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]

Shanghai, China

Email address: yhl_usst@outlook.com

Abstract

Background. An automatic bathing robot needs to identify the area to be bathed in order to perform visually-guided bathing tasks. Skin detection is the first step. The deep convolutional neural network (CNN)-based object detection algorithm shows excellent robustness to light and environmental changes when performing skin detection. The one-stage object detection algorithm has good real-time performance, and is widely used in practical projects.

Methods. In our previous work, we performed skin detection using Faster R-CNN (ResNet50 as backbone), Faster R-CNN (MobileNetv2 as backbone), YOLOv3 (DarkNet53 as backbone), YOLOv4 (CSPDarknet53 as backbone), and CenterNet (Hourglass as backbone), and found that YOLOv4 had the best performance. In this study, we considered the convenience of practical deployment and used the lightweight version of YOLOv4, i.e., YOLOv4-tiny, for skin detection. Additionally, we added three kinds of attention mechanisms to strengthen feature extraction: SE, ECA, and CBAM. We added the attention module to the two feature layers of the backbone output. In the enhanced feature extraction network part, we applied the attention module to the up-sampled features. For full comparison, we used other lightweight methods that use MobileNetv1, MobileNetv2, and MobileNetv3 as the backbone of YOLOv4. We established a comprehensive evaluation index to evaluate the performance of the models that mainly reflected the balance between model size and mAP.

Results. The experimental results revealed that the weight file of YOLOv4-tiny without attention mechanisms was reduced to 9.2% of YOLOv4, but the mAP maintained 67.3% of YOLOv4. YOLOv4-tiny's performance improved after combining the CBAM and ECA modules, but the addition of SE deteriorated the performance of YOLOv4-tiny. MobileNetvX_YOLOv4 (X=1,2,3), which used MobileNetv1, MobileNetv2, and MobileNetv3 as the backbone of YOLOv4, showed higher mAP than YOLOv4-tiny series (including YOLOv4-tiny and three improved YOLOv4-tiny based on the attention mechanism) but had a larger weight file. The network performance was evaluated using the comprehensive evaluation index. The model, which integrates the CBAM attention mechanism and YOLOv4-tiny, achieved a good balance between model size and detection accuracy.

Introduction

A convolutional neural network (CNN) 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). By using massive data as learning samples, we can obtain a CNN model with analysis, feature representation, and recognition capabilities in order to achieve skin detection.

Skin detection is a prerequisite for bathing by automatic bathing robots. The intelligent bathing system detects human skin in the bathing environment based on vision sensors. Skin detection in bathing scenes is a challenging task. The bathing environment is full of water mist and 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. Handcrafted features are not sensitive to environmental changes and are insufficient for bathing scenarios. Skin detection based on machine learning, which generally uses supervised methods to construct detectors in order to extract skin features, is less influenced by environmental factors and has gained more applications in recent years. Salah et al. utilized CNN trained by skin and non-skin patches to detect skin pixels (Salah, Othmani & Kherallah, 2022). Kim et al. exploited two CNNs for skin detection and compared performance using different training strategies (Kim, Hwang & Cho, 2017). Lin et al. conducted CNN-based facial skin detection and optimized the CNN using the Taguchi method (Lin et al., 2021).

Instead of merely identifying skin and non-skin areas, we needed to provide the robot with information about specific skin areas (hands, feet, trunk, etc.) to clean up skin using different modes. We faced a multi-classification problem rather than a secondary classification problem. In application areas, one-stage object detection models based on CNN achieve good real-time performance and are computationally efficient. YOLO series are typical one-stage algorithms. YOLOv2, YOLOv3, YOLOv4, YOLOv5, and YOLOv7 are anchor-based algorithms that use anchors as the prior knowledge of the bounding box. YOLOv2 is not good at detecting small targets and uses Darknet19 as the backbone (Redmon & Farhadi, 2017). YOLOv3 adopts Darknet53 to extract features (Redmon & Farhadi, 2018). YOLOv4 uses CSPDarknet53 to extract features and uses SPP and PANet for feature fusion (Bochkovskiy, Wang, & Liao, 2020). The backbone and the neck parts of YOLOv5 include the CSP structure, and the Focus structure is proposed. Four models (YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x) are provided. They have different depths and widths. The accuracy continuously improves, but the speed consumption also increases (Xie, Lin, & Liu, 2022). YOLOv7 proposed the ELAN structure (Wang, Bochkovskiy, & Liao, 2022). YOLOv1, YOLOv6, and YOLOX are anchor-free algorithms that lack the prior information of the bounding box and have better scene generalization in theory. YOLOv1 is less effective for small and dense targets (Redmon et al., 2016). YOLOv6 adopts SPPF and Rep-PAN structures, whose backbone is mainly composed of

80   RepVGGBlock modules (Li et al., 2022). YOLOX includes standard versions (YOLOX-s,
81   YLOLX-m, YOLOX-l, YOLOX-x, and YOLOX-Darknet53) and lightweight versions
82   (YOLOX-Nano and YOLOX-Tiny) (Ge et al., 2021). YOLOX and YOLOv6 decouple the
83   regression and classification in the detection head.
84   Our research is based on previous work by our team (Li et al., 2021) that found that
85   YOLOv4 had a high mAP for skin detection in bath environments. The extensive computation
86   led YOLOv4 having a slow speed after being deployed to embedded devices. In the study, we
87   adopted YOLOv4-tiny (Zhao et al., 2022a), the lightweight model of YOLOv4, for skin
88   detection and investigated the effect of attention mechanisms on YOLOv4-tiny. For full
89   comparison, we also adopt other lightweight methods, namely, using MobileNetv1 (Howard et
90   al., 2017), MobileNetv2 (Sandler et al., 2018), and MobileNetv3 (Howard et al., 2019) as the
91   backbone of YOLOv4.
92   The remaining parts of the paper are arranged as follows: the "Materials & Methods"
93   section offers an introduction to data set acquisition, YOLOv4-tiny, improved YOLOv4-tiny,
94   transfer learning, experimental setup, and evaluation indicators. In the "Results" section we
95   describe the experimental results. In the "Discussion" section we discuss the results related to
96   our application. In the "Conclusion" section we summarize our research and look at future work.

Materials & Methods

Data sets acquisition

99   A total of 1,500 images of human skin were collected on the internet, and we considered factors
100   such as position, illumination, skin color, blurring, and the presence of water mist. The position
101   factor meant that the skin could appear in the middle or border of an image. In some images, the
102   skin area was occluded. The difference of illumination in the data sets provided robustness. The
103   data sets included people with fair skin, medium skin, and dark skin color. Our data sets included
104   both clear and blurred pictures. The blur degree needed to ensure that the skin area in a picture
105   was recognizable to the naked eye. In the data sets, some pictures included water mist and some
106   did not include water mist. Some images included all the areas of the human body, and other
107   images only included some areas. Ultimately, 1,000 images were selected based on image quality.
108   We counted the number of pictures using the above factors, and the results are shown in Table 1.
109   According to different regions, we divided skin into six categories: "Face_skin", "Trunk_skin",
110   "Upperlimb_skin", "Lowerlimb_skin", "Hand_skin", and "Foot_skin". The image annotation
111   tool LabelImg (Bhatt et al., 2022) was used to generate XML files corresponding to the images.
112   The XML file includes the file name, ground truth information, and category information.

YOLOv4-tiny

114   The structure of YOLOv4-tiny is shown in Fig. 1. The backbone is CSPDarknet53-tiny, which is
115   utilized for feature extraction. CSPDarknet53-tiny is composed of DarknetConv2D_BN_Leaky
116   modules and Resblock_body modules. A DarknetConv2D_BN_Leaky module combines a two-
117   dimensional convolutional layer, normalized processing layer, and activation function. The Mish
118   activation function (Misra, 2019) in the YOLOv4 is replaced by a Leaky Relu function (He et al.,
119   2015) to improve detection speed. The structure of Resblock_body is illustrated in Fig. 2. The

120   skip connection can better combine semantic information and let the model converge quickly,
121   preventing both model degradation and gradient disappearance (Furusho & Ikeda, 2020). Feat1
122   and Feat2 are the output feature layers from the Resblock_body module. The Feat2 output
123   branch of the first two Resblock_body modules is the input of the next module. FPN (Lin et al.,
124   2017) is used to enhance feature extraction and perform feature fusion to combine feature
125   information at different scales. For the output of the third Resblock_body module, Feat1 is
126   directly used as the first input of the FPN. The second input of the FPN is the result obtained by
127   processing Feat2 using the DarknetConv2D_BN_Leaky module. The output P2 of FPN is
128   obtained using convolution processing on the second input of the FPN. The output P1 of FPN is
129   obtained by stacking Feat1 and the result obtained using convolution and up-sampling operations
130   on P2. The structure of FPN is simple, allowing YOLOv4-tiny to have excellent real-time
131   performance. Compared with YOLOv4, YOLOv4-tiny has two detection heads and predicts at
132   two scales. The YOLO head is used to obtain classification and regression prediction results. The
133   structure of the YOLO head is straightforward. The two feature layers for prediction are acquired
134   using a small amount of convolution of P1 and P2. YOLOv4-tiny still makes the detection based
135   on anchors, using fixed-size anchors as a prior for object boxes, tiling many anchors on images,
136   and adjusting anchors to bounding boxes by the prediction results. "13×13" and "26×26"
137   represent the granularity of grids. "33" represents the prediction results adapted to our
138   application, i.e., 3×(4+1+6), where "3" represents the number of anchors, "4" indicates the
139   number of location parameters, "1" denotes the confidence score, and "6" is the number of
140   categories to be identified.
141   The loss function includes bounding box location loss L_{loc}, classification loss L_{cls}, and
142   confidence loss L_{conf}. The overall loss L is calculated as Eq. (1).

143
$$L = L_{loc} + L_{cls} + L_{conf} \ (1)$$

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

147
$$L_{loc} = 1 - IoU + \rho^2\left(b, b^{gt}\right)\big/ d^2 + \alpha \upsilon \ (2)$$

148
$$\upsilon = 4\big/\left(\pi^2\right) * \left(\arctan\left(w^{gt}/h^{gt}\right) - \arctan\left(w/h\right)\right)^2 \ (3)$$

149
$$\alpha = \upsilon\big/\left(\upsilon + 1 - IoU\right) \ (4)$$

150   Where $\rho^2(b, b^{gt})$ represents the European distance between the central points of the prediction box
151   and the GT box, d represents the diagonal distance of the minimum area enclosing the prediction
152   box and the GT box, α is weight, and υ expresses the consistency of the aspect ratio. υ and α are
153   calculated as demonstrated in Eq. (3) and Eq. (4).
154   L_{cls} measures the category error between the prediction box and the GT box, as shown in Eq.
155   (5). $K×K$ represents the number of grids on feature maps of different scales, and c represents the
156   category. If the j-th prior box of the i-th grid has objects to be predicted, $I_{ij}^{obj}=1$; otherwise,
157    $I_{ij}^{obj}=0$. $q_i(c)$ and $p_i(c)$ represent the actual value and predicted value of the probability that the j-

158 th prior box of the i-th grid belongs to category c, respectively. The L_{cls} is optimized using a label
159 smoothing approach to suppress the overfitting problem during training (Zhang et al., 2021). $q_i(c)$
160 is expressed as Eq. (6) where $y_{true}(c)$ represents the one-hot hard label, ε is a constant, and N
161 represents the total number of categories.

162
$$L_{cls} = -\sum_{i=0}^{K \times K} I_{ij}^{obj} \sum_{c \in classes} \left[q_i(c) \log(p_i(c)) + (1 - q_i(c)) \log(1 - p_i(c)) \right] \quad (5)$$

$$q_i(c) = (1 - \varepsilon) y_{true}(c) + \frac{\varepsilon}{N} \quad (6)$$
163

164 L_{conf} adopts a cross-entropy loss function, as shown in Eq. (7). M represents the number of
165 prior boxes. D_i and C_i represent the actual and predicted values of confidence. If the j-th prior
166 box of the i-th grid has no object to be predicted, $I_{ij}^{noobj}=1$; otherwise, $I_{ij}^{noobj}=0$.

167
$$L_{conf} = \sum_{i=0}^{K \times K} \sum_{j=0}^{M} I_{ij}^{obj} \left[D_i \log(C_i) + (1 - D_i) \log(1 - C_i) \right] - \sum_{i=0}^{K \times K} \sum_{j=0}^{M} I_{ij}^{noobj} \left[D_i \log(C_i) + (1 - D_i) \log(1 - C_i) \right] (7)$$

168 **Improved YOLOv4-tiny based on attention mechanisms**
169 The attention mechanism has a variety of implementations (Niu, Zhong, & Yu, 2022). The core
170 of the attention mechanism is to make the network pay attention to needed areas. In general,
171 attention mechanisms can be divided into the channel attention mechanism, the spatial attention
172 mechanism, and a combination of the two (Tian et al., 2021). In this paper, the following
173 attention mechanisms were used:
174   (1) Squeeze-and-excitation (SE) (Hu, Shen, & Sun, 2018). SE is a typical implementation of
175 the channel attention mechanism that obtains the weights of each channel in the feature maps.
176 The inter-dependencies among channels are modeled explicitly. Instead of introducing a new-
177 built spatial dimension for the fusion of feature channels, SE uses a feature rescaling strategy.
178 Specifically, the importance of each channel is acquired spontaneously by self-learning. SE
179 includes squeeze and excitation operations. The squeeze operation conducts feature compression
180 across the spatial dimension, and converts a two-dimensional feature map into a real number that
181 owns a global receptive field. The output size matches the number of input channels. The
182 excitation operation is equivalent to the mechanics of gates in recurrent neural networks, where
183 weights are created for each channel employing learned parameters, and explicitly models the
184 correlation between feature channels. Finally, the weights, which are output by excitation
185 operations, represent the importance of each channel. The rescaling of features in the channel
186 dimension is accomplished by multiplying the weights by features of each channel (Huang et al.,
187 2019). The specific implementation of SE is shown in Fig. 3.
188   (2) Efficient channel attention (ECA). ECA is an improved version of SE. Wang et al.
189 argued that seizing all channel dependencies is ineffective and unessential for the SE block
190 (Wang et al., 2020). Convolution operation owns the cross-channel information capture
191 capability. ECA removes the fully connected layer of SE and learns weights by 1D convolution
192 operation on the globally averaged pooled features. The specific implementation of ECA is
193 shown in Fig. 4.

194   (3) Convolutional block attention module (CBAM). CBAM (Woo et al., 2018) performs
195   channel attention and spatial attention mechanism processing for feature maps, as shown in Fig.
196   5. The implementation of the channel attention module (CAM) can be divided into two parts.
197   Global average pooling and maximum global pooling are performed separately for the input
198   feature maps. The outputs are processed using a shared, fully connected layer. Sum the two
199   processed results, and take the sigmoid operation to obtain the weights of each channel of the
200   input features. The weights are multiplied by the original input features to get the output of CAM.
201   The spatial attention module (SAM) takes the maximum and average value on each channel of
202   each feature point. The two results are stacked. Adjust the number of channels using a
203   convolution operation. Determine the weights of each feature point of the input features using
204   the sigmoid function. Obtain the output by multiplying the weights by the original input features.
205   In this study, the above attention mechanisms are applied to YOLOV4-tiny. As shown in
206   Fig. 6, we added attention mechanisms on the two feature layers extracted from the backbone
207   network and attention mechanisms on the up-sampled results in FPN.

208   **Transfer learning**
209   Training a network from scratch requires an enormous amount of labeled data. Manual labeling
210   of data sets is time-consuming and labor-intensive, which introduces the possibility of human
211   error. Small data sets combined with transfer learning techniques can quickly produce a desirable
212   model (Pratondo & Bramantoro, 2022). The ImageNet contains more than 14 million images
213   covering more than 20,000 categories, of which more than one million images have explicit
214   annotations and corresponding labels at objects' locations in the image (Russakovsky et al.,
215   2015). The pre-trained models on ImageNet can learn fundamental features such as textures and
216   lines, which are general in object detection. All models use the pre-trained weights on the
217   ImageNet as the initial weights in this study.

218   **Experimental setup and evaluation indicators**
219   For the fairness of model comparison, we used the same data sets as our previous work (Li et al.,
220   2021), with a ratio of 60%:20%:20% for the training, validation, and test sets. All models were
221   trained with the help of the high-performance computing center of the University of Shanghai for
222   Science and Technology. Mosaic data augmentation was used in the training process in which
223   four randomly stitched images were input to the network for training to increase the background
224   diversity (Bin et al., 2022). In order to compare with other lightweight methods, we replaced the
225   backbone of YOLOv4 with MobileNetv1, MobileNetv2, and MobileNetv3 to obtain three
226   network architectures. We used the Pytorch framework for model building and training. The
227   initial value of the learning rate was set to 0.001 and the decay rate was set to 0.01. The batch
228   size was set to 16, which indicated the number of images input to the model for training every
229   time. SGD was utilized as the optimizer for model training. When training, the weights of the
230   backbone were frozen first for 50 epochs, and all weights were trained after 50 epochs, which
231   increased the convergence speed and training performance of models.
232   Recall and precision can be used to measure performance but are not fully representative of
233   detector quality. Many sets of recall and precision values are obtained by taking different

234   thresholds. Plot a P-R curve (Naing et al., 2022). AP characterized the area enclosed by the P-R
235   curve and the coordinate axes. The sum of the AP values of all classes was then divided using
236   the total number of classes to get mAP, which is the crucial evaluation metric of detectors for
237   multiple category detection. Our research is application oriented. In addition to the mAP
238   indicators, we also focused on the model size. The smaller model and the higher mAP were more
239   desirable for our embedded applications.

240   **Results**

241   After the training was completed, models were selected based on the results of the validation sets
242   and the performance was tested using the test sets. The mAPs and weight file information of
243   models are exhibited in Table 3.

244   In our previous work, the mAP of YOLOv4 reached 78%, as shown in Table 4. Also, it had
245   a weight file of 244 MB. After the light-weighting process, the mAP of YOLOv4-tiny was 67.3%
246   of YOLOv4, but the weight file was reduced to 9.2% of YOLOv4. Based on YOLOv4-tiny, we
247   added attention mechanisms as shown in Fig. 6. As can be seen in Table 3, mAP is reduced by
248   0.9% after adding the SE. There was a 1.1% improvement in mAP after adding the ECA. The
249   mAP increased by nearly 5% with the addition of CBAM. After adding ECA, the weight file
250   hardly increased. After adding SE, the weight file increased by 0.2 M. After adding CBAM, the
251   weight file increased by 0.4 M. MobileNetv1_YOLOv4, MobileNetv2_YOLOv4, and
252   MobileNetv3_YOLOv4 represent the model obtained using MobileNetv1, MobileNetv2, and
253   MobileNetv3 as the backbone of YOLOv4. Table 3 shows that the maximum weight file of
254   YOLOv4-tiny series (including YOLOv4-tiny and three improved YOLOv4-tiny based on
255   attention mechanism) was 22.8MB, and the minimum weight file of MobileNetvX_ YOLOv4
256   (X=1,2,3) was 46.5MB. Also, the maximum mAP of YOLOv4-tiny series was 57.2%, and the
257   minimum mAP of MobileNetvX_ YOLOv4 (X=1,2,3) was 60.2%. Overall, the mAP of
258   MobileNetvX_ YOLOv4 (X=1,2,3) was higher than that of YOLOv4-tiny series, but the weight
259   file of MobileNetvX_ YOLOv4 (X=1,2,3) was bigger than that of the YOLOv4-tiny series. The
260   AP values for the six categories are shown in Table 5. CBAM_YOLOv4-tiny achieved the
261   highest AP values for the face and foot, MobileNetv3_ YOLOv4 achieved the highest AP values
262   for the hand and lower limb, MobileNetv2_ YOLOv4 achieved the highest AP values for the
263   upper limb, and MobileNetv1_ YOLOv4 achieved the highest AP value for the trunk. P-R curves
264   are shown in Fig. 7.

265   In embedded applications, we believe that model quality was not only related to mAP, but
266   also to model size. We hoped to achieve a better balance between model size and mAP. The size
267   of the weight file could reflect the model size to some extent. Based on the above analysis, a
268   comprehensive indicator W was established to describe the balance, as shown in Eq. (8), where
269    $A=U(i)-U(0)$ and $B=V(i)-V(0)$. $U(0)$ represents the weight file size of the original YOLOv4-tiny,
270    $U(i)$ represents the weight file size of the other model, $V(0)$ represents the mAP of the original
271   YOLOv4-tiny, and $V(i)$ represents the mAP of the other model. The smaller the A is, the smaller
272   the model is. The larger the B is, the better mAP the model has. Intuitively, establishing $W=B/A$
273   can ensure that the larger the W is, the better the balance between mAP and model size.

274   Considering that $A \geq 0$ ($U(i) \geq U(0)$ seen from Table 3) and $y = e^x$ is a monotonically increasing

275   function, we finally used e^A instead of A, which can avoid the situation where the denominator

276   equals 0 and can ensure that W decreases with the increase of A. Based on the above information,

277   W can depict the balance between mAP and model size. After calculation, A, B, and W are

278   indicated in Table 6. W of CBAM_YOLOv4-tiny is highest and CBAM_YOLOv4-tiny achieved

279   the best balance.

280
$$W = \frac{B}{e^A} (A \geq 0) \quad (8)$$

281   **Discussion**

282   To perform the bathing tasks, we needed to recognize the area that needed to be bathed in the

283   bathing scenario and send the recognition information to the bathing robot arm for bathing

284   behavior planning, as shown in Fig. 8. By combining the skin detection results of 2D images

285   with the depth information obtained from the depth camera, we could model the localization of

286   targets in 3D space. In order to facilitate the robot to implement distinct bathing patterns for

287   areas of the body, we needed to identify the skin located at diverse parts of the body. Therefore,

288   we built small data sets in the bathing scenarios to be used as learning samples for object

289   detection models. The manual annotation was performed with the labelImg tool to classify skin

290   regions into six categories according to different parts.

291   Among the object detection algorithms, one-stage detection algorithms are faster than two-

292   stage and are suitable for application in our scenario where real-time performance is required. In

293   our previous work, we explored the effectiveness of object detection models for skin detection

294   with multiple classifications and found the best YOLOv4 model from five models. For easy

295   deployment, we used lightweight YOLOv4-tiny and imposed three kinds of attention

296   mechanisms on YOLOv4-tiny. We found that both CBAM and ECA improved the detection

297   effect, yet SE made the detection effect worse instead, which implies that we need to carefully

298   choose the attention mechanism during practice. We also used another lightweight method that

299   replaced the backbone of YOLOv4 with MobileNetv1, MobileNetv2, and MobileNetv3. They

300   obtained a higher mAP than YOLOv4-tiny series and were accompanied by a larger weight file.

301   Compared with Salah's work, our data sets included images with six types of labels for

302   network training instead of skin and non-skin patches. We were not only able to discern between

303   skin and non-skin patches, but also the body part. To the best of our knowledge, this is the first

304   time a study on skin detection that has been able to identify different body parts. The innovation

305   of this paper is in the use of CNN-based object detection algorithms to complete regional skin

306   detection for bathing tasks with embedded applications.

307   When developing assistive robots, visual information is frequently used, especially since

308   there is are a number of image processing algorithms. Our application scenarios are special and,

309   fortunately, we also considered privacy protection. Mitigating damage to privacy mainly starts

310   when: (1) the data are not stored locally and the whole system does not display RGB information

311   (our hardware computing platform uses jetson TX2, which has limited storage space and does

312   not support the storage of visual data in the bath scene); (2) the whole system is not connected to

313 the network, ensuring that the data do not have the risk of transmission; (3) the data are
314 processed on TX2 rather than transmitted to the cloud for processing. In fact, we hoped to find
315 the two-dimensional pixel coordinate information with the help of a mature algorithm using
316 RGB and combine the depth information of the depth camera to model object regions in three-
317 dimensional space. In addition, we mainly applied the bathing robot to semi-disabled elderly
318 people who generally hope to complete bathing independently of nursing staff. The bathing robot
319 can provide the elderly with the opportunity to take care of themselves while bathing. Compared
320 with relying on nursing staff to complete bathing tasks, using the bathing robot can maintain the
321 dignity of the elderly to the greatest extent.
322 There has been little research on object detection-based skin detection combined with
323 robotic arms for bathing tasks. Our study explored the YOLOv4-tiny and which attention
324 mechanism works best on YOLOv4-tiny. However, the YOLOv4-tiny showed a reduction in
325 mAP compared with the YOLOv4, creating some challenges for high detection accuracy (Zhao
326 et al., 2022b). The relatively small number of trunks in the data sets resulted in the poor detection
327 of trunks. The foot occupies a small area in the whole-body range. Foot features tend to
328 disappear with repeated down-sampling operations, resulting in poor detection of the foot.

Conclusion

330 When using robots for autonomous bathing tasks, skin detection needs to be accomplished first.
331 To facilitate the embedded deployment, we used YOLOv4-tiny, a lightweight model of
332 YOLOv4, for skin detection based on our previous work. Three kinds of attention mechanisms
333 were overlaid in the YOLOv4-tiny, and we used the test sets to test the performance of models.
334 Compared to the original YOLOv4-tiny, the YOLOv4-tiny combined with the CBAM or ECA
335 attention modules showed a certain increase in mAP, while the addition of SE produced some
336 degree of decrease. It is feasible to use attention mechanisms for performance improvement of
337 YOLOv4-tiny, but not every attention mechanism is suitable. Compared with the lightweight
338 method of using MobileNetv1/MobileNetv2/MobileNetv3 as the backbone of YOLOv4, the
339 method of using YOLOv4-tiny and CBAM achieves a better balance between model size and
340 detection effect. In future work, we will improve the detection for trunk and foot by expanding
341 the trunk and foot samples in the self-built data sets, and aim to guarantee deployment
342 performance while achieving high detection accuracy. Then, we will convert the best model into
343 an open neural network exchange model for easy deployment.

Acknowledgments

345 We want to express our gratitude to the high-performance computing center of the University of
346 Shanghai for Science and Technology.

347

References

349 Bhatt S, Soni H, Pawar T, Kher H. 2022. Diagnosis of pulmonary nodules on CT images using
350 YOLOv4. *International Journal of Online and Biomedical Engineering* 18(5):131-146 doi:
351 10.3991/ijoe.v18i05.29529.

352 Bin Z, Sun CF, Fang SQ, Zhao YH, Su S. 2022. Workshop safety helmet wearing detection
353 model based on SCM-YOLO. *Sensors* 22(17):6702 doi: 10.3390/s22176702.
354 Bochkovskiy A, Wang CY, Liao HYM. 2020. YOLOv4: optimal speed and accuracy of object
355 detection. *arXiv preprint* arXiv: 2004.10934.
356 Furusho Y, Ikeda K. 2020. Theoretical analysis of skip connections and batch normalization
357 from generalization and optimization perspectives. *Apsipa Transactions on Signal and*
358 *Information Processing* 9:e9 doi: 10.1017/ATSIP.2020.7.
359 Ge Z, Liu S, Wang F, Li Z, Sun J. 2021. YOLOX: Exceeding YOLO Series in 2021. *arXiv*
360 *preprint* arXiv: 2107.08430.
361 He KM, Zhang XY, Ren SQ, Sun J. 2015. Delving deep into rectifiers: surpassing human-level
362 performance on ImageNet classification. In: *2015 IEEE International Conference on Computer*
363 *Vision (ICCV)*. Santiago, CHILE, 1026-1034 doi: 10.1109/ICCV.2015.123.
364 Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H.
365 2017. Mobilenets: efficient convolutional neural networks for mobile vision applications. *arXiv*
366 *preprint* arXiv: 1704.04861.
367 Howard A, Sandler M, Chu G, Chen LC, Chen B, Tan MX, Wang WJ, Zhu YK, Pang RM,
368 Vasudevan V, Le QV, Adam H. 2019. In: 2019 IEEE International Conference on Computer
369 Vision (ICCV). Seoul, SOUTH KOREA, 1314-1324 doi: 10.1109/ICCV.2019.00140.
370 Hu J, Shen L, Sun G. 2018. Squeeze-and-excitation networks. In: *2018 IEEE/CVF Conference*
371 *on Computer Vision and Pattern Recognition(CVPR)*. Salt Lake City, UT, 7132-7141 doi:
372 10.1109/CVPR.2018.00745.
373 Huang GQ, Wan ZN, Liu XG, Hui JP, Wang Z, Zhang ZY. 2019. Ship detection based on
374 squeeze excitation skip-connection path networks for optical remote sensing images.
375 *Neurocomputing* 332:215-223 doi: 10.1016/j.neucom.2018.12.050.
376 Kim Y, Hwang I, Cho NI. 2017. Convolutional neural networks and training strategies for skin
377 detection. In: *24th IEEE International Conference on Image Processing (ICIP)*. Beijing,
378 PEOPLES R CHINA, 3919-3923.
379 Krizhevsky A, Sutskever I, Hinton GE. 2017. ImageNet classification with deep convolutional
380 neural networks. *Communications of the ACM* 60(6), 84-90 doi: 10.1145/3065386.
381 Li C, Li L, Jiang H, Weng K, Geng Y, Li L, Ke Z, Li Q, Cheng M, Nie W, Li Y, Zhang B, Liang
382 Y, Zhou L, Xu X, Chu X, Wei X, Wei X. 2022. YOLOv6: a single-stage object detection
383 framework for industrial applications. *arXiv preprint* arXiv: 2209.02976.
384 Li P, Yu HL, Li SJ, Xu P. 2021. Comparative study of human skin detection using object
385 detection based on transfer learning. *Applied Artificial Intelligence* 35(15):2370-2388 doi:
386 10.1080/08839514.2021.1997215.
387 Lin HY, Lin CJ, Jeng SY, Yu CY. 2021. Integrated image sensor and hyperparameter
388 optimization of convolutional neural network for facial skin detection. *Sensors and Materials*
389 33(8):2911-2924 doi: 10.18494/SAM.2021.3301.

390 Lin TY, Dollar P, Girshick R, He KM, Hariharan B, Belongie S. 2017. Feature pyramid
391 networks for object detection. In: *30th IEEE/CVF Conference on Computer Vision and Pattern*
392 *Recognition (CVPR)*. Honolulu, HI, 936-944 doi: 10.1109/CVPR.2017.106.
393 Liu Y, Soh LK, and Lorang E. 2021. Investigating coupling preprocessing with shallow and deep
394 convolutional neural networks in document image classification. *Journal of Electronic Imaging*
395 30(4):043024 doi: 10.1117/1.JEI.30.4.043024.
396 Misra D. 2019. Mish: a self regularized non-monotonic neural activation function. *arXiv*
397 *Preprint* arXiv: 1908.08681.
398 Naing KM, Boonsang S, Chuwongin S, Kittichai V, Tongloy T, Prommongkol S, Dekumyoy P,
399 Watthanakulpanich D. 2022. Automatic recognition of parasitic products in stool examination
400 using object detection approach. *Peerj Computer Science* 8:e1065 doi: 10.7717/peerj-cs.1065.
401 Niu ZY, Zhong GQ, Yu H. 2022. A review on the attention mechanism of deep learning.
402 *Neurocomputing* 452:48-62 doi: 10.1016/j.neucom.2021.03.091.
403 Pratondo A, Bramantoro A. 2022. Classification of zophobas morio and tenebrio molitor using
404 transfer learning. *Peer Computer Science* 8:e884 doi: 10.7717/peerj-cs.884.
405 Redmon J, Divvala S, Girshick R, Farhadi A. 2016. You only look once: unified, real-time object
406 detection. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
407 Seattle, WA, 779-788 doi: 10.1109/CVPR.2016.91.
408 Redmon J, Farhadi A. 2017. YOLO9000: better, faster, stronger. In: *30th IEEE/CVF Conference*
409 *on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI, 6517-6525 doi:
410 10.1109/CVPR.2017.690.
411 Redmon J, Farhadi A. 2018. YOLOv3: an incremental improvement. *arXiv preprint* arXiv:
412 1804.02767.
413 Rezatofighi H, Tsoi N, Gwak JY, Sadeghian A, Reid I, Savarese S. 2019. Generalized
414 intersection over union: a metric and a loss for bounding box regression. In: *2019 IEEE/CVF*
415 *Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA, 658-
416 666 doi: 10.1109/CVPR.2019.00075.
417 Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang ZH, Karpathy A, Khosla A,
418 Bernstein M, Berg AC, Li FF. 2015. ImageNet large scale visual recognition challenge.
419 *International Journal of Computer Vision* 115(3):211-252 doi: 10.1007/s11263-015-0816-y.
420 Salah KB, Othmani M, Kherallah M. 2022. A novel approach for human skin detection using
421 convolutional neural network. *Visual Computer* 38(5):1833-1843 doi: 10.1007/s00371-021-
422 02108-3.
423 Sandler M, Howard A, Zhu ML, Zhmoginov A, Chen LC. 2018. MobileNetV2: Inverted
424 residuals and linear bottlenecks. In: 2018 IEEE/CVF Conference on Computer Vision and
425 Pattern Recognition (CVPR). Salt Lake City, UT, 4510-4520 doi: 10.1109/CVPR.2018.00474.
426 Tian SS, Chen ZX, Chen BL, Zou WB, Li X. 2021. Channel and spatial attention-based Siamese
427 network for visual object tracking. *Journal of Electronic Imaging* 30(3):033008 doi:
428 10.1117/1.JEI.30.3.033008.

429 Wang CY, Bochkovskiy A, Liao HYM. 2022. YOLOv7: Trainable bag-of-freebies sets new
430 state-of-the-art for real-time object detectors. *arXiv preprint* arXiv:2207.02696.
431 Wang QL, Wu BG, Zhu PF, Li PH, Zuo WM, Hu QH. 2020. ECA-Net: efficient channel
432 attention for deep convolutional neural networks. In: *2020 IEEE/CVF Conference on Computer*
433 *Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA, 11531-11539 doi:
434 10.1109/CVPR42600.2020.01155.
435 Woo SH, Park J, Lee JY, Kweon IS. 2018. CBAM: convolutional block attention module.
436 *Computer Vision - ECCV 2018* 11211:3-19 doi: 10.1007/978-3-030-01234-2_1.
437 Xie F, Lin BJ, Liu YC. 2022.
438 Research on the coordinate attention mechanism fuse in a YOLOv5 deep learning detector for th
439 e SAR ship detection task. *Sensors* 22(9):3370 doi: 10.3390/s22093370.
440 Zhang CB, Jiang PT, Hou QB, Wei YC, Han Q, Li Z, Cheng MM. 2021. Delving deep into label
441 smoothing. *IEEE Transactions on Image Processing* 30:5984-5996 doi:
442 10.1109/TIP.2021.3089942.
443 Zhao SJ, Zheng JC, Sun SD, Zhang L. 2022a. An improved YOLO algorithm for fast and
444 accurate underwater object detection. *Symmetry-Basel* 14(8):1669 doi: 10.3390/sym14081669.
445 Zhao L, Zhang Q, Peng B, Yang L. 2022b. Real-time object detector for low-end devices.
446 *Journal of Electronic Imaging* 31(1):013016 doi: 10.1117/1.JEI.31.1.013016.
447 Zheng ZH, Wang P, Liu W, Li JZ, Ye RG, Ren DW. 2019. Distance-IoU loss: faster and better
448 learning for bounding box regression. *arXiv preprint* arXiv:1911.08287.
449 Zhou QK, Zhang W, Li RZ, Wang J, Zhen SH, Niu F. 2022. Improved YOLOv5-S object
450 detection method for optical remote sensing images based on contextual transformer. *Journal of*
451 *Electronic Imaging* 31(4):043049 doi: 10.1117/1.JEI.31.4.043049.

Figure 1

The structure of YOLOv4-tiny

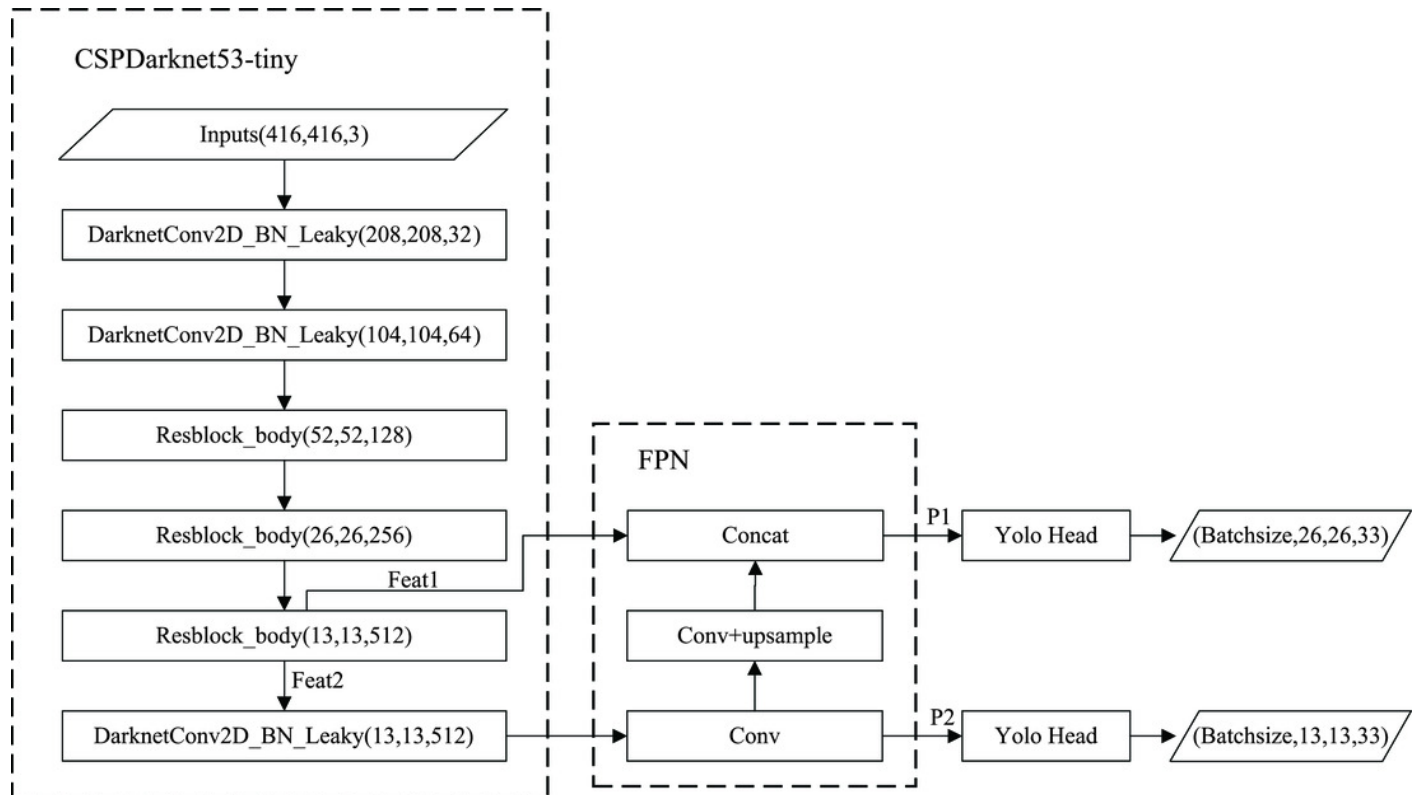



Figure 2

The structure of Resblock_body

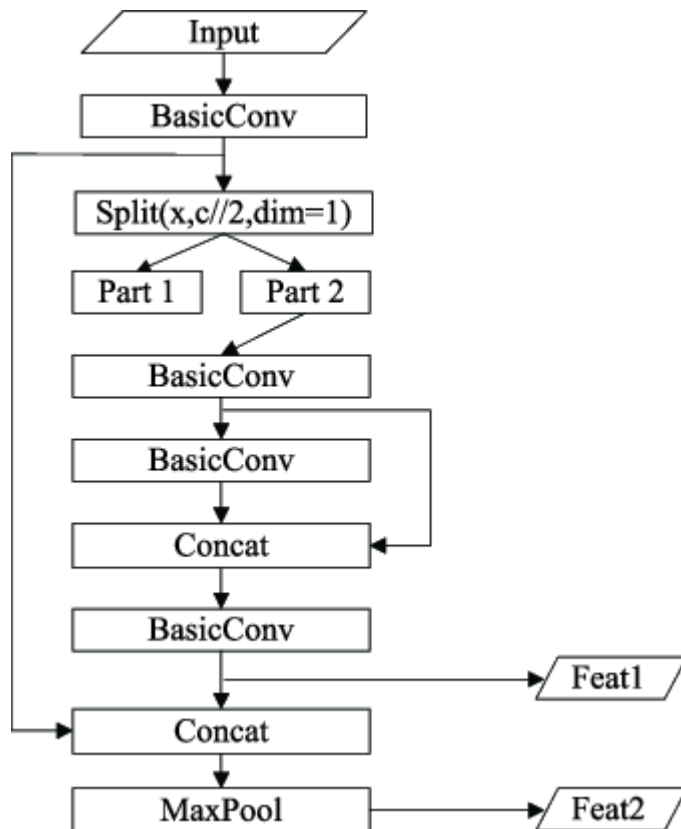



Figure 3

The specific implementation of SE

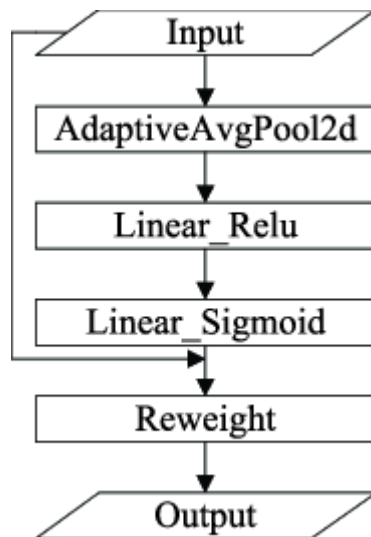



Figure 4

The specific implementation of ECA

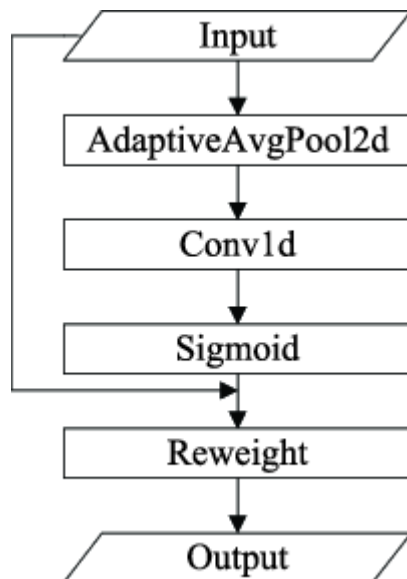



Figure 5

The specific implementation of CBAM

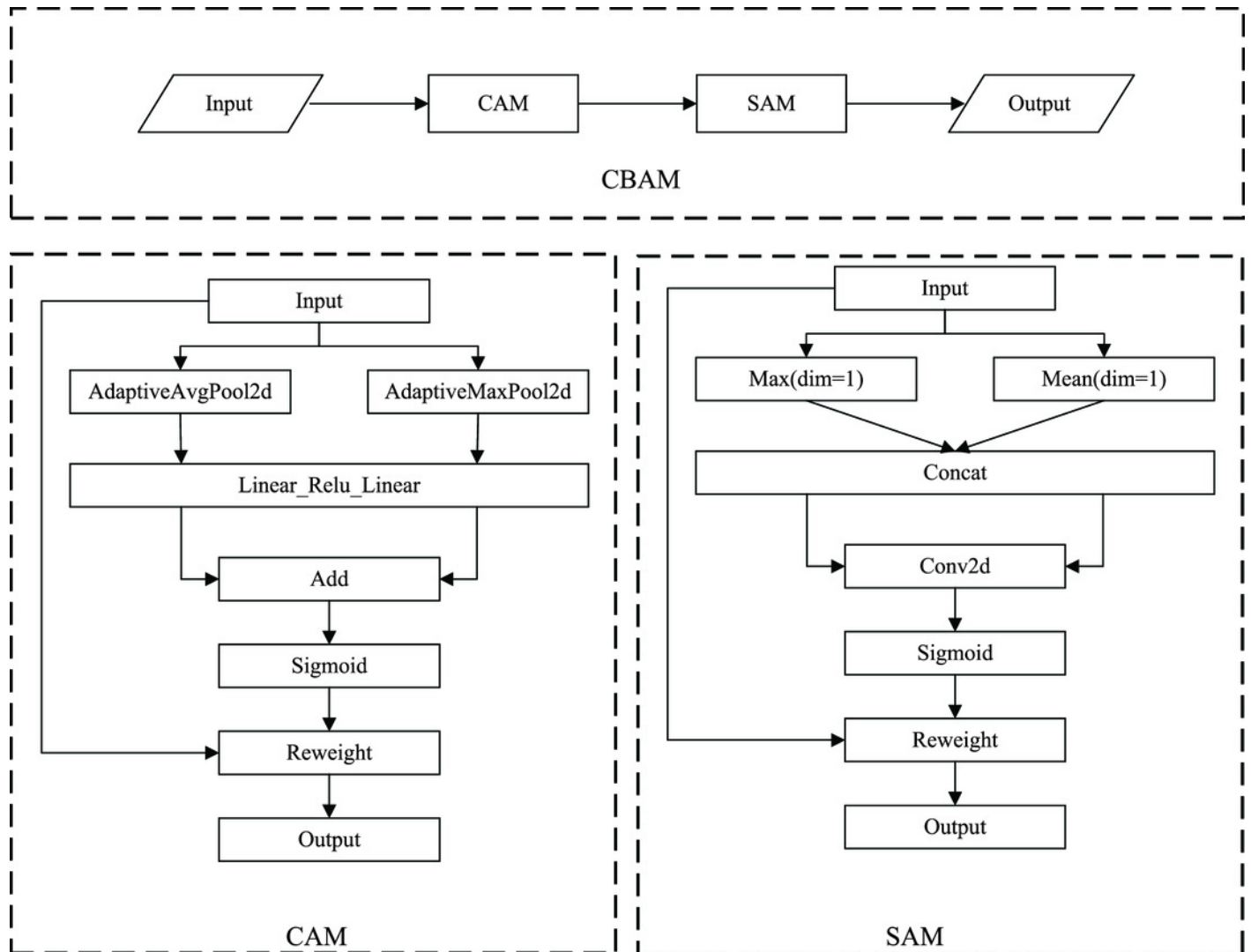



Figure 6

Improved YOLOv4-tiny based on attention mechanisms

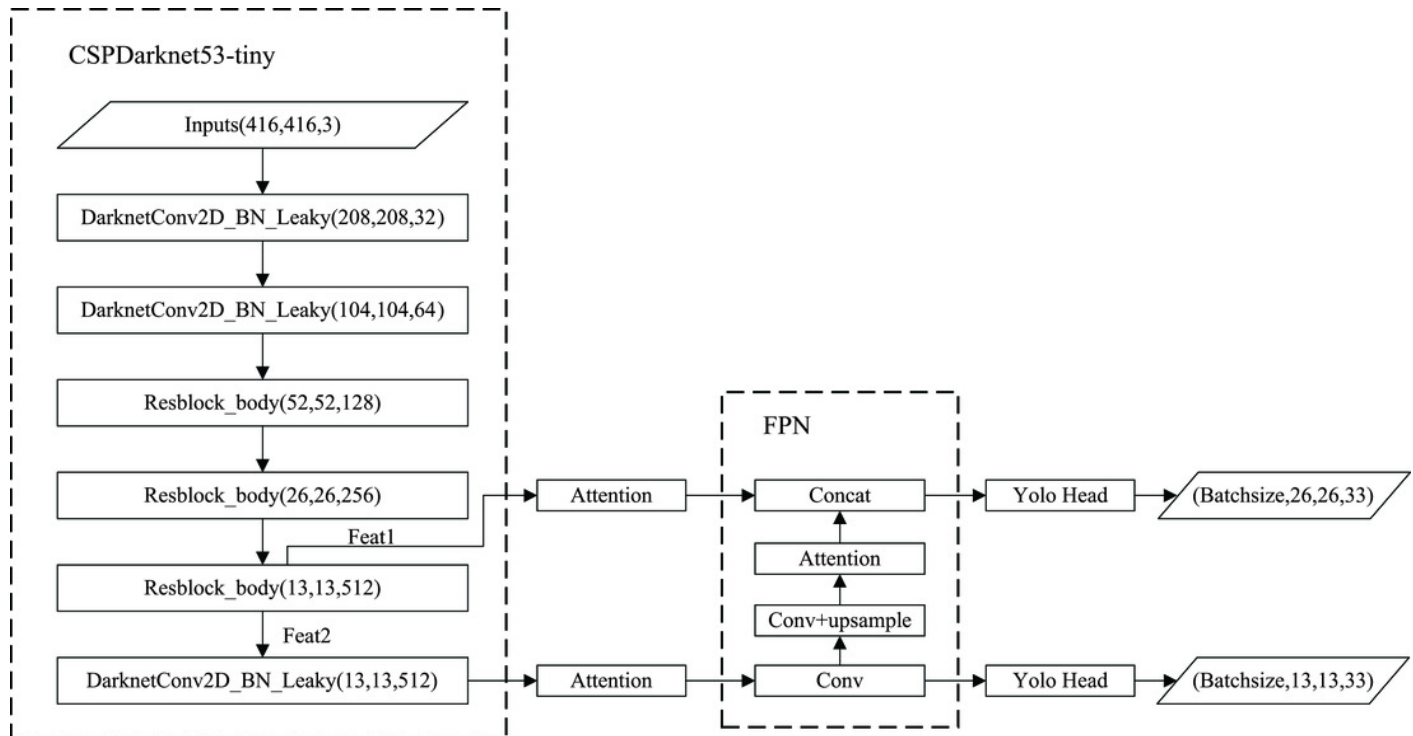



Figure 7

P-R curves

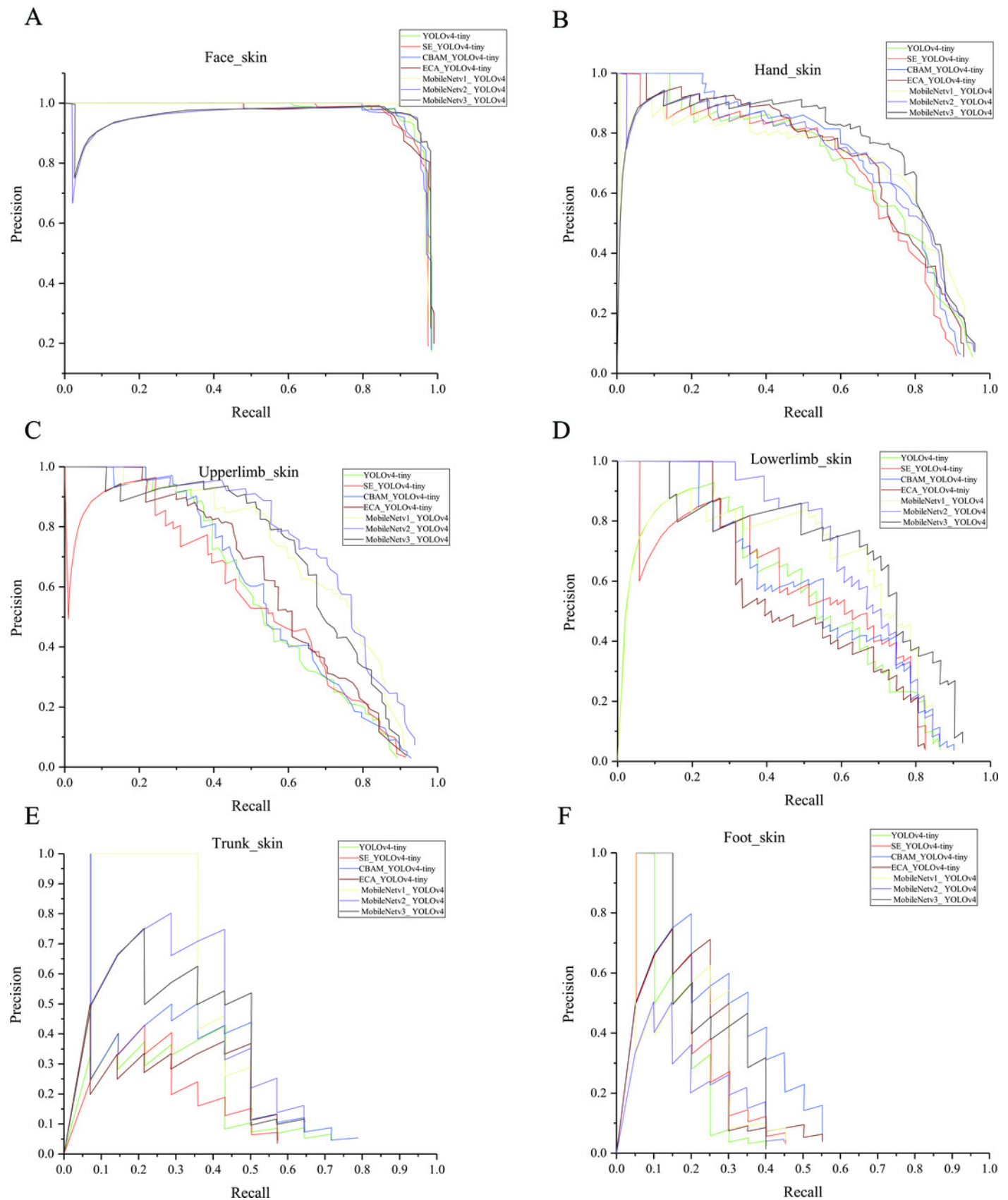
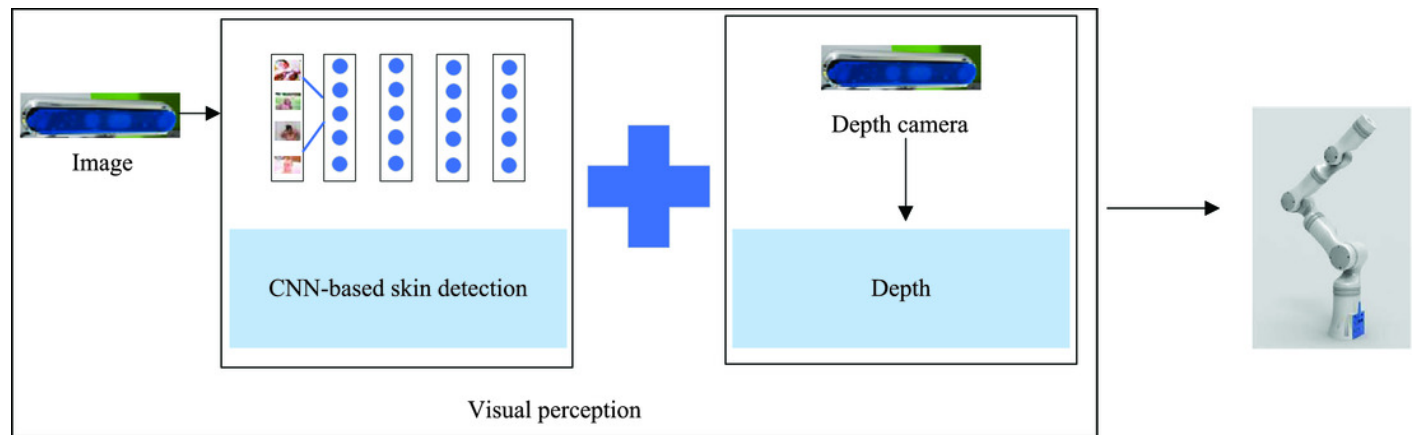



Figure 8

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

The number of pictures for different factors

Skin position		Occlusion		Illumination		Blur degree		Water mist		Number of regional categories		Skin color	
Middle	589	Yes	687	Sufficient	708	Blur	424	Yes	323	All	289	Fair	277
Border	411	No	313	Insufficient	292	Clear	576	No	677	Some	711	Medium	421
-	-	-	-	-	-	-	-	-	-	-	-	Dark	302

Summary of IOU, GIOU, DIOU, and 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

Model 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
MobileNetv1_YOLOv4	-	60.5%	51.1
MobileNetv2_YOLOv4	-	60.2%	46.5
MobileNetv3_YOLOv4	-	61.8%	53.7

Model comparison results in previous work

Models	Backbone	mAP
Faster R-CNN	ResNet50	0.72
Faster R-CNN	MobileNetv2	0.55
YOLOv3	DarkNet53	0.70
YOLOv4	CSPDarkNet53	**0.78**
CenterNet	Hourglass	0.66

AP values for the six categories

	Face_skin	Hand_skin	Upperlimb_skin	Lowerlimb_skin	Trunk_skin	Foot_skin
YOLOv4-tiny	0.97	0.68	0.57	0.54	0.21	0.18
SE_YOLOv4-tiny	0.96	0.67	0.55	0.55	0.17	0.21
CBAM_YOLOv4-tiny	**0.97**	0.72	0.58	0.56	0.30	**0.30**
ECA_YOLOv4-tiny	0.97	0.70	0.60	0.51	0.21	0.23
MobileNetv1_YOLOv4	0.96	0.70	0.70	0.64	**0.41**	0.22
MobileNetv2_YOLOv4	0.96	0.71	**0.73**	0.67	0.40	0.14
MobileNetv3_YOLOv4	0.97	**0.76**	0.69	**0.69**	0.34	0.26

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

Models	A	B	W
SE_YOLOv4-tiny	0.2	-0.9	-0.74
CBAM_YOLOv4-tiny	0.4	4.7	**3.15**
ECA_YOLOv4-tiny	0	1.1	1.1
MobileNetv1_YOLOv4	28.7	8	2.75
MobileNetv2_YOLOv4	24.1	7.7	2.63
MobileNetv3_YOLOv4	31.3	9.3	2.37