Animal Pose Estimation Algorithm Based on the Lightweight Stacked Hourglass Network

Zhang Wenwen  
Guizhou University

Xu Yang (✉️ xuy@gzu.edu.cn)  
Guizhou University

Bai Rui  
Guizhou University

Li Li  
Guizhou University

Keywords:

Posted Date: May 23rd, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1627999/v1

License: ☕️️ This work is licensed under a Creative Commons Attribution 4.0 International License. Read Full License
Animal Pose Estimation Algorithm
Based on the Lightweight Stacked Hourglass Network

Wenwen Zhang\(^1\), Yang Xu\(^1,\)\(^2\), Rui Bai\(^1\) & Li Li\(^1\)

Abstract: Pose estimation has been a hot topic in the field of machine vision in recent years. In the pose estimation task, a lightweight stacked hourglass network (SHN) algorithm is proposed. Moreover, aiming at the problem of large parameters in depthwise convolutional neural networks, a lightweight residual module is proposed, that is, based on the lightweight efficient channel attention improved conditional channel-weighted method (ICCW-Bottle), which replaces bottleneck module, thereby reducing the weight of the network and obtaining the feature information of different scales. Given the problem that a large amount of feature information is easily lost after the network pooling operation, a lightweight dual-branch fusion module is proposed that fully integrates high-level semantic information and low-level detailed features under the condition of a small number of parameters. Finally, the training model of synthetic animal dataset and real animal dataset was jointly applied. Compared with the consistency-constrained semi-supervised learning (CC-SSL) method, the proposed method increased in accuracy of pose estimation by 5.5\%. It also reduced the number of network parameters and the calculation amount. The results of the ablation experiment verify the advancement and effectiveness of the overall network.

Introduction

With the rapid development of machine vision, pose estimation\(^1\) has gradually become an important part of many computer vision fields. It is widely used in many fields, such as human–computer interaction\(^2\), intelligent surveillance, and pose tracking. Pose estimation mainly refers to the process of identifying and estimating the position of each part or joint point of a detection target from an image. Applications such as pose tracking and gait analysis require accurate pose estimation as support. Therefore, it is particularly important to conduct in-depth research on pose estimation. The earliest methods of pose estimation are mainly graph structure model matching and K-means clustering, but traditional methods are susceptible to the interference of complex backgrounds and lead to repeated calculations when there is occlusion of the body, resulting in low accuracy of the extracted features and suboptimal results. With the rapid development of computer vision, the convolutional neural network (CNN) method is used for calculation, the parameters for calculating the weight of convolutional layer are less, and the CNN can still recognize the target in the picture well when it is panned to another position, which can greatly improve the accuracy of pose estimation.

\(^1\)College of Big Data and Information Engineering, Guizhou University, Guiyang 550025, China. \(^2\)Guiyang
Currently, there are two main problems in pose estimation. First, when the pose estimation finds the detection target from an image, the detection target is vulnerable to the camera angle and the detection and recognition of occluded target. Second, the pose estimation algorithm mainly focuses on improving its performance while ignoring its computational efficiency. Toshev et al. proposed the DeepPose network to perform regression learning on the keypoints of the detection target, which is superior to the traditional human pose estimation method. Lin et al. proposed the feature pyramid network (FPN), which can detect precise small targets but lacks contextual information, resulting in low detection performance and difficulty in identifying keypoints in occluded parts. Chen et al. proposed the cascaded pyramid network (CPN), which improves the performance of keypoint detection, but its generalization performance for multi-scale pose estimation is poor owing to the lack of structural information between joints. Sun et al. proposed the high-resolution network (HRNet), which always maintains high-resolution feature maps and solves the problem of low detection accuracy of human keypoints in medium and low resolutions. However, if serious occlusion occurs, the estimation results of the HRNet have large errors. Lan et al. proposed the generative adversarial network (GAN), which improves the generalization performance and accuracy of the network through adversarial training between the generator and the discriminator; but it has problems; for example, the loss function cannot indicate the training process, and the generated samples lack diversity. Newell et al. proposed the stacked hourglass network (SHN), which creates the network feature layer by concatenating the network from high resolution to low resolution and then from low resolution to high resolution. The layers are superimposed, and the features of all layers are retained so that the human body joint point information is learned in the way of heat map detection. Moreover, the detection results of the entire image are inferred, thereby improving the resolution of the image features.

In recent years, with the continuous development of deep learning, animal pose estimation has been widely used in many fields, such as zoology, biology, and aquaculture. At present, although there are datasets containing animals, most animal datasets are built for classification and detection, and only a small number of animal datasets are used to resolve animal keypoints. Because a large number of datasets for animals are labeled, which is extremely expensive, existing work has mostly addressed this problem by employing synthetic animal datasets. Therefore, the requirement for a large-scale animal dataset remains a problem for animal pose estimation research. In recent years, most researchers have used a deeper CNN to improve the detection accuracy while ignoring the problems of the increasing number of parameters and computational complexity in the neural network, which has seriously hindered the method’s use in resource-constrained situations.

Based on the above research, this paper aims to improve the SHN with four stacks as the basic network framework for animal pose estimation and proposes a lightweight residual module, the improved residual module ICCW-Bottle, to replace the bottleneck module so as to lighten the network and obtain feature information at different scales from feature maps with different size perceptual fields. Aiming at the problem that a large amount of feature information is lost after the network pooling operation, a lightweight dual-branch fusion module is proposed, which fuses high-level semantic information and low-level detailed features. While we reduce the size and complexity of model parameters, the accuracy of the network model is improved. Considering the lack of a large number of animal datasets for animal pose estimation, we used a combination of synthetic animal datasets and real animal datasets to train the model, which could minimize the cost and achieve the best effect.
Methods

Model Process. In this paper, SHN is used as the basic network architecture, and a residual module based on the efficient channel attention improved conditional channel-weighted (ECA-ICCW) method, ICCW-Bottle, is proposed to replace the bottleneck module, which retains and accumulates the feature maps of different receptive fields while making the network more lightweight. Then, a lightweight dual-branch feature fusion module is added to the improved lightweight network, and the context information is aggregated under the condition that the parameter amount does not change much. This, in turn, reduces the size of the model parameter amount and improves the accuracy of pose estimation. The overall network model is shown in Figure 1.

Figure 1. Overall model.

Stacked Hourglass Network. The backbone network of this paper is the SHN for joint point localization. SHN uses the residual module Bottleneck as the basic module to construct the entire network. The module is divided into two paths. The main path includes three convolutional layers to extract high-level features. The size of the convolution kernel of the first layer is $11 \times 1$, the size of the second layer of the convolution kernel is $33 \times 3$, and the size of the third layer of the convolution kernel is $1 \times 1$. Before each convolutional layer, a batch normalization (BN) layer and a ReLu layer are passed through, and the structure is shown in Figure 7(a).

SHN is composed of multiple hourglass networks (HNs) stacked in a series, and each hourglass module consists of Maxpool, an upsampling layer, and a Residual Module. The feature map should be separated from the branch before passing through the pooling layer. To retain the spatial information of the image, the node information of the original size (Outsize) of different scales, $1/2$, $1/4$, and $1/8$, is extracted. After each pass through the pooling layer, the resolution and computational complexity of the feature map can be reduced, and the image features are also extracted through the residual module. Then, the feature map is upsampled by the Nearest Interpolation method to recover the high resolution. The features extracted by the network incorporate multi-scale and contextual information, which not only retains the information of all layers but also has the same size as the original image. Therefore, the network does not change the image size. Its structure is shown in Figure 2.
Hourglass network.

**Efficient channel attention (ECA).** Existing studies have shown that adding attention mechanisms to CNNs can greatly improve the performance of the network. For example, attention mechanisms such as SENet\textsuperscript{16}, CBAM\textsuperscript{17}, and CA\textsuperscript{18} can greatly improve the performance of the network. However, many existing methods are devoted to designing more complex attention mechanisms to achieve better performance, which inevitably increase the number of parameters and computation of the model, making the model more complex. Therefore, Wang et al.\textsuperscript{19} proposed ECA module, as shown in Figure 3. This module makes some improvements to SE attention, which interacts with cross-channel information without reducing the channel dimension. This can significantly reduce the model complexity while maintaining the model’s performance.

Suppose the output of a convolutional block is $\chi \in \mathbb{R}^{W \times H \times C}$, where $W$, $H$, and $C$ represent the width, height, and number of channels, respectively, and GAP represents global average pooling. The ECA module realizes the information interaction between channels through fast one-dimensional convolution with a convolution kernel size of $k$, as follows:

$$\omega = \sigma \left( C1D \left( y \right) \right),$$

where $\sigma$ is the sigmoid function, $C1D$ is the 1D convolution, $k$ is the kernel size of the 1D convolution, and $y$ is the output signal. When $k = 3$, the ECA module can achieve a similar effect as SE-Var3, while the model complexity is much lower compared to SE-Var3.

**Conditional Channel Weighting.** In existing research, pose estimation has usually required high-resolution feature maps to achieve high performance, which inevitably increases the number of parameters and the computation of the model. In an article on CVPR in 2021, Yu et al.\textsuperscript{20} proposed a lightweight unit, namely Conditional Channel Weighting (CCW), which embeds the Shuffle block in ShuffleNet into the CNN to obtain
a lightweight scheme with better performance than that of the CNN. Moreover, it aims at the computational bottleneck problem at the Shuffle block. The lightweight unit CCW is introduced to replace the \(1 \times 1\) convolution of the Shuffle block to obtain a high-performance lightweight network, as shown in Figure 4.

**Figure 4.** CCW module.

Assuming that the number of channels of the input and output feature maps of a convolution is \(C\), the time complexity of the \(1 \times 1\) convolution is \(O\left(C^2\right)\), and the linear time complexity of the \(3 \times 3\) depthwise convolution is \(O\left(9C\right)\). In the Shuffle block, the complexity of the two \(1 \times 1\) convolutions is much higher than that of the depthwise convolution: \(O\left(2C^2\right) > O\left(9C\right)\) when \(C > 5\). Therefore, to further reduce the computational complexity of the network, a lightweight unit CCW is used to replace the \(1 \times 1\) convolution of the Shuffle block, as follows:

\[
Y_s = W_s \odot X_s,
\]

where \(W_s\) is the 3D tensor of size \(W_s \times H_s \times C_s\), and \(\odot\) is the element-wise multiplication operator.

The complexity of this computing unit is linearly related to the number of channels \(O\left(C\right)\), which is much lower than the \(1 \times 1\) convolution in the Shuffle block, which plays a role in exchanging information across channels and resolutions.

**ECA-ICCW.** According to the structure of the SHN, there are too many residual modules in the network, which makes the network large and the data processing more redundant. In each residual module, a standard convolution of \(3 \times 3\) and two \(1 \times 1\) convolutions are needed. When the network inputs a high-resolution image, the standard convolution easily brings a huge number of parameters and computation. At the same time, the standard convolution lacks information exchange across channels. It is from this that convolution methods such as cavity convolution\(^{21}\), grouping convolution\(^{22}\), and depthwise separable convolution\(^{23}\) were derived.

Therefore, this paper proposes an improved conditional channel-weighted ECA-ICCW based on efficient channel attention ECA to replace the \(3 \times 3\) convolution in the residual module Bottleneck, as shown in Figure 5.
In ECA-ICCW, the channel is not divided, and the feature map is directly input into the two branches, which reduces the calculation amount of the network to a certain extent. As this module is an improvement of the lightweight unit CCW, this module greatly reduces the number of parameters and the calculation of the network, adds ECA efficient channel attention, and properly performs cross-channel information exchange, which can keep the network lightweight. At the same time, it brings obvious performance gains and significantly reduces the complexity of the model.

**Mish activation function.** An activation function refers to a function that operates on the neurons of an artificial neural network and is responsible for mapping the input of the neuron to the output. The activation function is the source of nonlinearity in the neural network. The introduction of the activation function can strengthen the learning ability of the network. The activation functions commonly used in neural networks are the Relu, Sigmoid, Tanh, and Swish functions.

The Mish activation function is used in the CNN in this paper. As shown in Figure 6, the function has no upper boundary and lower boundary, which avoids the saturation caused by the capping. Moreover, the function is smooth and non-monotonic; hence, it can obtain a better gradient descent effect. At the same time, better information is allowed to penetrate into the neural network; hence, better accuracy and generalization ability can be obtained. The function expression is as follows:

\[
 f(x) = x \tanh \left( \zeta(x) \right) \\
 = x \tanh \left( \ln \left( 1 + e^x \right) \right),
\]

where \( \zeta(x) \ln \left( 1 + e^x \right) \) is the softplus activation function.

The Mish activation function realizes the self-selection function, which is conducive to replacing an activation function such as RelU, which receives a single scalar input without changing the network parameters. When compute unified device architecture (CUDA) is enabled, Mish can reduce the graphics processing unit (GPU) transmission time and effectively improve the training efficiency of the model.
**Figure 6.** Mish function.

**ICCW-Bottle.** To ensure the constant number of network parameters and complete transmission of the required information flow and minimize the input loss of the next module, a residual module ICCW-Bottle (ECA-ICCW-Bottleneck) based on lightweight unit ECA-ICCW is proposed to improve the accuracy of the feature map, as shown in Figure 7(b).

**Figure 7.** Residual module.

Because the keypoints of different scales need to be detected in the pose estimation, the feature information of different scales needs to be obtained from the feature maps with different sizes of receptive fields. The module uses the jump layer connection method to connect the shallow convolution output and the deep convolution output by the channel cascade method and retains and accumulates the feature maps of multiple receptive fields, which is conducive to improving the feature representation ability of ICCW-Bottle. In addition, the Mish activation function is used to replace the original ReLu activation function in the ICCW-Bottle module, which further improves the accuracy and generalization ability of the
ICCW-Bottle module. Compared with Figure 7(a), the ICCW-Bottle module has better feature extraction ability.

**Depthwise Separable Convolution.** To solve the problem of large network parameters, Chollet et al. proposed depthwise separable convolution based on the traditional convolution structure. The traditional convolution structure is divided into two convolutions: DepthwiseConv (DW) and PointwiseConv (PW). The depthwise convolution performs a convolution operation on each channel in the input feature and the corresponding single-channel convolution kernel, keeping the number of feature maps unchanged so as to filter the input features. Pointwise convolution combines the feature maps’ output by depthwise convolution in the depth direction to generate new feature maps. Using these two smaller convolutions to replace the standard convolution of feature fusion can significantly reduce the number of parameters and the computation of the model. Standard convolution and depthwise separable convolution are shown in Figure 8:

![Convolution module diagram](image)

Figure 8. Convolution module.

Assume that the input feature is \( C \times H \times W \), and after a convolution kernel of size \( K \times K \), the output feature is \( N \times H \times W \).

Then, the parameter \( P_1 \) of standard convolution is given by

\[
P_1 = K \times K \times C \times N ,
\]  

The computational cost \( F_1 \) of standard convolution is given by

\[
F_1 = K \times K \times C \times N \times H \times W ,
\]  

The parameter \( P_2 \) of the depthwise separable convolution is given by

\[
P_2 = K \times K \times C + C \times N ,
\]  

The computational cost \( F_2 \) of depthwise separable convolution is given by

\[
F_2 = K \times K \times C \times H \times W + C \times N \times H \times W ,
\]  

Then, the ratio of depthwise separable convolution to standard convolution is \( \gamma \), which is given by

\[
\gamma = \frac{P_2}{P_1} = \frac{F_2}{F_1} = \frac{1}{N} + \frac{1}{K^2} ,
\]  

Compared with the standard convolution, each convolution kernel of depthwise convolution in
Depthwise separable convolution is a single-channel calculation, and $1 \times 1$ convolutions are added. This significantly reduces the amount of calculation and parameters of the network and makes the network more lightweight.

**Dual-branch feature fusion module.** The shallow features of the CNN contain more location information, but the semantic information is insufficient. The deep features of the network contain richer semantic information, which is conducive to the regression prediction of the center point of the heat map, but the location information is rough. Also, the shallow localization information of SHN is easily lost. Considering the detection accuracy of large-scale animal targets, deep and shallow feature fusion is introduced. In this paper, the preliminarily extracted features of the network are input into the last hourglass network through depthwise separable convolution for feature transmission. As shown in Figure 1, under the condition of keeping the number of parameters unchanged, the high-level semantic information and the bottom detail features are fully integrated to obtain more accurate feature information and to effectively improve the detection accuracy of difficult samples.

**Loss function.** We trained the model using only synthetic data at the beginning and obtained the initial model $f^{(0)}$. Then, we used synthetic data and real data to train $f^{(0)}$, repeating the iteration training $N$ times. On the $n$th iteration, we used $(X_t, Y_t)$ and $(X_t, \hat{Y}_t^{(n)})$ combined with $L^{(n)}$ to train the model. In this paper, the loss function $L^{(n)}$ is defined as the mean square error of the heat map of the source dataset $(X_s, Y_s)$ and the target dataset $X_t$ and is given by

$$L^{(n)} = \sum_i L_{MSE}(f^{(n)}(X'_i), Y'_i) + \gamma \sum_j L_{MSE}(f^{(n)}(X'_j), \hat{Y}_t^{(n-1), j}) \quad (9)$$

where $f^{(n)}$ is the trained model, $X'_i$ is the $i$th image in the target dataset, and $\hat{Y}_t^{(n)}$ is the pseudo-label generated by the $n$th training.

**Discussion**

This paper combines the advantages of CCW and ECA and performs dual-branch feature fusion to propose the model in this paper, whose specific network structure is shown in Table 1. $H$ and $W$ represent the height and width of the input image, respectively. A lightweight unit ECA-ICCW replaces the $3 \times 3$ convolution of the residual module, and the residual module is improved to obtain an ICCW-Bottle that can reduce the weight of the network while retaining the feature maps of different receptive fields. At the same time, the depthwise separable convolution is used to perform dual-branch feature fusion to obtain deep and shallow information. Compared with the original SHN network, the lightweight SHN more effectively reduces the parameter size of the network.
<table>
<thead>
<tr>
<th>Branch</th>
<th>Module</th>
<th>Internal output</th>
<th>Output shape</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Conv $7 \times 7 + \text{BN} + \text{ReLU}$</td>
<td>$64 \times \frac{1}{2} H \times \frac{1}{2} W$</td>
<td>$64 \times \frac{1}{2} H \times \frac{1}{2} W$</td>
</tr>
<tr>
<td>ICCW-Bottle</td>
<td>$[64 \times \frac{1}{2} H \times \frac{1}{2} W]$</td>
<td>$128 \times \frac{1}{2} H \times \frac{1}{2} W$</td>
<td>$128 \times \frac{1}{2} H \times \frac{1}{2} W$</td>
</tr>
<tr>
<td>MMPM-S</td>
<td></td>
<td>$128 \times \frac{1}{4} H \times \frac{1}{4} W$</td>
<td>$128 \times \frac{1}{4} H \times \frac{1}{4} W$</td>
</tr>
<tr>
<td>ICCW-Bottle</td>
<td>$[128 \times \frac{1}{4} H \times \frac{1}{4} W]$</td>
<td>$256 \times \frac{1}{4} H \times \frac{1}{4} W$</td>
<td>$256 \times \frac{1}{4} H \times \frac{1}{4} W$</td>
</tr>
<tr>
<td>HN*4</td>
<td>ICCW-Bottle</td>
<td>$[128 \times \frac{1}{4} H \times \frac{1}{4} W]$</td>
<td>$256 \times \frac{1}{4} H \times \frac{1}{4} W$</td>
</tr>
<tr>
<td>ICCW-Bottle</td>
<td></td>
<td>$256 \times \frac{1}{4} H \times \frac{1}{4} W$</td>
<td>$256 \times \frac{1}{4} H \times \frac{1}{4} W$</td>
</tr>
<tr>
<td>ICCW-Bottle</td>
<td></td>
<td>$256 \times \frac{1}{4} H \times \frac{1}{4} W$</td>
<td>$256 \times \frac{1}{4} H \times \frac{1}{4} W$</td>
</tr>
<tr>
<td>ICCW-Bottle</td>
<td></td>
<td></td>
<td>$256 \times \frac{1}{4} H \times \frac{1}{4} W$</td>
</tr>
<tr>
<td>Conv $1 \times 1$</td>
<td></td>
<td></td>
<td>$256 \times \frac{1}{4} H \times \frac{1}{4} W$</td>
</tr>
<tr>
<td>Concat+Conv $1 \times 1$</td>
<td></td>
<td></td>
<td>$256 \times \frac{1}{4} H \times \frac{1}{4} W$</td>
</tr>
</tbody>
</table>

Table 1. Architecture of the network.

**Experimental results and analysis**

**Dataset.** The SHN was used as the basic network of the experiment, and the experiment used the synthetic animal dataset and the TigDog real animal dataset to train, validate, and test the network. The TigDog...
dataset provides keypoint annotations for horses and tigers. The images of horses were from YouTube. The ratio of the training set to the test set was 5:1. Then, 8380 images were used as the training set, and the remaining 1772 images were used as the test set. The images of tigers came from the national geographic documentary. The ratio of the training set to the test set was 4:1. Then, 6523 images were used as the training set, and the remaining 1765 images were used as the test set. The synthetic animal dataset contained five types of animal images, including horses, tigers, sheep, dogs, and elephants. Each animal category had 10000 images: 8000 images being used as the training set and 2000 images being used as the test set.

To verify the generalization performance of the network model, we also tested it on the VisDA2019 dataset. The dataset had six areas: real, sketch, clipart, painting, infograph, and quickdraw. We mainly used sketch, painting, and clipart to test the generalization performance of the network and verify the advanced nature of the network.

**Evaluation.** Percentage of Correct Keypoints (PCK) is the most commonly used pose estimation evaluation standard. PCK refers to the proportion of correctly estimated keypoints, that is, the proportion of the normalized distance between the detected keypoints and their corresponding ground truth less than the set threshold. A detected joint was considered correct if the distance between the predicted joint and the true joint was within a certain threshold. PCK@0.05 refers to the percentage of correct keypoints at a threshold of 0.05. The specific formula is as follows:

$$PCK_i^k = \frac{\sum_p \delta \left( \frac{d_{pi}}{d_{pi}^{def}} \leq T_k \right)}{\sum_p 1},$$

(10)

$$PCK_{mean}^k = \frac{\sum_p \sum_i \delta \left( \frac{d_{pi}}{d_{pi}^{def}} \leq T_k \right)}{\sum_p \sum_i 1},$$

(11)

where $PCK_i^k$ represents the PCK index of the key point with an id of $i$ under the $T_k$ threshold, $PCK_{mean}^k$ represents the algorithm PCK index under $T_k$ thresholds, $k$ represents the $k$th threshold, the key point $i$ represents the id of $i$, $p$ represents the $P$th animal, $d_{pi}$ represents the Euclidean distance between the predicted value of the key point with the id of 3 in the $P$th animal and the manually labeled value, $d_{pi}^{def}$ represents the scale factor of the $P$th animal, and $T_k$ represents a manually set threshold.

**Implementation Details.** The PyTorch framework was used to realize the network architecture. In order to obtain accurate network parameters and effectively train and optimize the network, the high-performance NVIDIA GeForce GTX 3090 was used to train the network. The software platform used in this experiment was Python 3.8.

The number of SHNs was 4. In this experiment, RMSProp optimizer was selected to optimize the
model. The training cycle of epoch was 200, the batch size was 10, the initial learning rate was $2.5 \times 10^{-4}$, the attenuation coefficient of the learning rate was 0.1, and the learning rate decayed once in 120 and 180 cycles, 10 times each time.

To evaluate the advancement and effectiveness of our method, we improved and optimized the original network and then conducted experiments on the TigDog dataset and the synthetic dataset. The input image was cut to $256 \times 256$, and the image was randomly rotated and flipped to enhance the data. Finally, it was compared with other advanced animal pose estimation networks. Figure 9 shows the heatmap results obtained when training on images.

![Figure 9](image)

#### Experimental results and analysis

Table 2 shows the comparison of the parameters and computational complexity of our method and other methods. The experimental results show that compared with the current CC-SSL that achieved higher performance, the parameters and computational complexity of the method in this paper were significantly reduced, indicating the effective performance of the proposed model. Compared with CC-SSL, the number of parameters was reduced by 57.6%, and the computational complexity was reduced by 53.0%.

<table>
<thead>
<tr>
<th>Method</th>
<th>Params (M)</th>
<th>GFLOPs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cycgan</td>
<td>34.11</td>
<td>37.62</td>
</tr>
<tr>
<td>BDL</td>
<td>27.50</td>
<td>30.10</td>
</tr>
<tr>
<td>Cycada</td>
<td>39.25</td>
<td>42.31</td>
</tr>
<tr>
<td>CC-SSL</td>
<td>15.29</td>
<td>17.51</td>
</tr>
<tr>
<td>Ours</td>
<td>6.48</td>
<td>8.23</td>
</tr>
</tbody>
</table>

Table 2. Comparison of model parameters and computational complexity.

Table 3 shows the experimental results on the TigDog dataset. When the animal was a horse, the PCK@0.05 accuracy of the method proposed in this paper and the existing method was compared. "Real" means the model was trained only with the real animal pose dataset, while "Syn" means the model was trained only with synthetic data. The experimental results from Table 3 show that compared with the CC-SSL method, the method in this paper reduced the size of the model parameters in the horse experiment and simultaneously improved the PCK@0.05 accuracy of pose estimation by 5.5%. The results of training directly on real images were close. Table 4 shows the experimental results on the TigDog dataset. When the animal was a tiger, the PCK@0.05 accuracy of the proposed method and other advanced methods was compared. From the experimental results in Table 4, the PCK@0.05 of the method in this paper was 3.9%
higher than that of CC-SSL in the tiger experiment. As tigers usually live in forests and are often shaded by surrounding creatures, however the synthetic animal dataset used for training had no such shading; it was difficult for the model to adapt to heavily occluded scenes. Thus, all methods in Table 4 failed to achieve the same accuracy as horses.

As shown in Tables 3 and 4, the proposed method in this paper improved the accuracy of PCK@0.05 compared to Cycgan\textsuperscript{27}, BDL\textsuperscript{28}, Cycada\textsuperscript{29}, and CC-SSL\textsuperscript{30} while reducing the size of model parameters.

<table>
<thead>
<tr>
<th>Method</th>
<th>Eye</th>
<th>Chin</th>
<th>Shoulder</th>
<th>Hip</th>
<th>Elbow</th>
<th>Knee</th>
<th>Hooves</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real</td>
<td>79.04</td>
<td>89.71</td>
<td>71.38</td>
<td>91.78</td>
<td>82.85</td>
<td>80.80</td>
<td>72.76</td>
<td>78.98</td>
</tr>
<tr>
<td>Syn</td>
<td>46.08</td>
<td>53.86</td>
<td>20.46</td>
<td>32.53</td>
<td>20.20</td>
<td>24.20</td>
<td>17.45</td>
<td>25.33</td>
</tr>
<tr>
<td>Cycgan\textsuperscript{27}</td>
<td>70.73</td>
<td>84.46</td>
<td>56.97</td>
<td>69.30</td>
<td>52.94</td>
<td>49.91</td>
<td>35.95</td>
<td>51.86</td>
</tr>
<tr>
<td>BDL\textsuperscript{28}</td>
<td>74.37</td>
<td>86.53</td>
<td>64.43</td>
<td>75.65</td>
<td>63.04</td>
<td>60.18</td>
<td>51.96</td>
<td>62.33</td>
</tr>
<tr>
<td>Cycada\textsuperscript{29}</td>
<td>67.57</td>
<td>84.77</td>
<td>56.92</td>
<td>76.75</td>
<td>55.47</td>
<td>48.72</td>
<td>43.08</td>
<td>55.57</td>
</tr>
<tr>
<td>CC-SSL\textsuperscript{30}</td>
<td>84.60</td>
<td>90.26</td>
<td>69.69</td>
<td>85.89</td>
<td>68.58</td>
<td>68.73</td>
<td>61.33</td>
<td>70.77</td>
</tr>
<tr>
<td><strong>Ours</strong></td>
<td><strong>90.06</strong></td>
<td><strong>93.14</strong></td>
<td><strong>74.43</strong></td>
<td><strong>84.27</strong></td>
<td><strong>75.12</strong></td>
<td><strong>75.46</strong></td>
<td><strong>66.13</strong></td>
<td><strong>74.63</strong></td>
</tr>
</tbody>
</table>

Table 3. Comparison results of PCK@0.05 accuracy of different methods (Horse).

<table>
<thead>
<tr>
<th>Method</th>
<th>Eye</th>
<th>Chin</th>
<th>Shoulder</th>
<th>Hip</th>
<th>Elbow</th>
<th>Knee</th>
<th>Hooves</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real</td>
<td>96.77</td>
<td>93.68</td>
<td>65.90</td>
<td>94.99</td>
<td>67.64</td>
<td>80.25</td>
<td>81.72</td>
<td>81.99</td>
</tr>
<tr>
<td>Syn</td>
<td>23.45</td>
<td>27.88</td>
<td>14.26</td>
<td>52.99</td>
<td>17.32</td>
<td>16.27</td>
<td>19.29</td>
<td>21.17</td>
</tr>
<tr>
<td>Cycgan\textsuperscript{27}</td>
<td>71.80</td>
<td>62.49</td>
<td>29.77</td>
<td>61.22</td>
<td>36.16</td>
<td>37.48</td>
<td>40.59</td>
<td>46.47</td>
</tr>
<tr>
<td>BDL\textsuperscript{28}</td>
<td>77.46</td>
<td>65.28</td>
<td>36.23</td>
<td>62.33</td>
<td>35.81</td>
<td>45.95</td>
<td>54.39</td>
<td>52.26</td>
</tr>
<tr>
<td>Cycada\textsuperscript{29}</td>
<td>75.17</td>
<td>69.64</td>
<td>35.04</td>
<td>65.41</td>
<td>38.40</td>
<td>42.89</td>
<td>48.90</td>
<td>51.48</td>
</tr>
<tr>
<td>CC-SSL\textsuperscript{30}</td>
<td>96.75</td>
<td>90.46</td>
<td>44.84</td>
<td>77.61</td>
<td>55.82</td>
<td>42.85</td>
<td>64.55</td>
<td>64.14</td>
</tr>
<tr>
<td><strong>Ours</strong></td>
<td><strong>96.94</strong></td>
<td><strong>92.43</strong></td>
<td><strong>46.72</strong></td>
<td><strong>78.64</strong></td>
<td><strong>56.18</strong></td>
<td><strong>43.25</strong></td>
<td><strong>66.45</strong></td>
<td><strong>66.74</strong></td>
</tr>
</tbody>
</table>

Table 4. Comparison results of PCK@0.05 accuracy of different methods (Tiger).

Figure 10 shows the visualization result of pose estimation and local segmentation. This method could produce more accurate prediction, even for some extreme postures; for example, the tiger lying down, horse being ridden, hurdles, and fuzzy horses could be accurately predicted. In addition, as shown in Figure 11, this method could perform good pose estimation for other animal categories, such as sheep, elephants, and dogs.

![Figure 10](image-url)
Figure 11. Visualization of other animals.

Generalization test on VisDA2019. To verify the generalization performance of the model, we also used images from the Vision Domain Adaptation Challenge dataset (VisDA2019) for testing. The dataset had six areas, including real, sketch, clipart, painting, infograph, and quickdraw. This paper mainly used sketch, painting, and clipart to test the generalization performance of the network, as shown in Figure 12 (from left to right for each animal: clipart, painting, and sketch).

Figure 12. Visual results of VisDA2019.

Experiments showed that both the CC-SSL method and the method in this paper outperformed the model trained on real images, thus proving the feasibility of jointly training the model using synthetic datasets and real datasets, as shown in Table 5.

<table>
<thead>
<tr>
<th>Method</th>
<th>Visible Kpts Accuracy</th>
<th>Full Kpts Accuracy</th>
<th>Visible Kpts Accuracy</th>
<th>Full Kpts Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Horse</td>
<td>Tiger</td>
<td>Horse</td>
<td>Tiger</td>
</tr>
<tr>
<td></td>
<td>Sketch</td>
<td>Painting</td>
<td>Clipart</td>
<td>Sketch</td>
</tr>
<tr>
<td>Real</td>
<td>65.37</td>
<td>64.45</td>
<td>64.43</td>
<td>61.28</td>
</tr>
<tr>
<td>CC-SSL</td>
<td>72.29</td>
<td>73.71</td>
<td>73.47</td>
<td>70.31</td>
</tr>
<tr>
<td>Ours</td>
<td>75.71</td>
<td>78.57</td>
<td>74.56</td>
<td>72.43</td>
</tr>
</tbody>
</table>

Table 5. PCK@0.05 accuracy of VisDA2019 dataset.

Ablation Study and Analysis. Based on CC-SSL, this paper proposes a lightweight SHN algorithm. Considering the problem of large parameters of deep CNN, a lightweight residual module is proposed to obtain feature information at different scales while making the network lightweight. A lightweight dual-branch fusion module is proposed to solve the problem where a large number of feature information is easily lost after the network pooling operation. In order to prove the effectiveness and advancement of each key module in the proposed model, we performed ablation experiments on the TigDog dataset and synthetic animal dataset with horses as the experimental subjects. At the same time, we compared it with CC-SSL. The “√” representation model contains this module, and the experimental results are shown in Table 6.
### Table 6. Results of ablation experiments.

Table 6 shows that under the PCK@0.05 index, compared with CC-SSL, a residual module ICCW-Bottle (without ECA-ICCW) was proposed to replace the original Bottleneck, which improved the accuracy by 3.1%. On this basis, it was proposed that the convolution of the lightweight unit ECA-ICCW to replace the residual module was improved by 3.4%, and the addition of the dual-branch feature fusion module was improved by 1.5%. As can be seen from the table, the final model was 5.5% more accurate than the baseline, fully demonstrating the feasibility of the method proposed in this paper.

### Conclusions

In this paper, the SHN was improved and optimized. A lightweight unit ECA-ICCW was designed to replace the $3 \times 3$ convolution of the residual module, and the residual module was improved to effectively reduce the number of parameters of the model and improve the ability of the model to obtain information at different scales. A dual-branch feature fusion method was proposed to make the network fully extract and fuse the feature information of the context. The experimental results on the TigDog dataset show that the method in this paper improved the accuracy of pose estimation by 5.5% compared with CC-SSL. The proposed network also reduced the number of parameters and the computation of the network, as well as achieved parallel accuracy and speed. However, the study topic still has room for improvement, such as solving the problem of animals being occluded.

### Data availability

All data used to support the findings of this study are available from the corresponding author upon request.

### References

Acknowledgments

We thank LetPub (www.letpub.com) for its linguistic assistance during the preparation of this manuscript.

Author contributions

Y. X. conceived the idea and the experiments. W. Z. improved the idea and conducted the experiments. L. L. and R. B. supervised the work. All authors reviewed the manuscript.

Competing interests

The authors declare no competing interests.
Data availability

All data used to support the findings of this study are available from the corresponding author upon request.

Additional information

Correspondence and requests for materials should be addressed to Y. X.