Dangerous Driving Behavior Recognition Based on Improved YoloV5 and Openpose

N. Yang, J. Zhao

***Abstract*—This paper focuses on the automatic detection of two dangerous driving behaviors, smoking and talking on the phone. First, our own datasets are constructed for the two anomalous behaviors of smoking and making phone calls. Furthermore, a two-stage behavioral anomaly detection method is proposed. The object detection network model improves the prediction head of YoloV5, making it more effective in detecting small target objects, and optimizes the loss function for the categorically mutually exclusive datasets. The pose estimation network structure is improved, and the attention mechanism of the Coordinate Attention (CA) structure is introduced to improve the efficiency and accuracy of information processing. Finally, the end result is evaluated through Euclidean distance calculation, and the elbow joint angle is used as an auxiliary judgment condition. The proposed dangerous driving behavior recognition model achieves a mean average precision of 93.4% under the condition of about 61FPS and improves the detection accuracy by 8.2%. It meets real-time requirements and increases accuracy while maintaining speed.**

***Index Terms*—Behavioral anomaly, YoloV5, Openpose, Euclidean distance.**

I. Introduction

Behavior recognition of moving objects is a thriving area in many research fields, especially in monitoring and safety management [1]. In daily life, there are many uncertainties in human behaviors. Some people may exhibit abnormal physical behaviors due to alterations of the body (e.g., disease, ageing). In such events, the center of gravity and focus of the human body is variable or its symmetry is disrupted, and the body cannot hold original balance, resulting in falls [2]. In addition, people who violate relevant regulations can also be identified as having abnormal behaviors: driving while fatigued, talking on the phone, texting, engaging in passenger conversations, and being distracted while driving, which will cause dangerous driving behaviors [3]. Students' slapping, rolling, stair-sliding, and other behaviors in school and other fixed scenes, are typically relatively dangerous behaviors that fall within the category of abnormal detection targets.

Manuscript received May 12, 2022; revised October 27, 2022. This work was supported by the Natural Science Foundation of Liaoning Province, grant number 20180551048.

N. Yang is a postgraduate student of School of Computer Science and Software Engineering, University of Science and Technology LiaoNing, Anshan 114051, China (e-mail: y_nan98@163.com).

J. Zhao is a Professor of School of Computer Science and Software Engineering, University of Science and Technology LiaoNing, Anshan 114051, China (corresponding author, phone: 86-13998086167; e-mail: zhaoji@ustl.edu.cn).

Behavior recognition based on deep learning can be roughly divided into two types. The first is behavior recognition based on skeleton key points, and some of the most popular methods include ST-NBNN [4], MiCT [5] and ST-GCN [6]. The second is the behavior recognition of spatio-temporal features, which are represented by two-stream convolutional networks [7], P3D [8], and LRCN [9]. These methods are often computationally intensive and require strong correlations between successive frames. Extensive calculations will bring a great burden to the computer, and the relationship between consecutive frames places a heavy constraint on the dataset. Therefore, our work focuses on improving the detection accuracy with lower cost, constructing scenario-specific network models, and reducing the computational complexity through the newly developed model.

In conventional deep learning algorithms, object detection and pose estimation are independent of behavior recognition [10]. Some studies have proved that the combination of object detection and pose estimation can improve the effectiveness of behavior recognition under certain circumstances. Fang et al. [11] proposed RMPE pose estimation method in 2017; an object detection network is used in a first stage, and pose estimation is then used to extract skeleton key points location information of the human body faster and more accurately. Inspired by the staged pose estimation technique, we propose a method to detect the occurrence of abnormal behavior. First, it must be considered whether there are hazardous targets in the scene, such as mobile phones and cigarettes. Second, through the estimation of posture, the possibility of danger caused by human actions can be identified. With this staged approach, in some special scenarios like cars captured on a highway, the first stage can detect the presence or absence of distracting objects like cigarettes and mobile phones, which are potential factors for driver behavior abnormalities, while in the second stage, it is necessary to estimate the posture of the people in the scene; by estimating the driver's body posture, it can be determined whether the driver smokes or takes phone calls while driving the motor vehicle.

II. Related Work

A. YoloV5 Algorithm Principle

YoloV5 is an improved version of Yolov4 framework. It outperforms Yolov4 in terms of flexibility and speed, and offers significant advantages in model deployment. It provides four kinds of pretraining models for object detection, which are YoloV5s, YoloV5m, YoloV5l and YoloV5x. The pre-training models differ in that the depth and width of the network are successively deepened and widened.

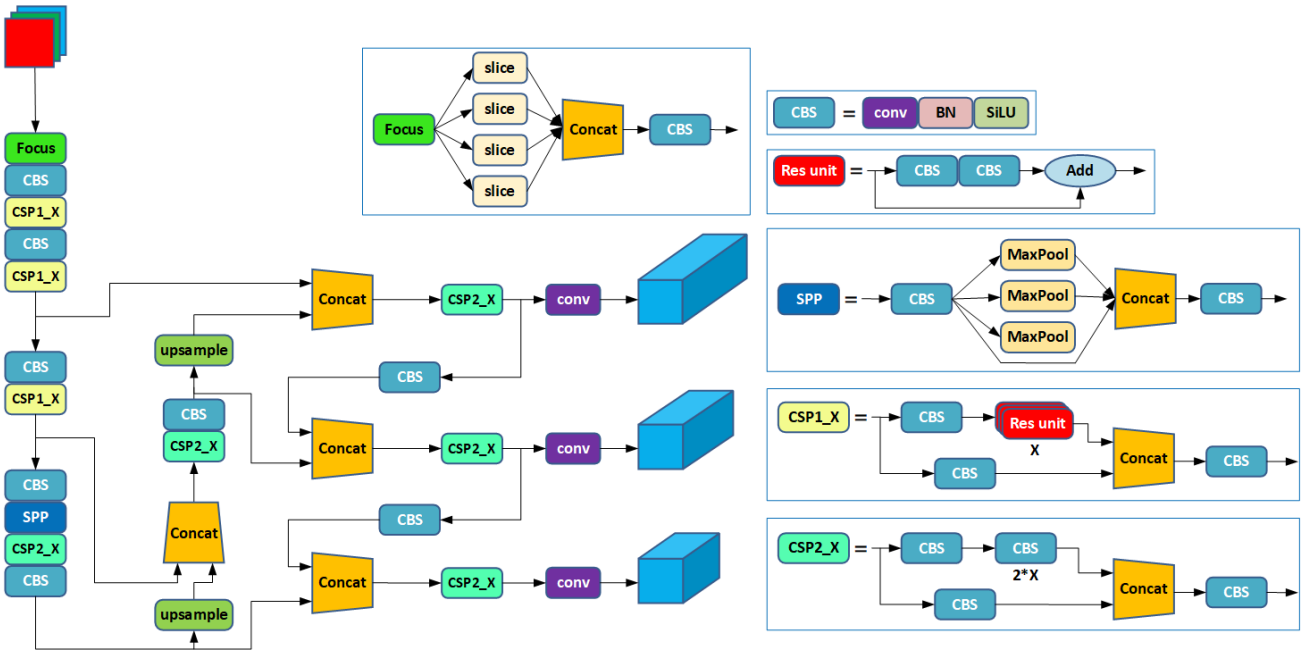


Fig. 1. YoloV5 network architecture.

Compared with YoloV4 [12], YoloV5 makes further innovations on it while still retaining three main structures: Backbone, Neck, and Prediction. For instance, Mosaic data enhancement is used in the input position, the Focus structure is added at the beginning of the trunk, and the loss function is replaced at the position of the prediction head.

The general flow of object image detection using YoloV5 network architecture is shown in Fig. 1. The input end receives images into the CSPDarkNet53 network of the backbone for feature extraction, and then through the SPP structure [13], input images of multiscale are output to the pooling layer in an immobilized dimension. In the PANet structure [14], features of all levels are pooled to shorten the distance between the bottom and the top layer of the model; it enriches the characteristic information of each level. After detection and classification of prediction heads, the decoding network outputs, and the output results are suppressed by non-maximum suppression. Finally, the calculated results are saved.

B. Lightweight Openpose Algorithm Principle

There are typically two solutions to human pose estimation network. The first is a top-down approach, which uses object detection network to detect the character region in the image and then applies pose estimation algorithm. The representative algorithms include DeepPose [15], CPN [16], G-RMI [17], and HRNet [18]. The second is the bottom-up method where the skeleton key points of all figures in the image are first detected, and then grouped according to the number of people. The representative algorithms include DeeperCut [19], Associative Embedding [20], PifPaf [21], and HigherHRNet [22].

In 2017, Openpose [23], a human posture estimation network was proposed by Z. Cao et al. It is a bottom-up method as shown in Fig. 2. The network is mainly divided into two parts: neural network part and instantiation part. In the neural network part, the corresponding relation between the part confidence maps (PCM) and the part affinity fields (PAF) is used for inference operation. In the instantiation

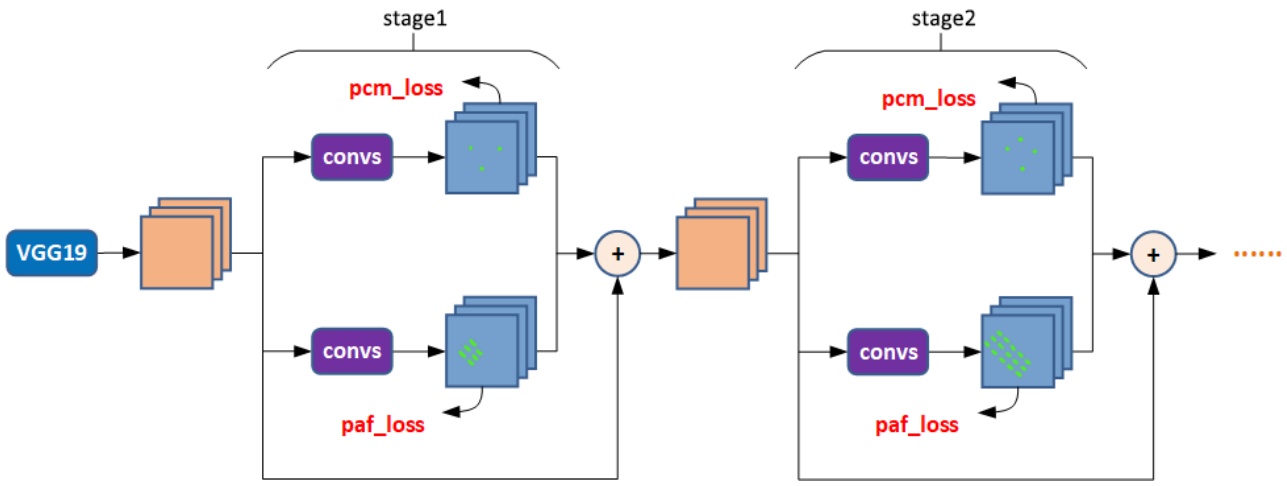


Fig. 2. Openpose network architecture.

section, the key points are grouped to make it clear which points belong to the same person. Finally, each refinement stage is combined to get the final result.

In 2018, the original Openpose network was optimized and a new Network Lightweight Openpose was proposed by Daniil Osokin [24]. Compared to Openpose with two refinement stages, the number of parameters is only 15%, the performance is similar, and it can reach 26FPS on CPU.

First, Lightweight Openpose abandoned the original VGG19 backbone, and chose MobileNetV1 [25] with fewer parameters and lower computation as the new backbone, thus improving the model computing speed. MobileNet consists of depthwise separable convolution, that is, standard convolution is decomposed into depthwise and pointwise convolutions, and uses dilated convolution [26] to enhance the receptive field.

In the refinement stage, the original Openpose constructs two branches, as shown in Fig. 3 (left). One branch is used to generate the part confidence maps, and the other branch is used to generate the part affinity fields. By comparing the two branches of the refinement stage, it can be found that the structure of the other stages is the same except the output stage. Therefore, Lightweight Openpose adopts the method of merging branches as shown in Fig. 3 (right) to reduce the redundant calculation of the model in the training stage.

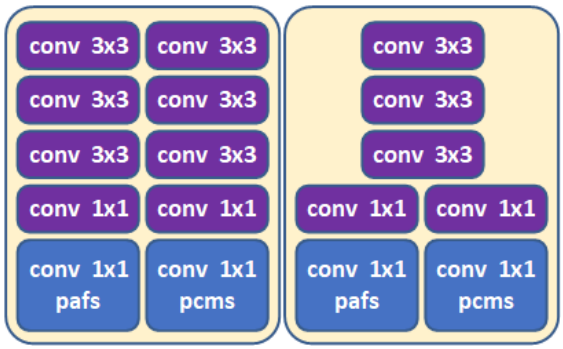


Fig. 3. Refinement stage of two network model structures.

In Lightweight Openpose, all 7×7 convolutions in original Openpose are replaced by three continuous convolutions as shown in Fig. 4. The computational load can be reduced by reducing the size of the convolution kernel, and the same receptive field can be guaranteed with the original 7×7 convolution.

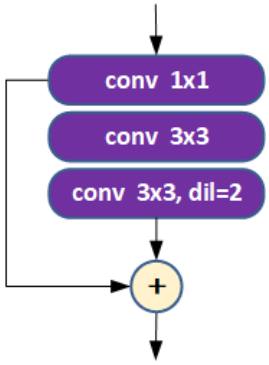


Fig. 4. Three continuous convolution blocks.

III. Improvement Strategy

A. Improvement Of Object Detection Prediction Head

In the problem of object detection, there are two definitions of small objects. The first is that the object area is less than 0.12% of the 256×256 image, which is defined as a relatively small object. The second is in the COCO dataset, where an object is defined as an absolute small size object when the dimensions of its box are less than 32×32 pixels. Taking our entire dataset as an example, the proportion of cigarettes and telephones conforms to the definition of relatively small objects. Therefore, the prediction head is improved such that the proposed model can detect small-size objects more accurately, to alleviate the negative impact caused by the drastic change of object size.

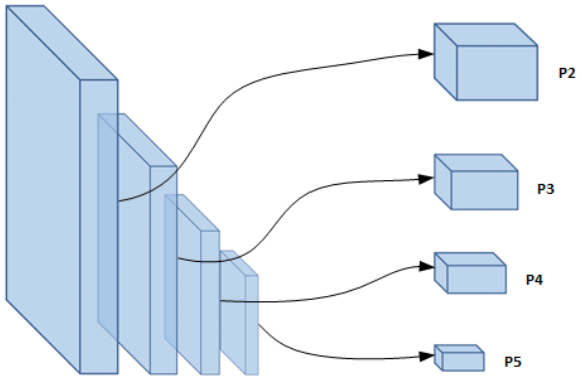


Fig. 5. Improved predictive head.

The original YoloV5 model has three prediction headers defined as P3, P4, and P5. As shown in Fig. 5, the detection layers corresponding to P3, P4, and P5 prediction heads are respectively 80×80, 40×40, and 20×20, which can be used to detect objects with a size above 8×8, 16×16, and 32×32, respectively. In shallow neural networks, neurons have smaller receptive fields and can extract more basic features by using fine-grained feature information, smaller objects in images can therefore be detected. On the contrary, as the number of convolutions increases, the receptive field gradually increases, and the overall contour of the image becomes clearer but the details in the image become more blurred. In order to detect small objects more accurately, the prediction head is improved by adding a P2 prediction head to the shallow network, which ensures that the network can capture more details when the receptive field overlapping area corresponding to each pixel of the feature map is small.

B. Structure Optimization Of Attitude Estimation Network

Pose estimation networks are often accompanied by high computational costs, which makes computing resources scarce. In the case of limited computing resources, an attention mechanism can more effectively focus on local information and provide a more reasonable resource allocation scheme for important tasks, thus reducing the attention to other information and improving the efficiency and accuracy of information processing.

To effectively eliminate the interference of irrelevant information and rationalize the weight distribution of the network, combining the spatial attention mechanism of

Coordinate Attention (CA) structure [27] with Openpose network structure is proposed. Each CA module acts as a computing unit to enhance the capability of feature representation. It not only considers the importance of modeling channel relations, but also considers the importance of location information of the feature space to generate space selection, so it can better bring maximum benefits to downstream tasks. Fig. 6 shows the CA module structure.

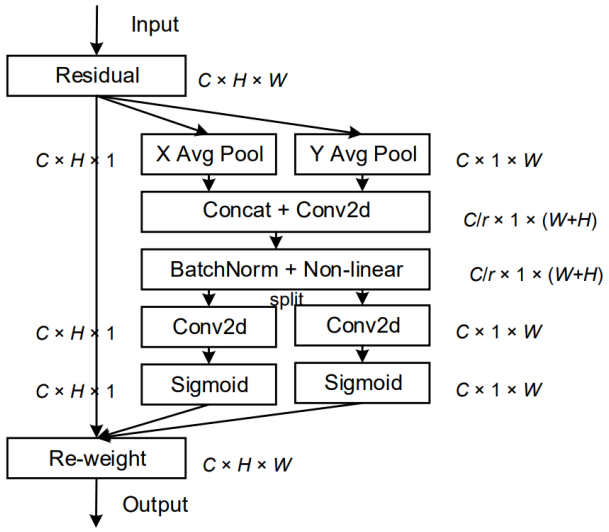


Fig. 6. Coordinate Attention module structure diagram.

As can be seen from the figure, the CA module first carries out average pooling of horizontal and vertical location information to obtain the perceptive attention diagram in two directions. In the spatial dimension, Concat and combine with 1×1 convolution to retrench the channels, and Batch Normalization (BN) and Non-Linear activation are used to encode spatial information in vertical and horizontal directions. After splitting, each channel is restored to the same channel number as the input feature graph through 1×1 convolution, and finally weighting is normalized. The specific CA module has two core operations: coordinate information embedding and coordinate attention generation.

First, in the coordinate information embedding operation, global pooling is decomposed into one-dimensional feature encoding operation. For the input feature $C \times H \times W$, pooling kernels of size (H, 1) and (1, W) are used to encode each channel along horizontal and vertical coordinate directions. The output representations of X average pooling and Y average pooling are shown in (1) and (2), respectively:

$$z_c^h(h) = \frac{1}{W} \sum_{0 \le i \le W} x_c(h,i) \tag{1}$$

$$z_c^w(w) = \frac{1}{H} \sum_{0 \le i \le H} x_c(j,w) \tag{2}$$

In the coordinate attention generation operation, in order to make better use of the feature graph with precise position information generated by the coordinate information embedding operation, a shared 1×1 convolution is used to transform F_1 as shown in (3) and generate $f \in R^{C/r \times (H+W)}$.

$$f = \delta(F_1(z^h, z^w)) \tag{3}$$

Then, f is cut into two separate tensors $f^h \in R^{C/r \times H}$ and $f^w \in R^{C/r \times W}$ along the spatial dimension. f^h and f^w are respectively restored to the same number of channels as the input feature graph by 1×1 convolution, as shown in (4) and (5):

$$g^h = \sigma(F_h(f^h)) \tag{4}$$

$$g^w = \sigma(F_w(f^w)) \tag{5}$$

The normalized weighting is performed as in (6):

$$y_c(i,j) = x_c(i,j) \times g_c^h(i) \times g_c^w(j) \tag{6}$$

After the CA module is embedded, part of the network structure diagram extracted by Openpose features is shown in Table I.

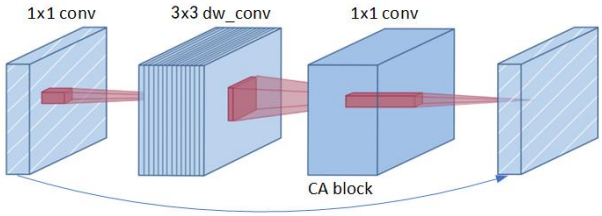


Fig. 7. Block module structure diagram.

The block module structure is shown in Fig. 7. On the original network, only use 3×3 depthwise convolution to extract image features. Since the depthwise convolution has no ability to change channels, it can only work in a lower dimension. Before each 3×3 depthwise convolution, use 1×1 pointwise convolution to raise the dimension, so that the 3×3 depthwise convolution can extract features in a higher-dimensional space, and then 1×1 pointwise convolution is used to reduce the dimension to restore the

TABLE I
OPENPOSE FEATURE EXTRACTION PART OF THE NETWORK STRUCTURE DIAGRAM

Network	Input	Conv Kernel	Input Channels	Output Channels	Stride	CA
Conv1	512×512	3×3	3	16	2	x
Block2	256×256	3×3	16	16	1	x
Block3	256×256	3×3	16	24	2	x
Block4	128×128	3×3	24	24	1	x
Block5	128×128	5×5	24	40	2	√
Block6	64×64	5×5	40	40	1	√
Block7	64×64	5×5	40	40	1	√
Block8	64×64	3×3	40	80	2	x
Block9	32×32	3×3	80	80	1	x
Block10	32×32	3×3	80	80	1	x
Upsampling	32×32	–	80	80	2	x

original number of channels. Finally, the input and output are connected through ShortCut structure to prevent network degradation caused by the increase of network depth.

C. Optimization of the Loss Function

The loss function is used to measure the difference between the predicted and the expected results of the neural network. In the object detection task, it usually consists of three parts: bounding box regression loss, classification loss, and confidence loss. In YoloV5, usually uses the binary cross entropy loss function to figure up the loss score of classification and confidence, and the BCEWithLogitsLoss function is used to calculate the loss in practical application. The BCEWithLogitsLoss function is used to contain relational categories. A target can belong to one or more categories, i.e., the target could be a cat, a British shorthair, Garfield, etc., with inclusion relations.

The BCEWithLogitsLoss function is composed of a Sigmoid function added into the BCELoss function, as shown in (7) and (8):

$$l(x,y)=\frac{1}{N}\sum_{n=1}^{N}l_n, \tag{7}$$

$$l_n=-w_n\left[y_n\cdot\log\sigma(x_n)+(1-y_n)\cdot\log(1-\sigma(x_n))\right] \tag{8}$$

Where $\sigma(x_n)$ represents the added Sigmoid function shown in (9):

$$\sigma(x_n)=Sigmoid(x)=\frac{1}{1+e^{-x_n}} \tag{9}$$

The Softmax function is used with probability and output of 1 and optimized to make its loss function more specific to our category mutual-exclusive dataset scenario. The form of the Softmax function is given in (10). For any real vector of length T, the Softmax function can compress its values in the (0,1) interval, and the sum of the elements in the vector is 1.

$$Softmax(z_i)=\frac{e^{z_i}}{\sum_{t=1}^{T}e^{z_i}},i=1,2,...,T \tag{10}$$

For the exponential terms shown in (10), the e^x function is depicted in Fig. 8. When x is large, "NaN" memory overflow occurs during code implementation, and when x is small, "invalid value encountered in true_divide" occurs.

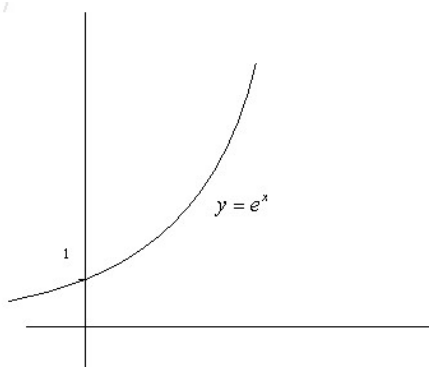


Fig. 8. e^x function graph.

To make the Softmax more numerically stable, the values of all indicators are processed at the same quantity level. First, the numerator and denominator of the Softmax function are multiplied by a constant W, and then normalized using the log function, as shown in (11):

$$\begin{aligned}Softmax(z_i)&=\frac{e^{z_i}}{\sum_{t=1}^{T}e^{z_i}}\\&=\frac{We^{z_i}}{W\sum_{t=1}^{T}e^{z_i}}\\&=\frac{e^{z_i+\log(W)}}{\sum_{t=1}^{T}e^{z_i+\log(W)}}\end{aligned} \tag{11}$$

Finally, the product of the vector output by the Softmax function and the target label vector is calculated to acquire the end result. The optimized loss function can be better applied to our scene to effectively improve the convergence effect of the overall model.

IV. Experiment

A. Building Our Dataset

Since there is currently no public data set on smoking and phone calls, our own dataset is used. The dataset is constructed in two forms: the first form mainly comes from self-shooting in real scenes, and the second form mainly comes from the network.

In the first form, data are collected from cameras placed in the rearview mirror of the car and on the dashboard of the driver under different angles, different lighting, and complex background conditions. A pixel resolution of 1920×1080 is used at 30FPS to film 10 volunteers' smoking driving behavior for 5 minutes and talking on the phone driving behavior for 5 minutes, and videos are generated in a safe and closed place.

Fig. 9 shows some examples from the dataset constructed for this paper.

Fig. 9. Real scene sample.

To improve the diversity of the dataset and the reliability of the model training results, a lot of pictures of smoking and making phone calls are collected from the Internet. Our model is trained in a more complex environment, with limited clarity in pictures and different pixel sizes, which makes it more challenging in the process of learning and training, as shown in Fig. 10.

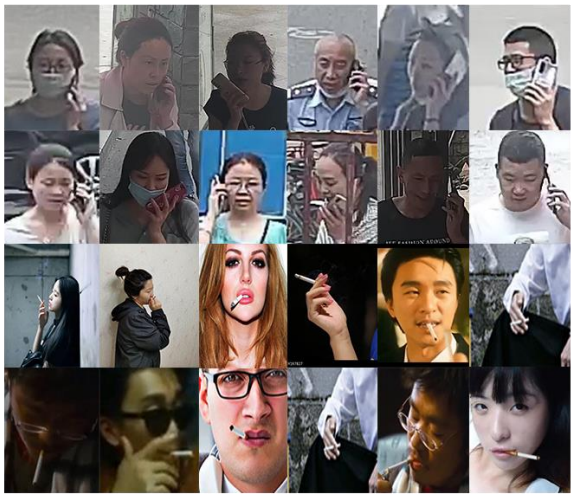


Fig. 10. Sample diagram of network collection.

LabelImg [28], a popular open-source annotation tool for object detection, is used to annotate the dataset.

B. *Experimental Method*

The experimental environment was built on Ubuntu 21.10 operating system installed with CUDA10.0 computing platform. Anaconda was used to create two virtual environments for object detection and pose estimation respectively. PyTorch 1.11 framework was used in the virtual environment of object detection, while version 1.2 was used in the pose estimation virtual environment. The object detection framework conducts training on the dataset split into training and validation sets in a 9:1 ratio. The pose estimation framework uses COCO2017 dataset [29] divided into training and validation sets in an 8:2 ratio. During the training phase, Masic and Mixup data augmentation are utilized as enhance the model's generalization ability and prevent over-fitting during training.

The overall process of the experiment is shown in Fig. 11. First, the input image is preprocessed by Gaussian filtering and histogram equalization to restrain the noise in the image, improve the image definition, and deal with the image defects. The improved Yolov5 model is used next to find the cigarettes and mobile phones in the image and obtain their location information. The improved Openpose model is then used to estimate the posture of the human body in the image, and the position information of the human nose and ears is obtained. Finally, the Euclidean distance is used as the main judgment condition, and elbow angle information is used as the auxiliary judgment condition to judge and match the final recognition results.

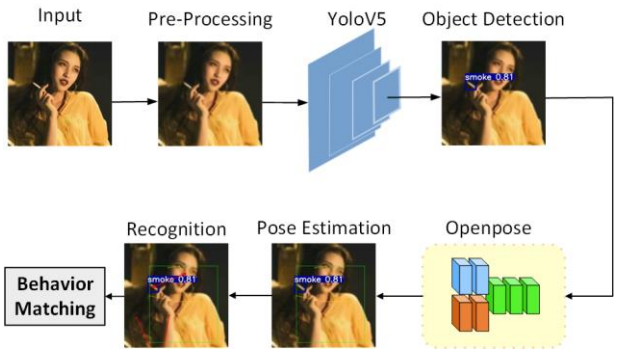
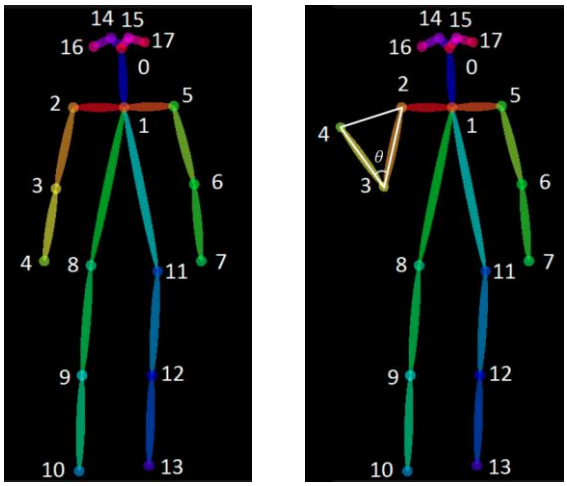


Fig. 11. Overall flow chart of experiment.

In our two-stage network structure, the model first uses the YoloV5 object detection framework to obtain features from the original image, and after the calculations of the model, detects target objects like cigarettes, telephones, and other objects that may cause abnormalities under specific circumstances. The position of the target object on the image is obtained by YoloV5 framework as shown in Fig. 12, and the coordinates of the center point of the target object are computed and saved:

$$A = (x_i, y_i) \tag{12}$$

The Lightweight Openpose [24] framework is then used to extract the key points' information of human pose skeleton. The heatmaps of bone key points can be obtained, as well as the corresponding relationship between points through the frame. Finally, the location information is obtained for 18 key points as shown in Fig. 13(a).



(a) 18 key points (b) Angle judgment

Fig. 13. Pose estimation diagram.

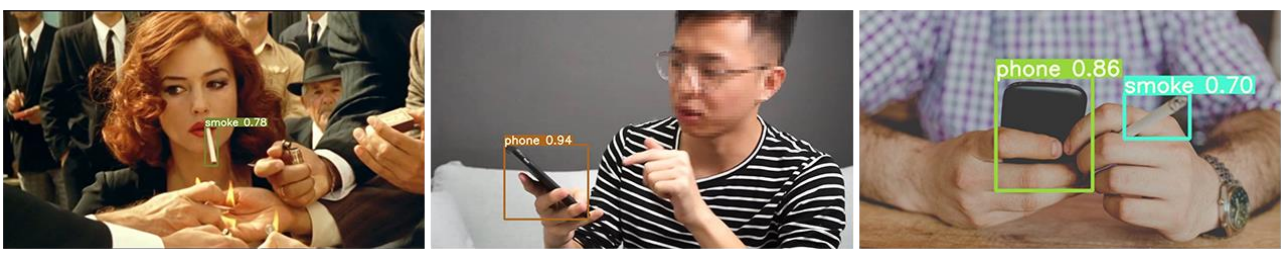
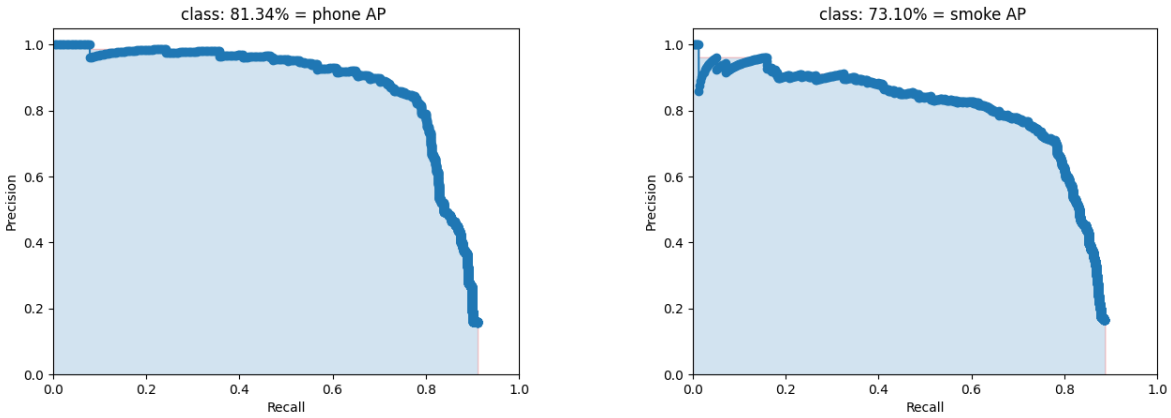


Fig. 12. Object detection framework sample result.



(a) "phone" P-R curve (b) "smoke" P-R curve

Fig. 14. P-R curve.

For smoking and talking on the phone, a combination of methods that calculate their distance and angular characteristics was developed to determine which behaviors the person exhibited. First, the location information of key points 0 (nose), 16 (right ear) and 17 (left ear) needs to be saved:

$$B = (x_j, y_j) \tag{13}$$

The Euclidean distance [30] between the phone and the left or right ear was used to determine whether the person in the image was talking on the phone, and the Euclidean distance between the cigarette and the nose was used to determine whether the person in the image was smoking. (12) and (13) can be used to obtain the 2D coordinates of the object and skeleton key points on the image. The Euclidean distance is calculated as follows:

$$d = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{14}$$

Through the calculation of Euclidean distance, when the result exceeds a certain threshold value, it can be judged whether the person has the behavior of smoking or talking on the phone. After judging the distance feature, the angle feature also needs to be considered. When people smoke or make phone calls, the elbow is bent at a certain angle. Judgment is made by calculating the angle θ between the wrist, elbow, and shoulder, as shown in Fig. 13(b).

Taking the right half of the human body as an example, the position coordinates of the right wrist, right elbow, and right shoulder can be obtained through the attitude estimation frame as in (15)-(17), respectively:

$$A = (x_4, y_4) \tag{15}$$

$$B = (x_3, y_3) \tag{16}$$

$$C = (x_2, y_2) \tag{17}$$

The side length is calculated for each side according to the coordinate information of the three points:

$$c = |AB| \tag{18}$$

$$a = |BC| \tag{19}$$

$$b = |AC| \tag{20}$$

The radian value is then calculated by the law of cosines:

$$\cos\theta = \frac{c^2 + a^2 - b^2}{2ac} \tag{21}$$

The use of angle features as an auxiliary judgment factor is because when a person is farther away from the camera, the person size will shrink, and the distance between the two key points will shrink. Similarly, if the person draw closer the camera, this person will be zoomed in, and the distance between the two key points will be enlarged. Therefore, it is far from enough to rely on the distance feature to judge, and the angle feature should be introduced as the constraint of the distance feature.

C. Experimental Results and Analysis

The precision, recall, mean Average Precision (mAP), and the frame rate (FPS), are frequently used performance evaluation metrics in the problem of object detection. The precision is from the perspective of the predicted results, which is used to indicate how many of the predicted positive results are true positive examples. The recall rate is from the perspective of real results, which is used to indicate how many real positive examples are recalled by the classifier. The calculation principles are shown in (22) and (23) respectively, where TP, FP, and FN represent the correct detection target, the wrongly detected target, and the undetected target, respectively.

$$Precision = \frac{TP}{TP + FP} \tag{22}$$

$$Recall = \frac{TP}{TP + FN} \tag{23}$$

The mAP is the average operation on the average precision (AP) of all categories. According to the precision and recall calculation rules, our model draws a P-R curve as shown in Fig. 14, which contains the relationship between precision and recall. The area enclosed by the P-R curve is AP, and mAP can be obtained by averaging all types of AP.

Table II shows a performance comparison of the object detection model before and after the improvement.

In the problem of pose estimation, the network structure is optimized, computational complexity is reduced, and the model's accuracy is improved. The initial learning rate is set to 0.00004, the batch-size to 80, and the stride to 16. We take Adam optimizer [31] to minimize the total training loss for all batches with backpropagation to update the model parameters. The model training is divided into three refinement stages, and their loss and average precision are

TABLE II
COMPARISON OF OBJECT DETECTION MODEL PERFORMANCE BEFORE AND AFTER IMPROVEMENT

Method	Precision(%)	Recall(%)	mAP(%)	Parameters	FLOPs	FPS
Before	88.83	29.17	68.20	20.85M	49.20G	27
After	82.51	70.38	77.20	22.98M	84.60G	25

TABLE III
COMPARISON OF THE PERFORMANCE OF THE POSE ESTIMATION MODEL BEFORE AND AFTER THE IMPROVEMENT

Method	AP(%)	AP^{50} (%)	AP^{75} (%)	AP^{M} (%)	AP^{L} (%)	FPS
Before	31.10	58.70	28.50	25.10	40.10	106
After	36.30	64.80	35.70	31.40	46.10	97

visualized. The loss value is saved after every epoch and compared with the loss value of the original model, as shown in Fig. 15(a). As can be seen from the figure, the convergence rate of our improved model is higher than that of the original model. For the evaluation indicators after the model improvement, as shown in Fig. 15(b), the validation set is verified every 15 epochs, and the average accuracy result is output after the verification.

The performance of the original model is compared with the performance of the improved model, as shown in Table III.

We use our own dataset to come true the end test experiment. Fig. 16 shows the number of objects in each category in our test set, including 473 data pictures in the smoke category, 305 pictures in the phone category, and 100 pictures in the normal category.

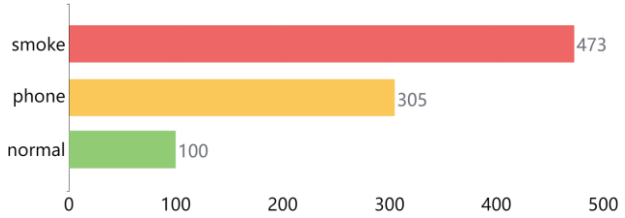


Fig. 16. Number of objects per category in the test set.

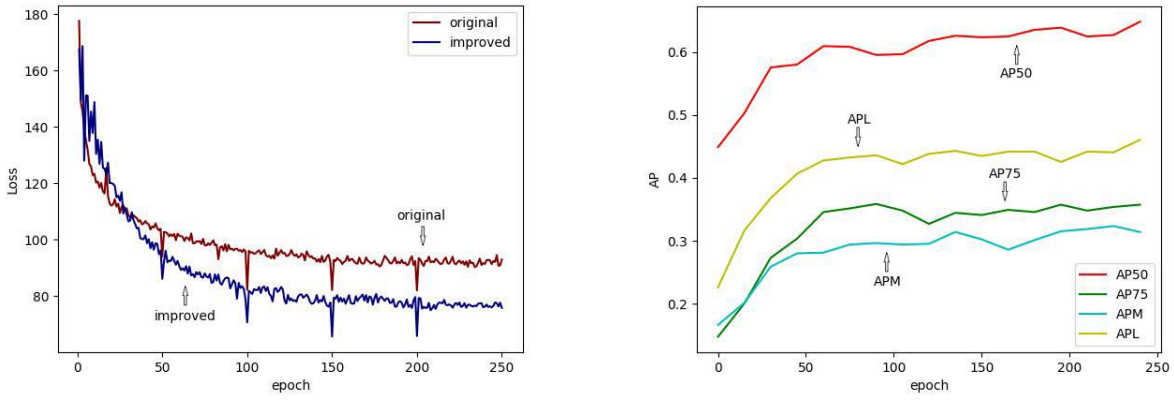
After repeated experiments on our test set, it is most appropriate to set the characteristic threshold of the distance between the target object and the human bone key points to 8. When the Euclidean distance is less than or equal to 8, the model judges that the driver has the dangerous driving behavior of smoking or talking on the phone. On the other hand, if the Euclidean distance is greater than 8, the model judges that the driver has no dangerous driving behavior. The improvement of the network and the implementation of the method are combined, and the accuracy of the proposed two-stage behavior anomaly detection method in the test set is shown in Table IV.

TABLE IV
TEST RESULTS OF OUR PROPOSED METHOD

Attribute No.	Attribute	Data Volume	Correct Quantity	AP(%)
0	smoking	473	430	91.00
1	phone-answering	305	290	95.08
2	normal	100	100	100.00
mAP	–	878	820	93.40

Attribute 0 means using the proposed method to identify whether the driver has smoking dangerous driving behavior. A total of 473 data pictures that meet the experimental conditions of attribute 0 in the test set, and the number of correct recognitions is 430; the recognition accuracy is therefore 91%. Experimental results are shown in Fig. 17. When the image is too bright or too dark, the behavior can be correctly recognized.

Fig. 17. Experimental results of Attribute 0.



(a) Loss value comparison image (b) Model-improved evaluation index image

Fig. 15. Pose estimation training visualization results.

Attribute 1 means using the proposed method to identify whether the driver has dangerous driving behavior when answering and making calls. In the test set, there are 305 data pictures that meet the experimental conditions of attribute 1, and the number of correct recognitions is 290, resulting in a recognition accuracy of 95.08%. Experimental results are shown in Fig. 18. In the process of receiving and making calls, the hand or face will cover a part of the mobile phone, which brings important challenges to the conventional recognition algorithm. Through the first stage of mobile phone target positioning, this problem can be well solved.

Fig. 18. Experimental results of Attribute 1.

Attribute 2 represents the use of the proposed method to identify the driver's normal driving behavior. There are 100 data pictures in the test set that meet the experimental conditions of attribute 2, and the number of correct recognitions is 100; the recognition accuracy is 100%. Experimental results are shown in Fig. 19. The phased identification and detection method saves a lot of time and cost. When the target is not detected in the first stage of the network model, it can be judged that the driver is not driving dangerously.

Fig. 19. Experimental results of Attribute 2.

The average accuracy of attributes 0, 1 and 2 is 93.4%, which indicates that our proposed method has certain recognition and detection ability.

D. Results Comparison

In order to verify the recognition effect of the proposed two-stage dangerous driving behavior recognition model, an experimental comparison is made models [32]-[34], with a comparative analysis in terms of accuracy. For fair comparison, training, verification, and testing, are conducted under the same equipment configuration, using the dataset presented in Section 4.1.

This paper needs to identify three driving behaviors: smoking while driving, talking on the phone while driving, and normal driving. Table V reports the results produced with different kinds of neural network algorithms on the recognition accuracy of dangerous driving behaviors. Convolutional neural network, cyclic neural network, and two-stage neural network algorithms were used for comparison and analysis.

TABLE V
COMPARISON OF THE ACCURACY OF DANGEROUS DRIVING BEHAVIOR RECOGNITION EXPERIMENTS

Reference	Method	mAP(%)
[32]	VGG-19	87.27
[33]	LSTM	86.61
[34]	Simple Baseline + ResNet	90.32
Our	YoloV5 + Openpose	93.40

In [32], VGG-19 convolutional neural network was used to treat the driver's dangerous driving problem as a multi-classification problem. Table VI shows the mutual comparison between error recognition data. From the table, it can be observed that the method proposed in [32] has the highest error rate in identifying normal driving. The main reason is that target occlusion and complex actions lead to misjudgments by the model. Additionally, the varying external environment is also a secondary factor affecting the accuracy.

In addition to using a convolutional neural network, the cyclic neural network long short-term memory (LSTM) is also used in [33] to identify and judge dangerous driving behaviors. In [33], a 3-layer LSTM structure that is more accurate than the convolutional neural network is used, but it also misjudges normal driving. It can be seen that the error rate is surprisingly high when the problem of dangerous driving behavior recognition is simply regarded as a classification problem.

[34] is somewhat similar to the one proposed in this paper, using a two-stage model to complete the identification process. Baseline is used in [34] to estimate the human posture first, and then ResNet is used for classification. The fusion of two-stage networks greatly reduces the occurrence of misjudgments, and the accuracy reaches a high level.

TABLE VI
ERROR IDENTIFICATION DATA COMPARISON

Reference	smoking	phone-answering	normal
[32]	44	52	16
[33]	51	41	8
[34]	56	29	0
Our	43	15	0

With reference to Table V and Table VI, the proposed method has the highest score in the comparison experiment in terms of recognition effect, which is attributed to the improved YoloV5 and Openpose having higher accuracy, and the recognition algorithm after the fusion of the two can still achieve the optimal effect.

E. Ablation Experiments

Since the difference of the two-stage recognition network is determined by the two model factors together, an ablation experiment was conducted on the part of the two-stage network model for the proposed dangerous driving behavior recognition. The effects of each component in the model are listed in Table VIII.

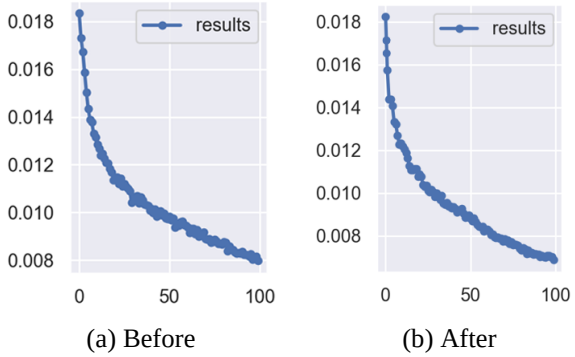
TABLE VIII
ABLATION STUDY OF COMPONENTS

Method	Pre-Head	Loss	Structure-Opt	mAP(%)
YoloV5+Openpose				85.20
Ours	√			89.64
Ours	√	√		90.89
Ours	√	√	√	93.40

When only the prediction head is added, the number of layers of YoloV5m is increased from 391 to 467. While increasing the amount of computation, mAP is also greatly improved. As can be seen from Fig. 20, our model still has good performance in the case of complex small target environment, limited picture resolution, and different pixel sizes.

Fig. 20. Add prediction head to visualize results.

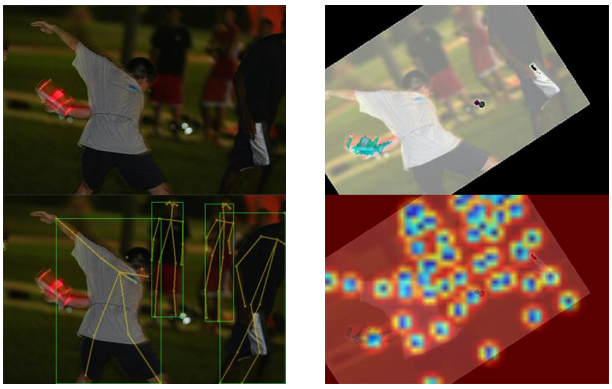
As can be seen from Fig. 21(a), the loss value of the model does not go below 0.008. After the object detection loss function is modified, this strategy can better converge advantageous during the training process, as shown in Fig. 21(b), the X-axis is the number of epochs, and the Y-axis is the loss.



(a) Before (b) After

Fig. 21. Training performance comparison before and after loss function optimization.

The network structure of attitude estimation is optimized to make it light enough and effective, and to ensure that accuracy and speed are both improved after further innovator. Fig. 22(a) shows the optimized network visualization result, which can detect the key points of the human body's skeleton very well, and the detection of the human body under a blurred background is also good. The heatmap is visualized in Fig. 22(b), and the robustness of the model is optimized by embedding an attention mechanism, so that our network achieves more competitive performance with less computations and parameters.



(a) Original and Result (b) Heatmap

Fig. 22. Pose estimation network structure visualization results.

V. CONCLUSION

In this study, a two-stage detection method for dangerous driving behavior recognition was proposed. The two dangerous driving behaviors, such as smoking or talking on the phone, can be well detected. Through this method, some unnecessary risks in our daily life can be avoided, such as traffic accidents or fires. A two-stage method of first object detection and then pose estimation was used, which can detect and locate targets, then judge the occurrence of a danger in advance. For the object detection model, the ability to detect small target objects was improved. Although some computational costs are increased, it brings a high accuracy improvement. For the pose estimation model, its network structure is optimized so that it can focus on more local information with lower computational resources. Our model improves accuracy while maintaining speed. It can be used for real-time camera monitoring and is an efficient behavioral anomaly detection model.

REFERENCES

[1] Jiasong Zhu, Weidong Lin, Ke Sun, Xianxu Hou, Bozhi Liu, Guoping Qiu: Behavior Recognition of Moving Objects Using Deep Neural Networks. SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI 2018: 45-52.

[2] Weiming Chen, Zijie Jiang, Hailin Guo, Xiaoyang Ni: Fall Detection Based on Key Points of Human-Skeleton Using OpenPose. Symmetry 12(5): 744 (2020).

[3] Haotian Wang: Residual Mask Based on MobileNet-V2 for Driver's Dangerous Behavior Recognition. CSAI 2019: 196-199.

[4] Junwu Weng, Chaoqun Weng, Junsong Yuan: Spatio-Temporal Naive-Bayes Nearest-Neighbor (ST-NBNN) for Skeleton-Based Action Recognition. CVPR 2017: 445-454.

[5] Yizhou Zhou, Xiaoyan Sun, Zheng-Jun Zha, Wenjun Zeng: MiCT: Mixed 3D/2D Convolutional Tube for Human Action Recognition. CVPR 2018: 449-458.

[6] Sijie Yan, Yuanjun Xiong, Dahua Lin: Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition. AAAI 2018: 7444-7452.

[7] Karen Simonyan and Andrew Zisserman, “Two-stream convolutional networks for action recognition in videos,” NIPS, vol. 27, pp. 568–576, 2014.

[8] Zhaofan Qiu, Ting Yao, Tao Mei: Learning Spatio-Temporal Representation with Pseudo-3D Residual Networks. ICCV 2017: 5534-5542.
[9] Jeff Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Trevor Darrell, Kate Saenko: Long-term recurrent convolutional networks for visual recognition and description. CVPR 2015: 2625-2634.
[10] Huayi Zhou, Fei Jiang, Hongtao Lu: student dangerous behavior detection in school. CoRR abs/2202.09550 (2022).
[11] Haoshu Fang, Shuqin Xie, Yu-Wing Tai, Cewu Lu: RMPE: Regional Multi-person Pose Estimation. ICCV 2017: 2353-2362.
[12] Alexey B, Chien-Yao W, Hong-Yuan M L. "YOLOv4: Optimal Speed and Accuracy of Object Detection," arXiv: 2004.10934, 2020.
[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun: Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. IEEE Trans. Pattern Anal. Mach. Intell. 37(9): 1904-1916 (2015).
[14] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, Jiaya Jia: Path Aggregation Network for Instance Segmentation. CVPR 2018: 8759-8768.
[15] Alexander Toshev, Christian Szegedy: DeepPose: Human Pose Estimation via Deep Neural Networks. CVPR 2014: 1653-1660.
[16] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, Jian Sun: Cascaded Pyramid Network for Multi-Person Pose Estimation. CVPR 2018: 7103-7112.
[17] George Papandreou, Tyler Zhu, Nori Kanazawa, Alexander Toshev, Jonathan Tompson, Chris Bregler, Kevin Murphy: Towards Accurate Multi-person Pose Estimation in the Wild. CVPR 2017: 3711-3719.
[18] Ke Sun, Bin Xiao, Dong Liu, Jingdong Wang: Deep High-Resolution Representation Learning for Human Pose Estimation. CVPR 2019: 5693-5703.
[19] Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, Bernt Schiele: DeeperCut: A Deeper, Stronger, and Faster Multi-person Pose Estimation Model. ECCV (6) 2016: 34-50.
[20] Alejandro Newell, Zhiao Huang, Jia Deng: Associative Embedding: End-to-End Learning for Joint Detection and Grouping. NIPS 2017: 2277-2287.
[21] Sven Kreiss, Lorenzo Bertoni, Alexandre Alahi: PifPaf: Composite Fields for Human Pose Estimation. CVPR 2019: 11977-11986.
[22] Bowen Cheng, Bin Xiao, Jingdong Wang, Honghui Shi, Thomas S. Huang, Lei Zhang: HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose Estimation. CVPR 2020: 5385-5394.
[23] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, Yaser Sheikh: OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. CoRR abs/1812.08008 (2018).
[24] Daniil Osokin: Real-time 2D Multi-Person Pose Estimation on CPU: Lightweight OpenPose. CoRR abs/1811.12004 (2018).
[25] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, Hartwig Adam: MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. CoRR abs/1704.04861 (2017).
[26] Fisher Yu, Vladlen Koltun, Thomas A. Funkhouser: Dilated Residual Networks. CVPR 2017: 636-644.
[27] Qibin Hou, Daquan Zhou, Jiashi Feng: Coordinate Attention for Efficient Mobile Network Design. CVPR 2021: 13713-13722.
[28] D Tzutalin, "Labelimg (2015)," GitHub repository https://github.com/tzutalin/labelImg, vol. 6.
[29] Umar Iqbal, Juergen Gall: Multi-person Pose Estimation with Local Joint-to-Person Associations. ECCV Workshops (2) 2016: 627-642.
[30] Zachary I. Bell, Patryk Deptula, Emily A. Doucette, J. Willard Curtis, Warren E. Dixon: Simultaneous Estimation of Euclidean Distances to a Stationary Object's Features and the Euclidean Trajectory of a Monocular Camera. IEEE Trans. Autom. Control. 66(9): 4252-4258 (2021).
[31] Diederik P. Kingma, Jimmy Ba: Adam: A Method for Stochastic Optimization. ICLR (Poster) 2015.
[32] Arief Koesdwiady, Safaa M. Bedawi, Chaojie Ou, Fakhri Karray: End-to-End Deep Learning for Driver Distraction Recognition. ICIAR 2017: 11-18.
[33] Li Li Research on Driver Fatigue and Distraction Status Recognition Based on CNNs and LSTM [D]. Hunan University, 2018.
[34] Zhishuai Yin, Shu Zhong, Linzhen Nie, Chen Ma. Detection of distracted driving behavior based on human posture estimation [J]. Journal of China Highway Engineering, 2022,35 (06): 312-323. DOI: 10.19721/j.cnki.1001-7372.2022.06.06.26.