC dataset, this training set has 20 classifications, and the obtained model has strong feature extraction and generalization capabilities, so the weight file of YOLOv4 is used as the pre-training model for vehicle detection. In this paper, we focus on the recognition of front vehicles and pedestrians, and use 7132 images from the open-source dataset KITTI as the training set, with the labeled classification of car and person. In order to take full advantage of the YOLOv4 pre-trained model of the backbone feature extraction network, the backbone network model is frozen for the first 50 epochs to speed up the training and prevent the weights from being corrupted at the beginning of training. In this paper, the experimental platform CPU is i7-8750H with 2.21GHz, 16G RAM, NVIDIA GeForce GTX1060 with 6G video memory, Python version is 3.6, CUDA version is 10.0, CUDNN version is 7.6.5, Keras version is 2.1.5, Tensorflow-gpu version is 1.13.2, the training uses Adam optimizer optimization algorithm, the initial learning rate is 0.0001, the minimum learning rate is set to 0.000001, the batch size is 16 when frozen and 8 after thawing, the mosaic data enhancement method the and learning rate cosine annealing decays are used, and the training loss value changes as shown in Figure 5. Finally, the video stream vehicle detection is performed according to the model weight file obtained by migration learning.

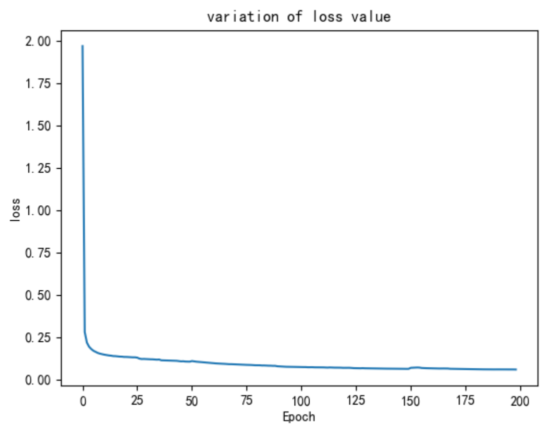


Fig. 5. Variation of loss during training

D. *Algorithm Fusion*

After obtaining the best weight file based on migration learning, this weight data is used to perform algorithmic fusion. YOLOv4 vehicle target detection is first performed and computed for the first frame passed into the algorithm, and then the detected tracking target and position information is passed to the Camshift algorithm. If the difference hash value between the current frame and the previous frame is large, the YOLOv4 algorithm is used for detection, otherwise, the Camshift algorithm is used for tracking, as shown in the flow chart in Figure 6.

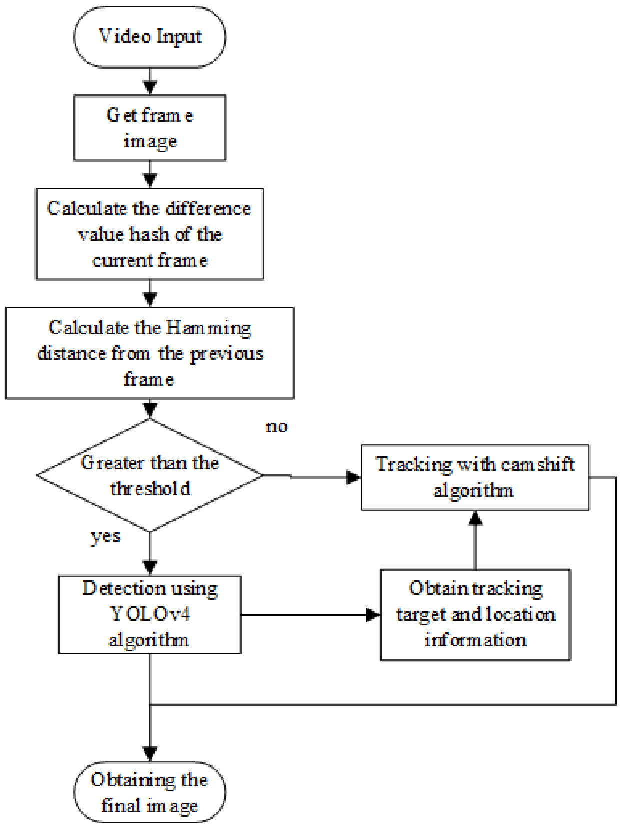
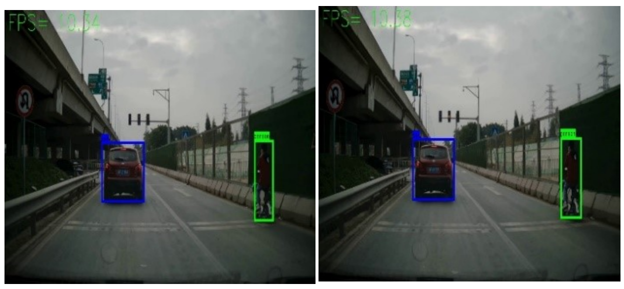

Video Input
Get frame image
Calculate the difference value hash of the current frame
Calculate the Hamming distance from the previous frame
Greater than the threshold
no
yes
Tracking with camshift algorithm
Detection using YOLOv4 algorithm
Obtain tracking target and location information
Obtaining the final image

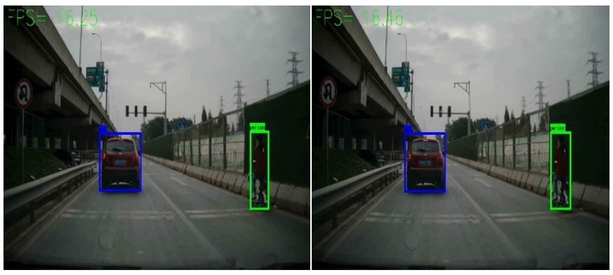
Fig. 6. Algorithm flow chart

IV. Real-world road experiments

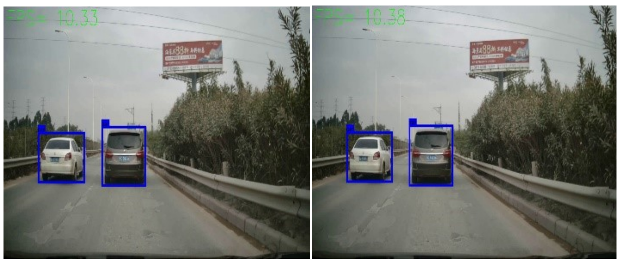
To verify the real-time and effectiveness of the vehicle detection algorithm in this paper, the effectiveness and real-time analysis of the algorithm is performed by fixing an industrial-grade U2291 USB camera at the central rearview mirror in the vehicle to capture video from a real vehicle on a city road in Guangzhou, and the real-time is mainly evaluated by FPS (Frames Per Second). The set score threshold is 0.5, this means detection target will be displayed if the score exceeds 0.5. The detection effect of the obtained continuous frames is shown in Figure 7.



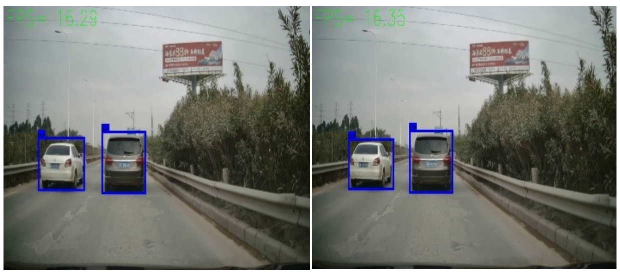
before algorithm fusion



after algorithm fusion



before algorithm fusion



after algorithm fusion

Fig. 7. Detection effect diagram

V. conclusion

From the results of the real-world road experiments, it can be seen that the improved YOLOv4 detection algorithm can effectively detect vehicles and pedestrians in front. The fused vehicle detection algorithm improves the overall detection speed from about 10 FPS to about 16 FPS, which can effectively improve the overall detection speed, and basically does not degrade the detection effect of the algorithm when tracking with the Camshift algorithm. Therefore, the algorithm designed in this paper has improved the detection speed of video streams.

REFERENCES

[1] GIRSHICK R. Fast R-CNN[C]//IEEE International Conference on Computer Vision. Santiago,2015.

[2] Jiwoong Choi, Dayoung Chun, Hyun Kim, and Hyuk-Jae Lee. Gaussian YOLOv3: An accurate and fast object detector using localization uncertainty for autonomous driving. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), pages 502–511, 2019.

[3] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-FCN:Object detection via region-based fully convolutional networks. In Advances in Neural Information Processing Systems (NIPS), pages 379–387, 2016.

[4] Cheng-Yang Fu, Mykhailo Shvets, and Alexander C Berg. RetinaMask: Learning to predict masks improves stateof-the-art single-shot detection for free. arXiv preprint arXiv:1901.03353, 2019

[5] Ross Girshick. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), pages 1440–1448, 2015.

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

[7] J. G. Allen and J. S. Jin. Mean shift object tracking for a SIMD computer. In ICITA '05: Proceedings of the Third International Conference on Information Technology and Applications (ICITA'05) Volume 2, pages 692–697, Washington, DC, USA, 2005. IEEE Computer Society.

[8] Xiu C. and Ba F. 2016 2016 Chinese Control and Decision Conference (CCDC) (Yinchuan) Target tracking based on the improved Camshift method 3600-3604

[9] Wei W., Zhijun L., Zhiping C., Xiaomeng M., Rui Z. and Ruo Y. 2017 2017 International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII) (Wuhan) The Application of Improved Camshift Algorithm in Hand Tracking 87-90

[10] YOLOv4: Optimal speed and accuracy of object detection. arXiv 2020 A Bochkovskiy, CY Wang, HYM Liao - arXiv preprint arXiv:2004.10934, 2020