Improved YOLOv4-tiny based on attention mechanism for skin detection

Ping Li [1,2], **Taiyu Han** [1], **Yifei Ren** [1], **Peng Xu** [1], **Hongliu Yu** [Corresp. 1]

[1] Institute of Rehabilitation Engineering and Technology, University of Shanghai for Science and Technology, Shanghai, China

[2] Department of Biomedical Engineering, Changzhi Medical College, Changzhi, Shanxi, China

Corresponding Author: Hongliu Yu
Email address: yhl_usst@outlook.com

Background. The automatic bathing robot needs to identify the area to be bathed to perform visually guided bathing tasks. The visual perception of the skin area is the first step in the operation of an automatic bathing robot. The deep CNN-based object detection algorithm has excellent robustness to light and environmental changes when performing skin detection. The one-stage object detection algorithm has good real-time performance, which is widely used in practical projects. **Methods.** In our previous work, we perform skin detection using several models and find that YOLOv4 has best comprehensive performance. This study uses the YOLOv4-tiny model for skin detection considering the convenience of practical deployment. In addition, we add three kinds of attention mechanisms to strengthen feature extraction, namely SE, ECA, and CBAM. In particular, we add the attention module to the two feature layers of the backbone output. In the enhanced feature extraction network part, we apply the attention module to the upsampled features. **Results.** The experimental results reveal that the weight file of YOLOv4-tiny without attention mechanisms is reduced to 9.2% of YOLOv4, but the mAP maintains 67.3% of YOLOv4. The performance of the YOLOv4-tiny is improved by combining the CBAM or ECA modules, but the addition of SE deteriorates the performance of YOLOv4-tiny instead. Among them, CBAM is the best, which can enhance YOLOv4-tiny detection accuracy by about 5%.

Improved YOLOv4-tiny based on attention mechanism for skin detection

Ping Li[1,2], Taiyu Han[1], Yifei Ren[1], Peng Xu[1], Hongliu Yu[1]
[1] Institute of Rehabilitation Engineering and Technology, University of Shanghai for Science and Technology, Shanghai, China
[2] Department of Biomedical Engineering, Changzhi Medical College, Changzhi, Shanxi, China
Corresponding Author:
Hongliu Yu[1]
Email address: yhl_usst@outlook.com

Improved YOLOv4-tiny based on attention mechanism for skin detection

Ping Li[1,2], Taiyu Han[1], Yifei Ren[1], Peng Xu[1], Hongliu Yu[1]
[1] Institute of Rehabilitation Engineering and Technology, University of Shanghai for Science and Technology, Shanghai, China
[2] Department of Biomedical Engineering, Changzhi Medical College, Changzhi, Shanxi, China
Corresponding Author:
Hongliu Yu[1]
No. 580, Jungong Road, Yangpu District, Shanghai, 200093, China
Email address: yhl_usst@outlook.com

Abstract

Background. The automatic bathing robot needs to identify the area to be bathed to perform visually guided bathing tasks. The visual perception of the skin area is the first step in the operation of an automatic bathing robot. The deep CNN-based object detection algorithm has excellent robustness to light and environmental changes when performing skin detection. The one-stage object detection algorithm has good real-time performance, which is widely used in practical projects.

Methods. In our previous work, we perform skin detection using several models and find that YOLOv4 has best comprehensive performance. This study uses the YOLOv4-tiny model for skin detection considering the convenience of practical deployment. In addition, we add three kinds of attention mechanisms to strengthen feature extraction, namely SE, ECA, and CBAM. In particular, we add the attention module to the two feature layers of the backbone output. In the enhanced feature extraction network part, we apply the attention module to the upsampled features.

Results. The experimental results reveal that the weight file of YOLOv4-tiny without attention mechanisms is reduced to 9.2% of YOLOv4, but the mAP maintains 67.3% of YOLOv4. The performance of the YOLOv4-tiny is improved by combining the CBAM or ECA modules, but the addition of SE deteriorates the performance of YOLOv4-tiny instead. Among them, CBAM is the best, which can enhance YOLOv4-tiny detection accuracy by about 5%.

Introduction

CNN (Convolutional Neural Network) is a machine learning model in a supervised learning framework. In 2012, AlexNet first used CNN for image classification (Krizhevsky, Sutskever & Hinton, 2017), winning the ImageNet large scale visual recognition challenge by an overwhelming margin. Since then, CNN has been widely used in computer vision tasks such as image classification (Liu, Soh & Lorang, 2021) and object detection (Zhou et al., 2022). Using massive data as learning samples, we can obtain a CNN model with analysis capability, feature representation capability, and recognition capability to achieve skin detection. In CNN models, the convolutional layer extracts features, the pooling layer performs dimensionality reduction and information integration, and the fully connected layer combines the extracted features and outputs data adapted to a specific problem. In general, an activation function is introduced to give the model a nonlinear representation capability. With the compression of the width and height of feature maps, CNN models acquire robust semantic information and abstract extracted features.

Skin perception for automatic bathing robots is a prerequisite for bathing. The intelligent bathing system detects the human skin in the bathing scene based on vision sensors. Skin detection in bathing scenes is a challenging task. From the environmental perspective, the bathing scene is full of water mist and own various lighting and backgrounds. A skin detection algorithm generally extracts skin features and then classifies them using a classifier. Traditional skin detection typically exploits handcrafted features to distinguish between skin and non-skin zones, such as color, texture, and statistical features. Zhang proposes a skin color model based on reference points of the face (Zhang et al., 2022). Shifa conducts skin detection by combined threshold rule-based segmentation in the RGB, HSV, and YCbCr color spaces (Shifa et al., 2020). Sun uses a local skin color model to change a global model performing skin detection for single images (Sun, 2010). Luo performs skin detection by face location method and facial structure estimation (Luo & Guan, 2017). Javadi conducts the skin lesions detection by color properties using a genetic algorithm for selecting the best features and determines the 3D position by Kinect camera (Javadi & Soltanizadeh, 2021). Handcrafted features are not robust to environmental changes and are insufficient for bathing scenarios. Skin detection based on machine learning, which generally uses supervised methods to construct detectors to extract skin features, is less influenced by environmental factors and has gained more applications in recent years. Salah utilizes CNN trained by the skin and non-skin patches to detect skin pixels (Salah, Othmani & Kherallah, 2022). Kim exploits two CNNs for skin detection and compares performance using different training strategies (Kim, Hwang & Cho, 2017). Lin conducts the CNN-based facial skin detection and optimizes the CNN with the Taguchi method (Lin et al., 2021).

Different from just identifying skin and non-skin areas, we need to provide the robot with information about specific skin areas (hands, feet, head, etc.) to clean up skins using different modes. We are facing a multi-classification problem rather than a secondary classification problem. For the bathing scenario, traditional algorithms become inadequate. Therefore, deep learning methods are introduced. Skin detection based on deep CNN does not rely on handcrafted features. In application areas, one-stage object detection models based on CNN achieve good real-time performance and are computationally efficient, such as the YOLO models (Redmon et al., 2016; Redmon & Farhadi, 2017; Redmon & Farhadi, 2018; Bochkovskiy, Wang & Liao, 2020). The one-stage framework eliminates the proposals generation and outputs the categories and bounding boxes directly.

We perform skin detection based on deep learning methods. Our research is based on previous work by our team (Li et al., 2022), which finds that the YOLOv4 algorithm has a high mAP for skin detection in bath scenes. At the same time, it has extensive computation, leading to the slow speed of YOLOv4 after being deployed to embedded devices. In the study, we adopt YOLOv4-tiny (Zhao et al., 2022a) for skin detection, the lightweight model of YOLOv4, and investigate the effect of attention mechanisms on the YOLOv4-tiny.

The remaining parts of the paper are arranged as follows: "Materials & Methods" section offers an introduction to data sets acquisition, YOLOv4-tiny, improved YOLOv4-tiny based on attention mechanisms, transfer learning, experimental setup, and evaluation indicators. The "Results" section describes the experimental results. The "Discussion" section discusses the results related to our application scenarios. The "Conclusion" section summarizes our research and looks at future work.

Materials & Methods

Data sets acquisition

A total of 1500 images containing human skin are collected, considering factors such as position, illumination, resolution, blurring, and the presence of water mist. Finally, our data sets choose 1000 images based on the image quality. The image annotation tool LabelImg (Bhatt et al., 2022) is used to generate XML files corresponding to the images. The XML file includes the file name, ground truth box information, and category information in the corresponding image. Example images in the data set are exhibited in Fig. 1.

YOLOv4-tiny

The structure of YOLOv4-tiny is shown in Fig. 2. The backbone is CSPDarknet53-tiny, which is utilized for feature extraction. CSPDarknet53-tiny is composed of DarknetConv2D_BN_Leaky modules and Resblock_body modules. A DarknetConv2D_BN_Leaky module combines a two-dimensional convolutional layer, a normalized processing layer, and an activation function. The Mish activation function (Misra, 2019) in the YOLOv4 is replaced by a Leaky Relu function (He et al., 2015) to improve detection speed. The structure of Resblock_body is illustrated in Fig. 3. The skip connection can better combine the semantic information and let the model converge quickly, preventing both model degradation and gradient disappearance (Furusho & Ikeda, 2020). Feat1 and Feat2 are the output feature layers from the Resblock_body module. The Feat2 output branch of the first two Resblock_body modules is the input of the next module. FPN (Lin et al., 2017) is used for enhancing feature extraction and performing feature fusion to combine feature information at different scales. For the output of the third Resblock_body module, Feat1 is directly used as the first input of the FPN. The second input of the FPN is the result obtained by processing Feat2 by the DarknetConv2D_BN_Leaky module. The output P2 of FPN is obtained by convolution processing on the second input of the FPN. The output P1 of FPN is obtained by stacking Feat1 and the result which is obtained by convolution and up-sampling operations on P2. The structure of FPN is simple, allowing YOLOv4-tiny to have excellent real-time performance. Compared with YOLOv4, YOLOv4-tiny has two detection heads and predicts at two scales. The YOLO head is used to obtain classification and regression prediction results. The structure of the YOLO head is straightforward. The two prediction feature layers for prediction are acquired by a small amount of convolution of P1 and P2. YOLOv4-tiny is still making the detection based on anchors, using fixed-size anchors as a prior for object boxes, tiling many anchors on images, and adjusting anchors to bounding boxes by the prediction results. “13×13” and “26×26” represent the granularity of grids. “33” represents the prediction results adapted to our application, i.e., 3×(4+1+6), where “3” represents the number of anchors, “4” indicates the number of location parameters, “1” denotes the confidence score, and “6” means the number of categories to be identified.

The loss function generally includes bounding box location loss L_{loc}, classification loss L_{cls}, and confidence loss L_{conf}. The overall loss L is calculated as Eq. (1).

$$L = L_{loc} + L_{cls} + L_{conf} \quad (1)$$

L_{loc} measures the position error (height h, width w, and central coordinates) between the prediction box and the GT box. The evaluation indicators include IOU, GIOU (Rezatofighi et al., 2019), DIOU, and CIOU (Zheng et al., 2019), as summarized in Table 1. We introduce CIOU loss as L_{loc}, as indicated in Eq. (2).

$$L_{loc} = 1 - IoU + \rho^2\left(b, b^{gt}\right) \big/ d^2 + \alpha\upsilon \quad (2)$$

$$\upsilon = 4 \big/ \left(\pi^2\right) * \left(\arctan\left(w^{gt}/h^{gt}\right) - \arctan\left(w/h\right)\right)^2 \quad (3)$$

$$\alpha = \upsilon \big/ \left(\upsilon + 1 - IoU\right) \quad (4)$$

$\rho^2(b,b^{gt})$ represents the European distance between the central points of the prediction box and the GT box, d represents the diagonal distance of the minimum enclosed area, including the prediction box and the GT box, α is weight, and υ expresses the consistency of aspect ratio. α and υ are calculated as demonstrated in Eq. (3) and Eq. (4).

L_{cls} measures the category error between the prediction box and the GT box, as shown in Eq. (5).

$$L_{cls} = -\sum_{i=0}^{K\times K} I_{ij}^{obj} \sum_{c\in classes} \left[\hat{p}_i(c)\log\left(p_i(c)\right) + \left(1-\hat{p}_i(c)\right)\log\left(1-p_i(c)\right)\right] \quad (5)$$

$K\times K$ represents the number of grids on feature maps of different scales, and c represents the category. If the j-th prior box of the i-th grid has objects to be predicted, I_{ij}^{obj}=1; otherwise, I_{ij}^{obj}=0.   (c) and $p_i(c)$ represent the actual value and predicted value of the probability that the j-th prior box of the i-th grid belongs to category c, respectively.

L_{conf} adopts a cross-entropy loss function, as shown in Eq. (6). M represents the number of prior boxes.   and C_i represent the actual and predicted values of confidence. If the j-th prior box of the i-th grid has no object to be predicted, I_{ij}^{noobj}=1; otherwise, I_{ij}^{noobj}=0.

$$L_{conf} = \sum_{i=0}^{K\times K}\sum_{j=0}^{M} I_{ij}^{obj}\left[\hat{C}_i \log\left(C_i\right) + \left(1-\hat{C}_i\right)\log\left(1-C_i\right)\right] - \sum_{i=0}^{K\times K}\sum_{j=0}^{M} I_{ij}^{noobj}\left[\hat{C}_i \log\left(C_i\right) + \left(1-\hat{C}_i\right)\log\left(1-C_i\right)\right] \quad (6)$$

Improved YOLOv4-tiny based on attention mechanisms

The attention mechanism is a normal tip for deep learning, which has a variety of implementations (Niu, Zhong & Yu, 2022). The core of the attention mechanism is to make the network pay attention to where it needs more attention. In general, attention mechanisms can be divided into the channel attention mechanism, the spatial attention mechanism, and a combination of the two (Tian et al., 2021).

In this paper, the following attention mechanisms are used:

(1) SE (Squeeze-and-Excitation) (Hu, Shen & Sun, 2018). SE is a typical implementation of the channel attention mechanism, obtaining the weights of each channel in the feature maps. The inter-dependencies among channels are modeled explicitly. Instead of introducing a new-built spatial dimension for the fusion of feature channels, SE uses a feature rescaling strategy. Specifically, the importance of each channel is acquired spontaneously by self-learning. Following the degree of matter, the helpful features are enhanced, and the useless features are suppressed, achieving the adaptive calibration of feature channels. SE includes squeeze and excitation operations. The squeeze operation conducts feature compression across the spatial dimension, converting a two-dimensional feature map into a real number that owns a global receptive field. The output size matches the number of input channels. The excitation operation is equivalent to the mechanics of gates in recurrent neural networks, where weights are created for each channel employing learned parameters, explicitly modeling the correlation between feature channels. Finally, the weights, which are output by excitation operation, represent importance of each channel. The rescaling of features in the channel dimension is accomplished by multiplying the weights by features of each channel (Huang et al., 2019). The specific implementation of SE is shown in Fig. 4.

(2) ECA. ECA is an improved version of SE. Wang argues that seizing all channel dependencies is ineffective and unessential for SE block (Wang et al., 2020). Convolution operation owns the cross-channel information capture capability. ECA removes the fully connected layer of SE and learns weights by 1D convolution operation on the globally averaged pooled features. The specific implementation of ECA is shown in Fig. 5.

(3) CBAM (Convolutional Block Attention Module). CBAM (Woo et al., 2018) performs channel attention and spatial attention mechanism processing for feature maps, respectively. The specific implementation of CBAM is shown in Fig. 6. The implementation of the channel attention module (CAM) can be divided into two parts. Global average pooling and maximum global pooling are performed separately for the input feature maps. The outputs are processed using a shared, fully connected layer. Sum the two processed results, and then take the sigmoid operation, obtaining the weights (between 0 and 1) of each channel of the input features. The weights are multiplied by the original input features to get the output of CAM. The spatial attention module (SAM) takes the maximum value and the average value on each channel of each feature point. The two results are stacked. Adjust the number of channels using a convolution operation. Get the weights of each feature point of the input features through the following sigmoid function. Obtain the final output by multiplying the weights by the original input features.

In this study, the above attention mechanisms are applied to the YOLOV4-tiny. As shown in Fig. 7, we add attention mechanisms on the two feature layers extracted from the backbone network and attention mechanisms on the up-sampled results in FPN.

Transfer Learning

Training a network from scratch requires a enormous amount of labeled data. Manual labeling of data sets is time-consuming and labor-intensive, which introduces human error. Small data sets combined with transfer learning techniques can produce a desirable model quickly (Pratondo & Bramantoro, 2022). The ImageNet contains more than 14 million images covering more than 20,000 categories, of which more than one million images have explicit annotations and corresponding labels at objects' locations in the image (Russakovsky et al., 2015). The pre-trained models on ImageNet can learn fundamental features such as textures, lines, etc., which are general in object detection. The pre-trained weights on the ImageNet are the initial weights for all models in this study.

Experimental setup and evaluation indicators

For the fairness of model comparison, we use the same data sets as our previous work (Li et al., 2022), with a ratio of 60%:20%:20% for the training, validation, and test sets. All models are trained with the help of the high-performance computing center of the University of Shanghai for Science and Technology. Mosaic data augmentation is used in the training process in which four randomly stitched images are input to the network for training to increase the background diversity (Bin et al., 2022). The learning rate cosine decline strategy is used during the model training. The loss function is optimized using a label smoothing approach to suppress the overfitting problem during training (Zhang et al., 2021). The probability (c) distribution before and after label smoothing is shown in Eq. (7), with $\delta = 0.05$.

$$\hat{p}_i(c) = \begin{cases} 1, i = y \\ 0, i \neq y \end{cases} \Rightarrow \hat{p}_i(c) = \begin{cases} 1-\delta, i = y \\ \delta, i \neq y \end{cases} \quad (7)$$

We use the Pytorch framework for model building and training. The initial value of the learning rate is set to 0.001, and the decay rate is set to 0.01. The batch size is set to 16, which indicates the number of images input to the model for training every time. SGD is utilized as the optimizer for model training. When training, the weights of the backbone are frozen first for 50 epochs, and all weights are trained after 50 epochs, which increases the convergence speed and training performance of models.

Recall and precision can be used to measure performance but are not fully representative of the detector quality. Many sets of recall and precision values are obtained by taking different thresholds. Then, plot a P-R curve (Naing et al., 2022). AP characterizes the area enclosed by the P-R curve and the coordinate axes. The sum of AP values of all classes is then divided by the total number of classes to get mAP, which is the crucial evaluation metric of detectors for multiple categories detection.

Results

After the training is completed, models are selected based on the results of the validation sets, and the performance is tested using the test sets. The mAPs and weight file information of models are exhibited in Table 2.

In our previous work, the mAP of YOLOv4 reaches 78%, but it has a weight file of 244 MB. After the light-weighting process, the mAP of YOLOv4-tiny is 67.3% of YOLOv4, but the weight file is reduced to 9.2% of YOLOv4. Based on the YOLOv4-tiny, we add attention mechanisms as shown in Fig. 7. As can be seen from Table 2, the detection result decreases instead, and mAP is reduced by 0.9% after adding the SE. There is a 1.1% improvement in mAP after adding the ECA. The performance improvement is the highest with the addition of CBAM, in which mAP is increased by nearly 5%. After adding ECA, the weight file has hardly increased. After adding SE, the weight file has increased by 0.2 M. After adding CBAM, the weight file has increased by 0.4 M. We have established a comprehensive indicator W, as shown in Eq. (8). A indicates the change of weight file, and B indicates the shift in mAP. When mAP is less than the mAP of the original YOLOv4-tiny model, B takes a negative value. Otherwise, B takes a positive value. The smaller the A, the better the effect. The higher the B, the better the performance. Overall, the higher the W, the better the outcome. After calculation, A, B, and W are indicated in Table 3. CBAM_YOLOv4-tiny, which introduces the CBAM modules, achieves the best outcome.

$$W = \frac{B}{e^{A}} \tag{8}$$

The AP values for the six categories are shown in Table 4, and P-R curves are exhibited in Fig. 8. Table 4 displays that CBAM_YOLOv4-tiny achieves the highest AP values in all categories except for the upper limb. For the upper limb, YOLOv4-tiny combined with ECA reaches the highest AP value.

Discussion

To perform the bathing tasks, we need to recognize the area to be bathed in the bathing scenario and send the recognition information to the bathing robot arm for bathing behavior planning, as shown in Fig. 9. By combining the skin detection results of 2D images with the depth information obtained from the depth camera, we can model the localization of targets in 3D space. To facilitate the robot to implement distinct bathing patterns for areas of the body, we need to identify the skin located at diverse parts of the body. Therefore, we build small data sets in the bathing scenarios to be used as learning samples for object detection models. And the manual annotation is performed with a labelImg tool to classify skin regions into six categories according to different parts.

Among object detection algorithms, one-stage detection algorithms are faster than two-stage and are suitable for application in our scenario where real-time performance is required. In our previous work, we explore the effectiveness of object detection models for skin detection with multiple classifications and find the best YOLOv4 model from five models. We lightweight YOLOv4 and impose three kinds of attention mechanisms on the YOLOv4-tiny. We find that both CBAM and ECA improve the detection effect. Yet, SE makes the detection effect worse instead, which implies that we need carefully choose the attention mechanism during practice. Compared with Salah's work, we input data sets including images with six types of labels for network training instead of skin and non-skin patches. We do not only identify skin or non-skin, but also we want to know to which part of the body the skin belongs. To the best of our knowledge, this is the first time we have investigated skin detection that can identify different body parts.

There is rare research on object detection-based skin detection combined with robotic arms for bathing tasks. Our study lightweights the YOLOv4 model and explores which attention mechanism works best by imposing attention mechanisms on the YOLOv4-tiny model. However, the YOLOv4-tiny possesses a reduction in mAP compared with the YOLOv4, creating some challenges for high detection accuracy (Zhao et al., 2022b). The relatively small number of trunks in the data sets results in poor detection of trunks because of individual privacy issues. The foot occupies a small area in the whole body range. Foot features tend to disappear with repeated down-sampling operations, resulting in poor detection of the foot.

Conclusions

When using robots for autonomous bathing tasks, the perception of skin in bathing scenarios needs to be accomplished first. To facilitate the embedded deployment, we use YOLOv4-tiny, a lightweight model of YOLOv4, for skin recognition research based on our previous work. Three kinds of attention mechanisms are overlaid in the YOLOv4-tiny. Use the test sets to test the performance of the four models. Compared to the original YOLOv4-tiny, the YOLOv4-tiny combined with the CBAM or ECA attention modules gives a certain increase in mAP, while the addition of SE produces some degree of decrease. It is feasible to use attention mechanisms for performance improvement of YOLOv4-tiny, but not every attention mechanism is suitable. In addition, the best YOLOv4-tiny based on CBAM with 57.2% mAP is insufficient in practice. In future work, we improve the detection for trunk and foot by expanding the trunk and foot samples in the self-built data sets, aiming to guarantee deployment performance while achieving high detection accuracy. Then, using the model with good performance, we convert the model trained by Pytorch into an open neural network exchange(ONNX) model for easy deployment.

Disclosures

The authors declare that they have no conflicts of interest.

Acknowledgments

I want to express my gratitude to the high-performance computing center of the University of Shanghai for Science and Technology.

References

Bhatt S, Soni H, Pawar T, Kher H. 2022. Diagnosis of pulmonary nodules on CT images using YOLOv4. *International Journal of Online and Biomedical Engineering* 18(5):131-146 doi: 10.3991/ijoe.v18i05.29529.

Bin Z, Sun CF, Fang SQ, Zhao YH, Su S. 2022. Workshop safety helmet wearing detection model based on SCM-YOLO. *Sensors* 22(17):6702 doi: 10.3390/s22176702.

Bochkovskiy A, Wang CY, Liao HYM. 2020. YOLOv4: optimal speed and accuracy of object detection. *arXiv preprint* arXiv: 2004.10934.

Furusho Y, Ikeda K. 2020. Theoretical analysis of skip connections and batch normalization from generalization and optimization perspectives. *Apsipa Transactions on Signal and Information Processing* 9:e9 doi: 10.1017/ATSIP.2020.7.

He KM, Zhang XY, Ren SQ, Sun J. 2015. Delving deep into rectifiers: surpassing human-level performance on ImageNet classification. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. Santiago, CHILE, 1026-1034 doi: 10.1109/ICCV.2015.123.

Hu J, Shen L, Sun G. 2018. Squeeze-and-excitation networks. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*. Salt Lake City, UT, 7132-7141 doi: 10.1109/CVPR.2018.00745.

Huang GQ, Wan ZN, Liu XG, Hui JP, Wang Z, Zhang ZY. 2019. Ship detection based on squeeze excitation skip-connection path networks for optical remote sensing images. *Neurocomputing* 332:215-223 doi: 10.1016/j.neucom.2018.12.050.

Javadi N, Soltanizadeh H. 2021. Automated detection, 3D position of facial skin lesions using genetic algorithm and Kinect camera. *Computer Methods in Biomechanics and Biomedical Engineering-Imaging and Visualization* 10(1):48-54 doi: 10.1080/21681163.2021.1972342.

Kim Y, Hwang I, Cho NI. 2017. Convolutional neural networks and training strategies for skin detection. In: *24th IEEE International Conference on Image Processing (ICIP)*. Beijing, PEOPLES R CHINA, 3919-3923.

Krizhevsky A, Sutskever I, Hinton GE. 2017. ImageNet classification with deep convolutional neural networks. *Communications of the ACM* 60(6), 84-90 doi: 10.1145/3065386.

Li P, Yu HL, Li SJ, Xu P. 2022. Comparative study of human skin detection using object detection based on transfer learning. *Applied Artificial Intelligence* 35(15):2370-2388 doi: 10.1080/08839514.2021.1997215.

Lin HY, Lin CJ, Jeng SY, Yu CY. 2021. Integrated image sensor and hyperparameter optimization of convolutional neural network for facial skin detection. *Sensors and Materials* 33(8):2911-2924 doi: 10.18494/SAM.2021.3301.

Lin TY, Dollar P, Girshick R, He KM, Hariharan B, Belongie S. 2017. Feature pyramid networks for object detection. In: *30th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI, 936-944 doi: 10.1109/CVPR.2017.106.

Liu Y, Soh LK, and Lorang E. 2021. Investigating coupling preprocessing with shallow and deep convolutional neural networks in document image classification. *Journal of Electronic Imaging* 30(4):043024 doi: 10.1117/1.JEI.30.4.043024.

Luo Y, Guan YP. 2017. Adaptive skin detection using face location and facial structure estimation. *IET Computer Vision* 11(7):550-559 doi: 10.1049/iet-cvi.2016.0295.

Misra D. 2019. Mish: a self regularized non-monotonic neural activation function. *arXiv Preprint* arXiv: 1908.08681.

Naing KM, Boonsang S, Chuwongin S, Kittichai V, Tongloy T, Prommongkol S, Dekumyoy P, Watthanakulpanich D. 2022. Automatic recognition of parasitic products in stool examination using object detection approach. *Peerj Computer Science* 8:e1065 doi: 10.7717/peerj-cs.1065.

Niu ZY, Zhong GQ, Yu H. 2022. A review on the attention mechanism of deep learning. *Neurocomputing* 452:48-62 doi: 10.1016/j.neucom.2021.03.091.

Pratondo A, Bramantoro A. 2022. Classification of zophobas morio and tenebrio molitor using transfer learning. *Peer Computer Science* 8:e884 doi: 10.7717/peerj-cs.884.

Redmon J, Divvala S, Girshick R, Farhadi A. 2016. You only look once: unified, real-time object detection. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, 779-788 doi: 10.1109/CVPR.2016.91.

Redmon J, Farhadi A. 2017. YOLO9000: better, faster, stronger. In: *30th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI, 6517-6525 doi: 10.1109/CVPR.2017.690.

Redmon J, Farhadi A. 2018. YOLOv3: an incremental improvement. *arXiv preprint* arXiv: 1804.02767.

Rezatofighi H, Tsoi N, Gwak JY, Sadeghian A, Reid I, Savarese S. 2019. Generalized intersection over union: a metric and a loss for bounding box regression. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA, 658-666 doi: 10.1109/CVPR.2019.00075.

Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang ZH, Karpathy A, Khosla A, Bernstein M, Berg AC, Li FF. 2015. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision* 115(3):211-252 doi: 10.1007/s11263-015-0816-y.

Salah KB, Othmani M, Kherallah M. 2022. A novel approach for human skin detection using convolutional neural network. *Visual Computer* 38(5):1833-1843 doi: 10.1007/s00371-021-02108-3.

Shifa A, Imtiaz MB, Asghar MN, Fleury M. 2020. Skin detection and lightweight encryption for privacy protection in real-time surveillance applications. *Image and Vision Computing* 94:103859 doi: 10.1016/j.imavis.2019.103859.

Sun HM. 2010. Skin detection for single images using dynamic skin color modeling. *Pattern Recognition* 43(4):1413-1420 doi: 10.1016/j.patcog.2009.09.022.

Tian SS, Chen ZX, Chen BL, Zou WB, Li X. 2021. Channel and spatial attention-based Siamese network for visual object tracking. *Journal of Electronic Imaging* 30(3):033008 doi: 10.1117/1.JEI.30.3.033008.

Wang QL, Wu BG, Zhu PF, Li PH, Zuo WM, Hu QH. 2020. ECA-Net: efficient channel attention for deep convolutional neural networks. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA, 11531-11539 doi: 10.1109/CVPR42600.2020.01155.

Woo SH, Park J, Lee JY, Kweon IS. 2018. CBAM: convolutional block attention module. *Computer Vision - ECCV 2018* 11211:3-19 doi: 10.1007/978-3-030-01234-2_1.

Zhang CB, Jiang PT, Hou QB, Wei YC, Han Q, Li Z, Cheng MM. 2021. Delving deep into label smoothing. *IEEE Transactions on Image Processing* 30:5984-5996 doi: 10.1109/TIP.2021.3089942.

Zhang K, Wang YD, Li WY, Li CL, Lei ZC. 2022. Real-time adaptive skin detection using skin color model updating unit in videos. *Journal of Real-time Image Processing* 19(2):303-315 doi: 10.1007/s11554-021-01186-9.

Zhao SJ, Zheng JC, Sun SD, Zhang L. 2022a. An improved YOLO algorithm for fast and accurate underwater object detection. *Symmetry-Basel* 14(8):1669 doi: 10.3390/sym14081669.

Zhao L, Zhang Q, Peng B, Yang L. 2022b. Real-time object detector for low-end devices. *Journal of Electronic Imaging* 31(1):013016 doi: 10.1117/1.JEI.31.1.013016.

Zheng ZH, Wang P, Liu W, Li JZ, Ye RG, Ren DW. 2019. Distance-IoU loss: faster and better learning for bounding box regression. *arXiv preprint* arXiv:1911.08287.

Zhou QK, Zhang W, Li RZ, Wang J, Zhen SH, Niu F. 2022. Improved YOLOv5-S object detection method for optical remote sensing images based on contextual transformer. *Journal of Electronic Imaging* 31(4):043049 doi: 10.1117/1.JEI.31.4.043049.

Figure 1

Example images in the data sets

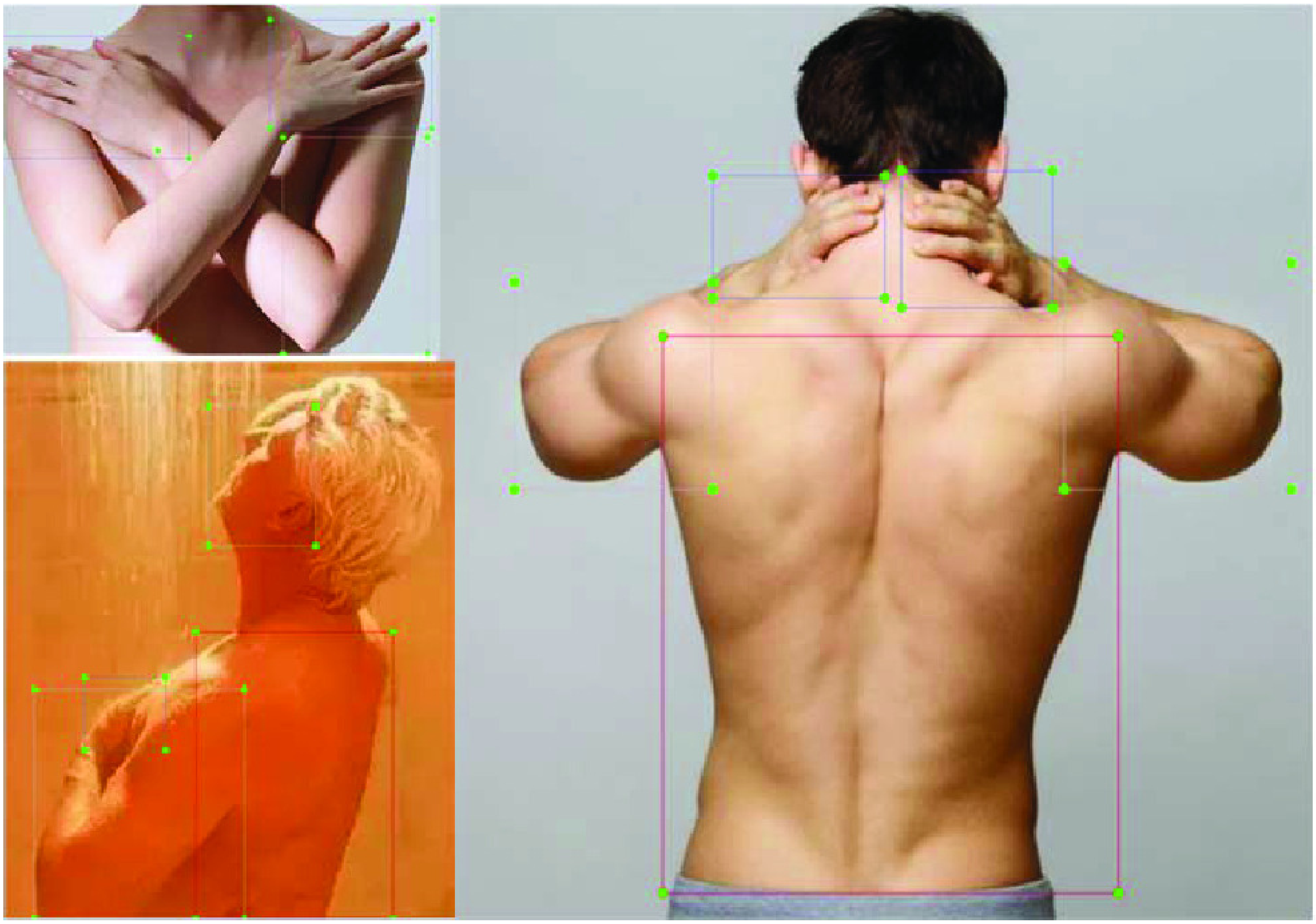

Figure 2

The structure of YOLOv4-tiny

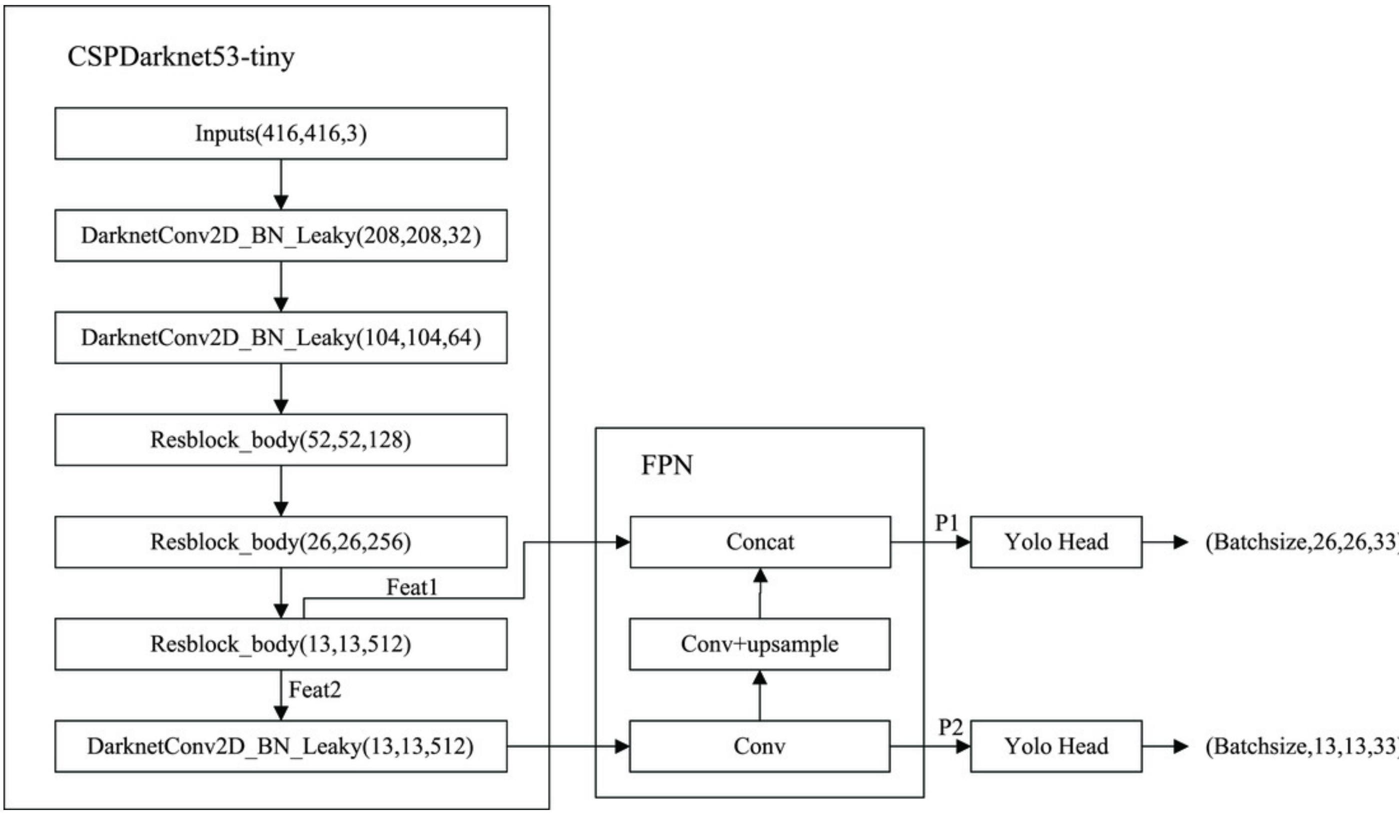

Figure 3

The structure of Resblock_body

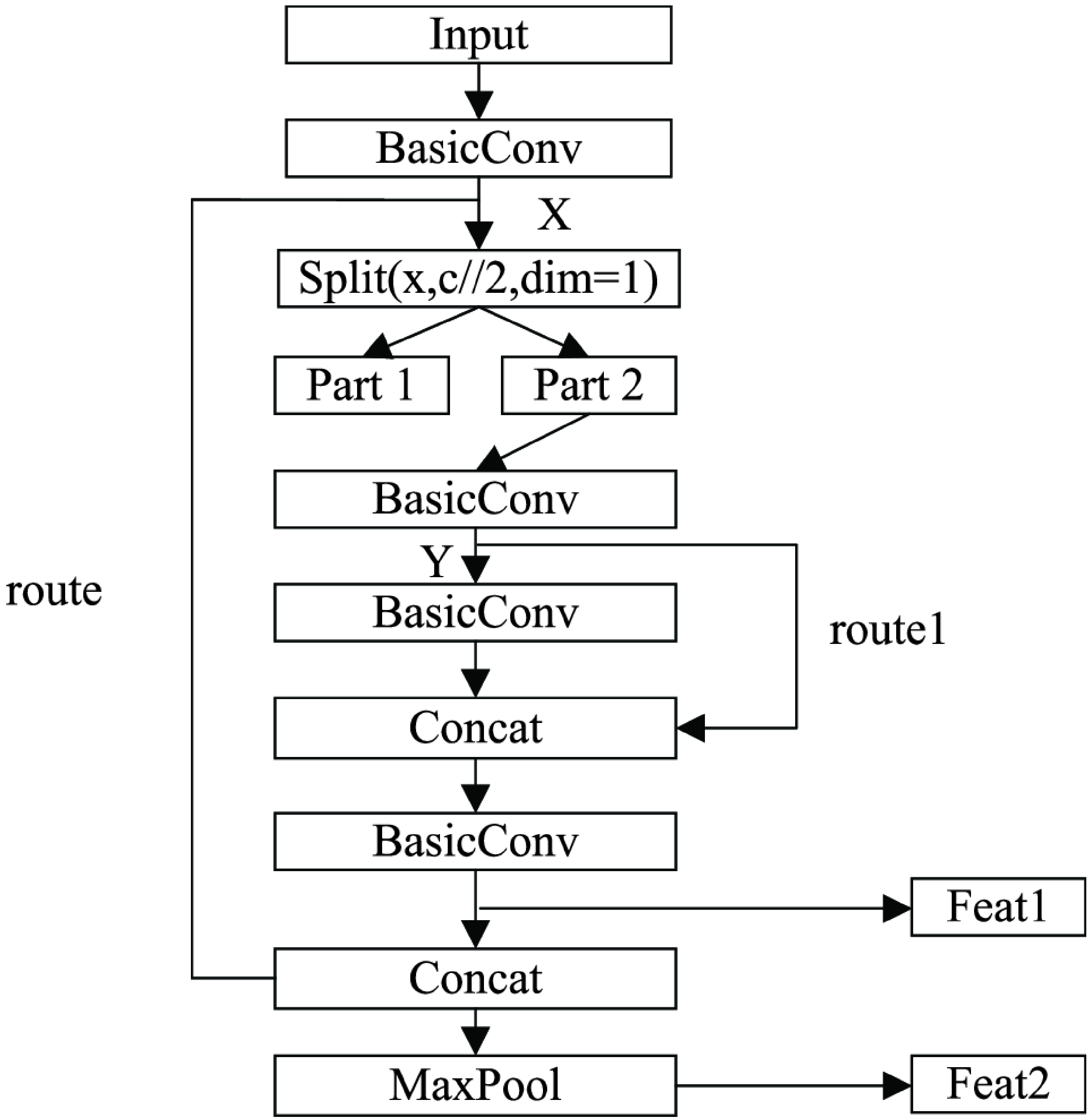

Figure 4

The specific implementation of SE

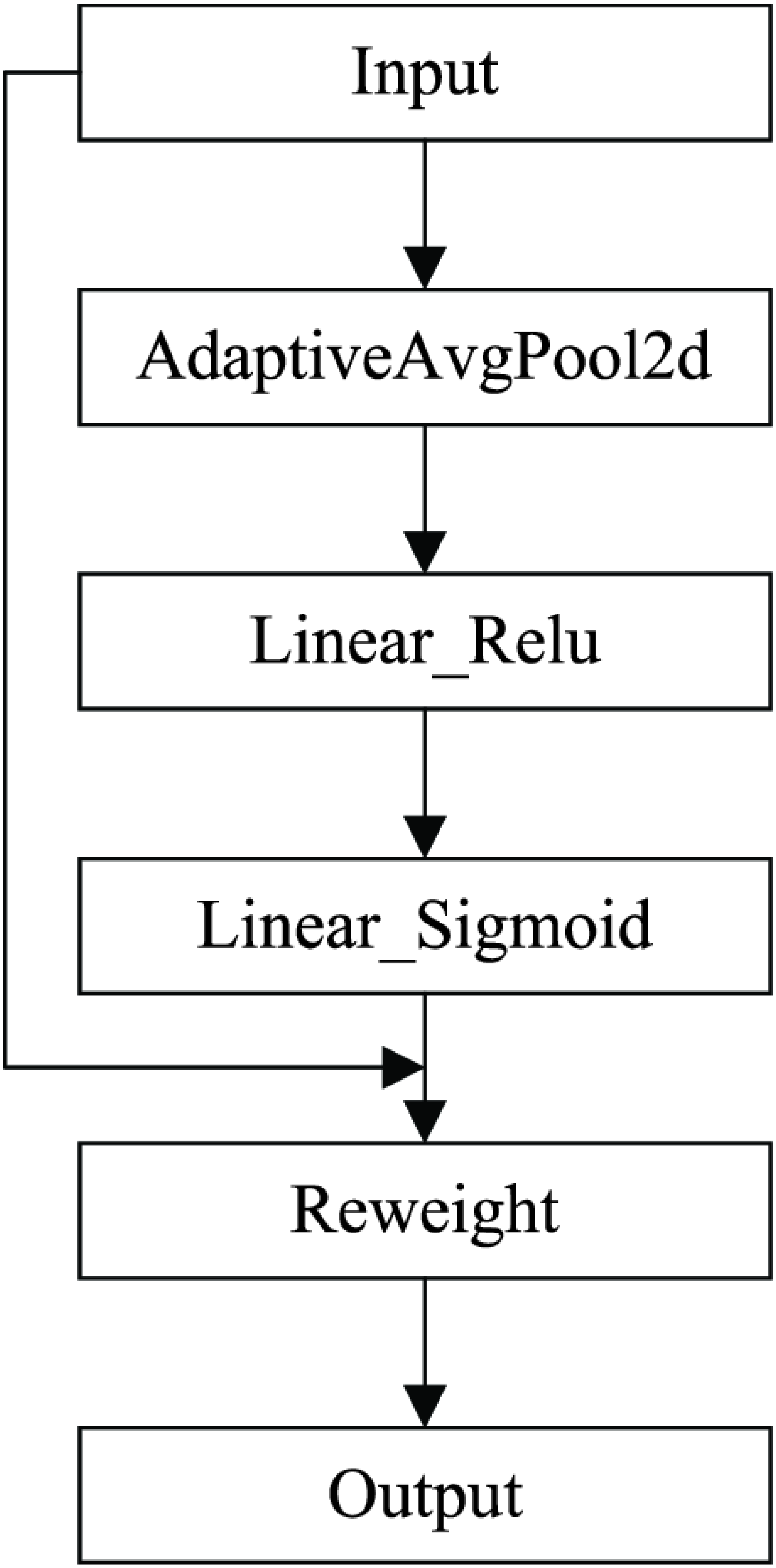
Input
AdaptiveAvgPool2d
Linear_Relu
Linear_Sigmoid
Reweight
Output

Figure 5

The specific implementation of ECA

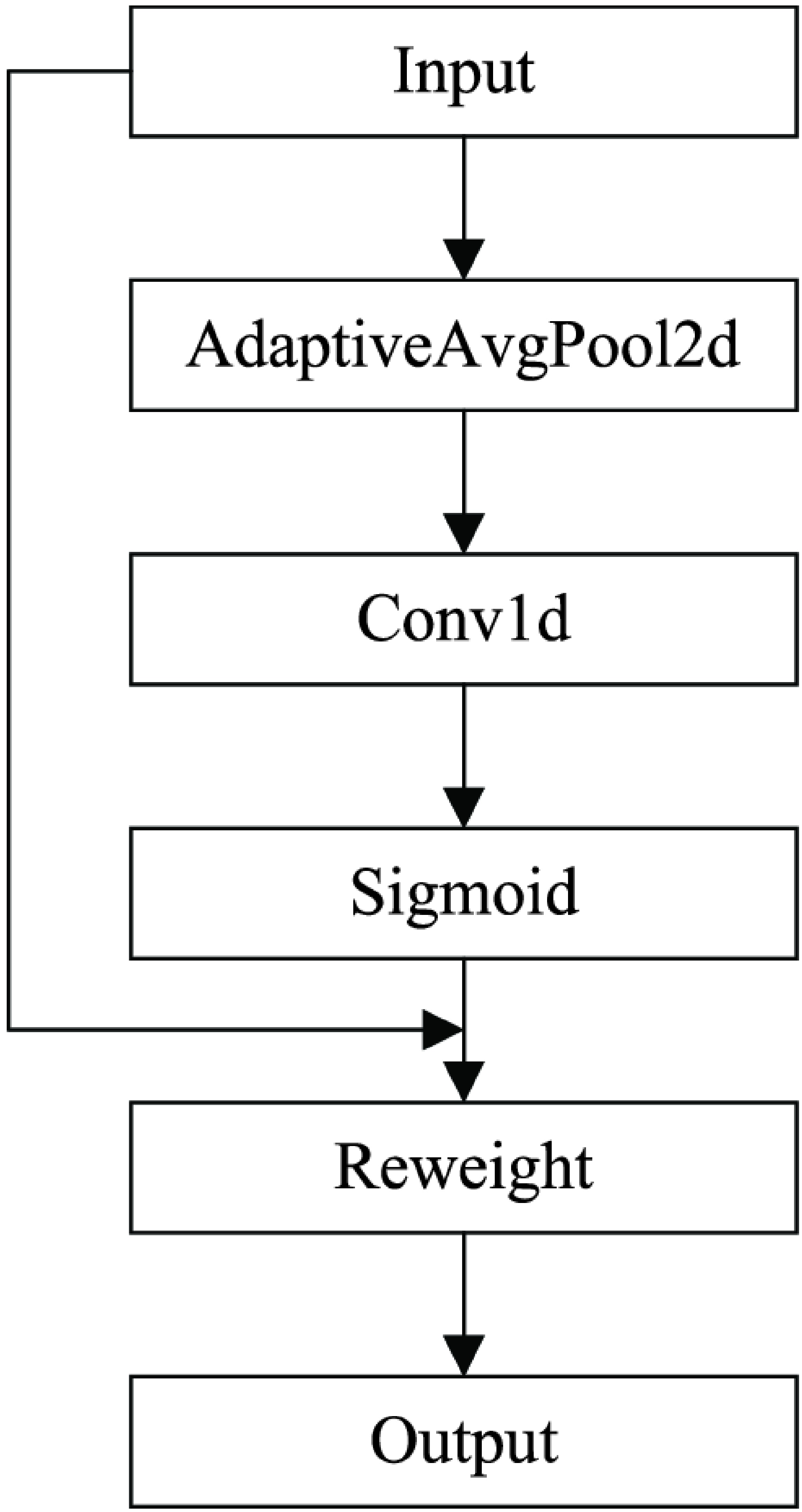
Input
AdaptiveAvgPool2d
Conv1d
Sigmoid
Reweight
Output

Figure 6

The specific implementation of CBAM

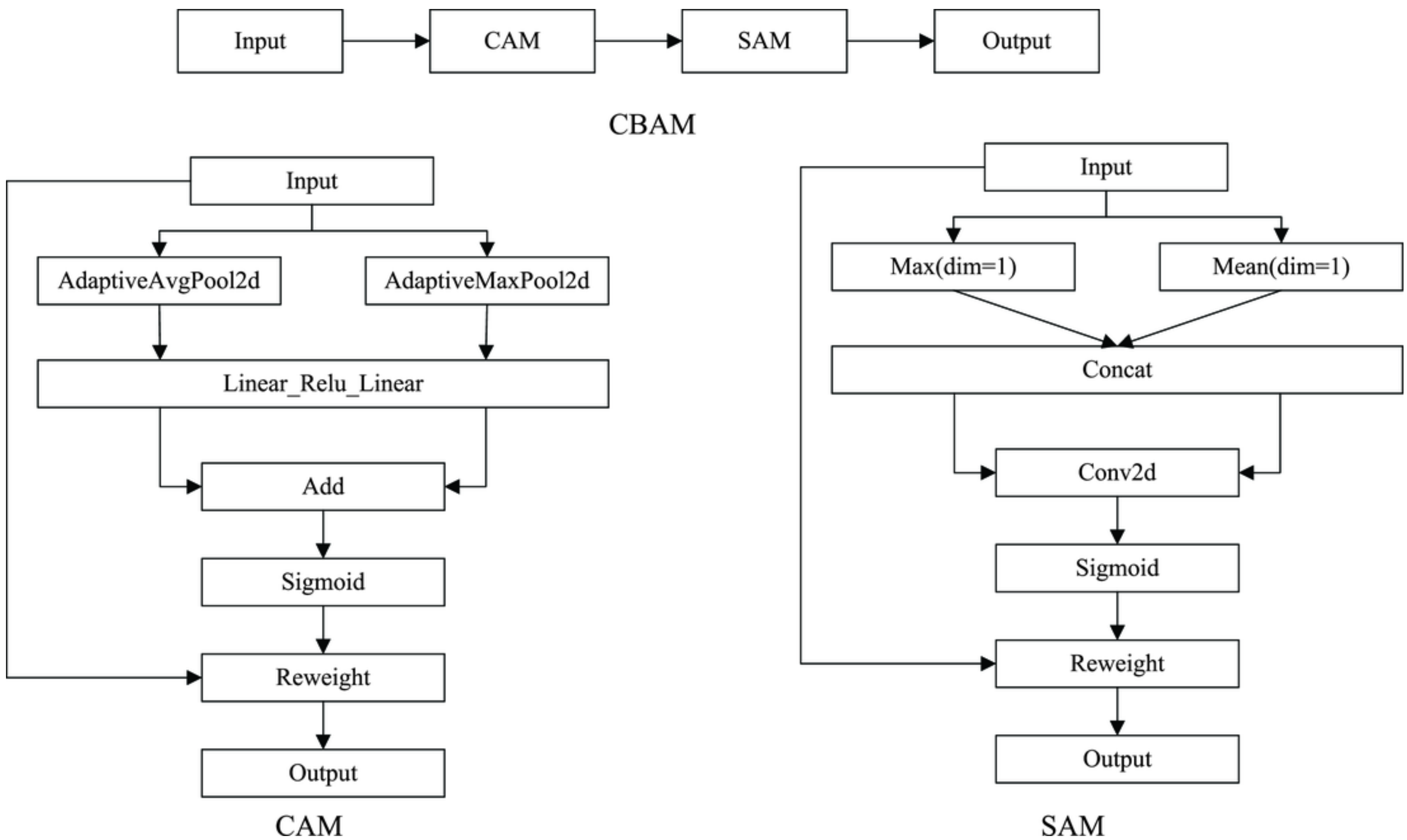

Figure 7

Improved YOLOv4-tiny based on attention mechanisms

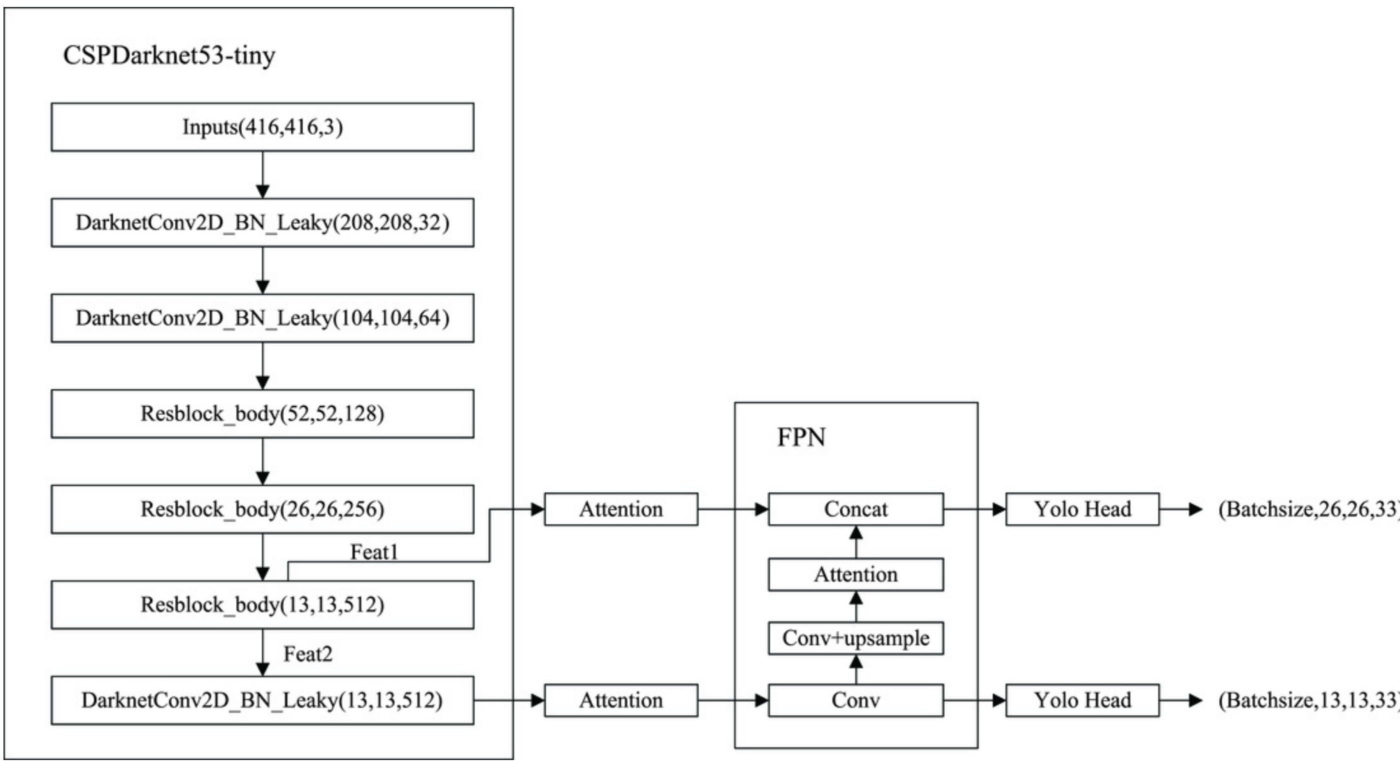

Figure 8

P-R curves

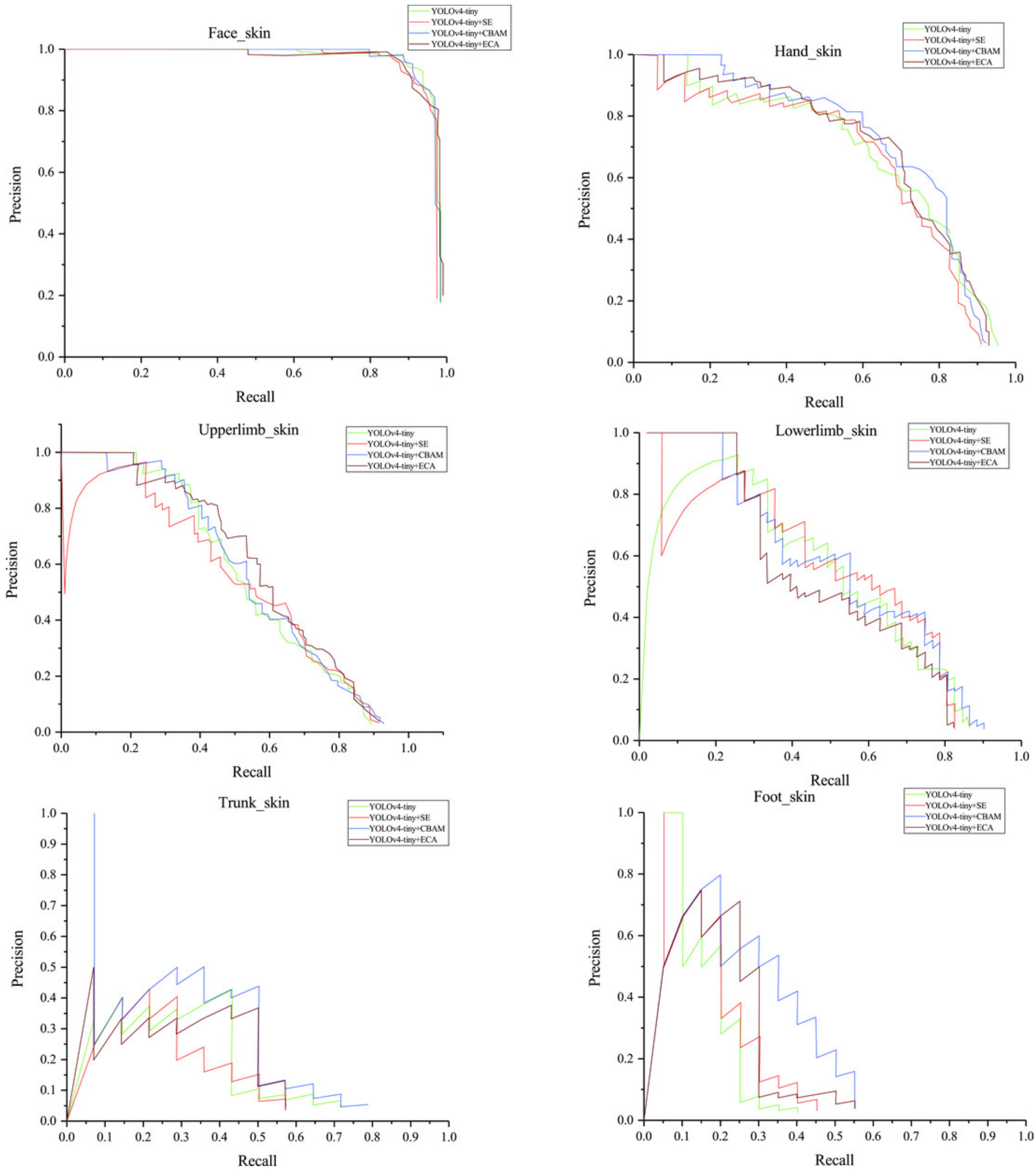
Face_skin
YOLOv4-tiny
YOLOv4-tiny+SE
YOLOv4-tiny+CBAM
YOLOv4-tiny+ECA
Precision
Recall
Hand_skin
YOLOv4-tiny
YOLOv4-tiny+SE
YOLOv4-tiny+CBAM
YOLOv4-tiny+ECA
Precision
Recall
Upperlimb_skin
YOLOv4-tiny
YOLOv4-tiny+SE
YOLOv4-tiny+CBAM
YOLOv4-tiny+ECA
Precision
Recall
Lowerlimb_skin
YOLOv4-tiny
YOLOv4-tiny+SE
YOLOv4-tiny+CBAM
YOLOv4-tiny+ECA
Precision
Recall
Trunk_skin
YOLOv4-tiny
YOLOv4-tiny+SE
YOLOv4-tiny+CBAM
YOLOv4-tiny+ECA
Precision
Recall
Foot_skin
YOLOv4-tiny
YOLOv4-tiny+SE
YOLOv4-tiny+CBAM
YOLOv4-tiny+ECA
Precision
Recall

Figure 9

The perception process in the bathing tasks: achieving the three-dimensional positioning of the target

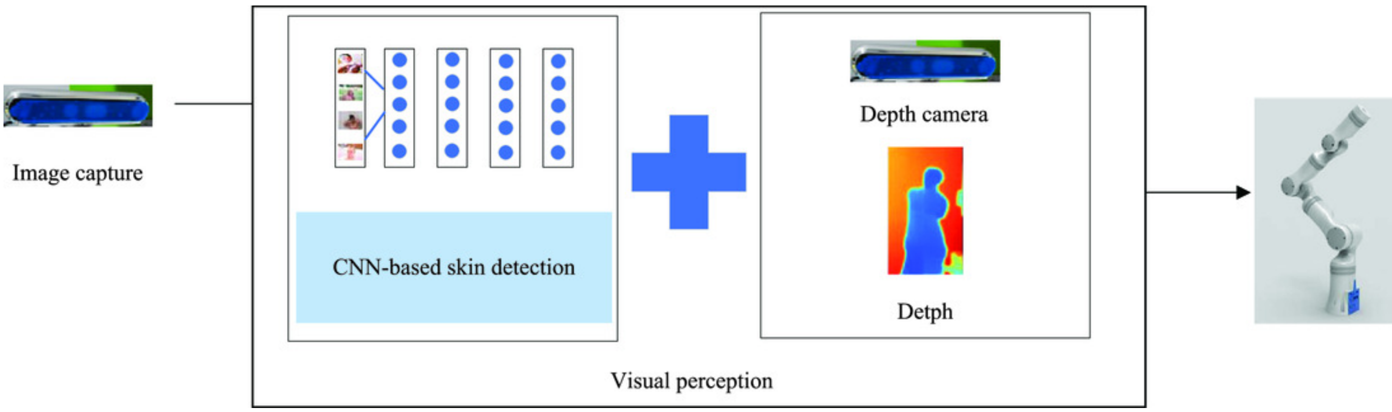

Table 1(on next page)

Summary of IOU, GIOU, DIOU, CIOU

	Features	Shortcomings
IOU	Representing the ratio of intersection and union of the GT box and the prediction box	When the prediction box and the GT box do not intersect, the loss function is not differentiable, leading losses cannot propagate
GIOU	scale invariant	Slow convergence speed and low positioning accuracy
DIOU	Overlapping area and center point distance are taken into account	Widely used in post-processing
CIOU	The consistency of aspect ratio is considered on the basis of DIOU	Widely used in post-processing

Table 2 (on next page)

Models information

Model	Attention	mAP	Weight file(MB)
YOLOv4-tiny	-	52.5%	22.4
SE_YOLOv4-tiny	SE	51.6%	22.6
CBAM_YOLOv4-tiny	CBAM	**57.2%**	22.8
ECA_YOLOv4-tiny	ECA	53.6%	22.4

Table 3 (on next page)

A, *B*, and *W* of all models

Models	Attention	A	B	W
SE_YOLOv4-tiny	SE	0.2	-0.9	-0.74
CBAM_YOLOv4-tiny	CBAM	0.4	4.7	**3.15**
ECA_YOLOv4-tiny	ECA	0	1.1	1.1

Table 4 (on next page)

AP values for the six categories

	Face_skin	Hand_skin	Upperlimb_skin	Lowerlimb_skin	Trunk_skin	Foot_skin
-	0.97	0.68	0.57	0.54	0.21	0.18
SE	0.96	0.67	0.55	0.55	0.17	0.21
CBAM	**0.97**	**0.72**	0.58	**0.56**	**0.3**	**0.3**
ECA	0.97	0.7	**0.6**	0.51	0.21	0.23