



Pose estimation for swimmers in video surveillance

Xiaowen Cao¹ · Wei Qi Yan¹

Received: 22 December 2021 / Revised: 8 July 2023 / Accepted: 21 August 2023
© The Author(s) 2023

Abstract

Traditional models for pose estimation in video surveillance are based on graph structures, in this paper, we propose a method that breaks the limitation of template matching within a range of pose changes to obtain robust results. We implement our swimmer pose estimation method based on deep learning. We take use of High-Resolution Net (HRNet) to extract and fuse visual features of visual object and complete the object detection using the key points of human joint. The proposed model could be applied to all kinds of swimming styles throughout appropriate training. Compared with the methods that require multimodel combinations and training, the proposed method directly achieves the end-to-end prediction, which is easily to be implemented and deployed. In addition, a cross-fusion module is added between parallel networks, which assists the network to make use of the characteristics of multiple resolutions. The proposed network has achieved ideal results in the pose estimation of swimmers by comparing HRNet-W32 and HRNet-W48. In addition, we propose an annotated key point dataset of swimmers which was created from the view of underwater swimmers. Compared with side view, the torso of swimmers collected by the underwater view is much suitable for a broad spectrum of machine vision tasks.

Keywords Pose estimation · Swimmer · HRNet · DCNN

1 Introduction

Swimming is a popular game in sport competitions. However, the special environment of swimmers increases the difficulty of event monitoring and daily coaching, it is not easy for people to intuitively understand real-time dynamics and movement details of swimmers. After the competition, analysts often need to replay game footages and spend hours manually recording player gestures and analyzing event insights from the footages [1]. This has brought resistance to swimming competitions and swimmers' daily training as well as limited the development of swimming. In recent years, digital image and video technology related to machine vision has been developed for swimming analysis, which includes the algorithms that automatically analyses the swimmer behaviors instead of using human

✉ Wei Qi Yan
wyan@aut.ac.nz

¹ Auckland University of Technology, CBD, Auckland 1010, New Zealand

visual systems, pose estimation of swimmers in machine vision is one of most important parts of this task. The pose estimation is the most essential description of human behaviors and actions, which could be employed for multi-swimmer pose estimation, action recognition of swimmers, kinematic pose rectification of swimmers, drowning detection and rescue, so on.

In the early days, underwater surveillance was not perfect, most postures of swimmers are estimated from a side perspective with visual data taken from an ordinary camera installed outside of the transparent swimming lanes [2–4]. In recent years, with the development of intelligent surveillance, in important swimming events, underwater monitoring is implemented, relevant equipment has been gradually installed in swimming pools. The underwater monitoring provides a new view for machine vision related to swimmers. Therefore, unlike most previous studies, we decide to take use of visual data obtained from the underwater perspective to implement the proposed model. Compared with the side views, the visibility under water is rarely affected by water surfaces, which reduces the limb dislocation due to lighting refraction or spray occlusion. In addition, we get a more comprehensive trunk of swimmers throughout underwater view which is helpful for the model to learn complete visual features, also better adaptive to other tasks of machine vision related to swimmers.

Pose estimation of swimmers is regarded as a problem in sports, which means to calculate pose parameters of various parts of human body according to visual information. It mainly encapsulates two-dimensional and three-dimensional methods. Two-dimensional estimation refers to the calculation of two-dimensional coordinates of each key point of human body in image plane [5]. The focus of this research work is mainly on the 2D single person pose estimation.

In the past, conventional methods based on graph structure are thought as the most general solution to the problem of swimmer pose estimation. For example, a pose estimation method has taken Deformable Part Models (DPMs) into account [2], but the result of this method is not ideal if the motion changes obviously. There are two main reasons: (1) The computational costs of low-level features such as specifically designed Histogram of Oriented Gradients (HOGs) and shift features are high, the features extracted by these methods are generally limited which cannot make use of image information. (2) If the angle of posture alters a lot, the template model with a single shape cannot accurately match the changed pose, there may be multiple feasible solutions, the result of pose estimation is not unique. This makes conventional methods hard to be employed in complicated scenes.

With the advent of big data, computing resources (e.g., GPU, etc) make deep learning computations fast and realistic which is highly dependent on the amount of big data. Compared with the methods which rely on specified visual features, the methods based on deep learning using CNNs (i.e., Convolutional Neural Networks) extract visual features from a large amount of image data which includes much abundant and effective information.

According to the way of acquiring visual key points, the methods based on deep learning are grouped into twofold: The methods based on regression and the methods based on heatmaps. The regression methods output key coordinates directly with an end-to-end structure. The difficulty of this kind of methods is that the direct regression of coordinates from the image sequence is a nonlinear problem, the fully connected layer for regression coordinates would have low ability of spatial generalization, which easily leads to overfitting. After that, a method of transition processing [6] is proffered to generate a heatmap and estimate the position of key points. Heatmap is to represent each kind of coordinates with a probability graph and generate a probability for each pixel position in the image, which is applied to represent the probability that the point belongs to the corresponding

class of key points. The closer the distance of the key points, the higher the probability of the pixel tends to 1.00, the shorter distance from the key points, the closer the probability of the pixel is 0, which is simulated by using Gaussian functions. Compared with the coordinates of key points directly, the heatmap can better preserve the spatial location information, which is beneficial for model training.

A slew of artificial neural networks was applied to generate heatmaps that continuously increase the receptive field through multiple downsampling, in order to gain effective spatial and representational information. The information loss is inevitable in the process of reducing the resolution. However, a passel of resolution enhancement methods expanded the resolution by adding 0 or simply estimating a similar value, as a result, the pixel loss cannot be completely reversed. For example, in hourglass [7], max pooling is accommodated to reduce the resolution of feature map, linear interpolation is offered to restore the resolution of the input.

In deep nets, intermediate supervision is added between the stages to optimize the training process. For example, the loss [6] is calculated at the end of each phase to optimize the training results. Each hourglass network [7] adopts loss for supervision, the output heatmaps reflect the error with the true value. In 3D U-Net [8], intermediate supervision was added into the process of upsampling. Intermediate supervision has been proved to be efficacy in network training.

In this paper, we implement a swimmer pose estimation method based on the heatmaps. HRNet [9] is retreated as the core network, the output of HRNet is the high-resolution representation of the input. Heatmaps are regressed through the output of HRNet. In the past, most of deep nets only harnessed unidirectional feature fusion from high resolution to low resolution or from low resolution to high resolution, while each fusion layer of HRNet includes a bi-directional fusion between high-resolution and low-resolution, which aids the network to continuously optimize the high-resolution representation. The network does not use supervised learning, which reduced the computational complexity, the accuracy was not affected. Compared with the net using a serial structure, HRNet has advantages in retaining high-resolution features and makes full use of all levels of visual features. In this paper, our contributions are:

- (1) We implemented a swimmer pose estimation method based on the heatmaps. HRNet [9] is retreated as the core network, the output of HRNet is the high-resolution representation of the input.
- (2) The proposed model saves a great deal of complex training costs which is easier to be deployed and implemented.
- (3) The proposed model has achieved excellent results in speed and accuracy, which deploys a great foundation for other machine vision tasks related to swimming game.

In this paper, related work is depicted in Section 2, the approach is iterated in Section 3, the experiments are demonstrated in Section 4, the conclusion will be drawn in Section 5.

2 Related work

Conventional methods utilize specially assigned visual features to represent the parts of human body, pair-wise relationship between the parts is constrained by a graphical model. For instance, Zecha et al. [2] proposed a method to estimate the pose of swimmers through

deformable models that are trained discriminatively [10]. The task of pose estimation is regarded as a pose detection problem. The sub-models are trained for a variety of poses of the same swimming style. The hypothesis is that every swimming style has periodicity, in the experiments, the swimming cycles are segmented into sections, regarding the continuous images in each period of the same pose, a sub-model was trained specifically. Finally, these models were combined into a hybrid model, each hybrid model was specific to only one swimming style. The experimental results show that there is not obvious boundary between the actions, this method often makes errors during the overrun, the sophisticated algorithms were employed to reduce the occurrence of errors.

In 2014, Toshev et al. [11] proposed DeepPose net, which was the first time that Deep Convolutional Neural Network (DCNN) has been employed to solve the problem of human pose estimation. Since then, the related work on pose estimation of swimmers has combined conventional methods with deep learning methods to improve the efficiency of the existing models.

Zecha et al. [3] proposed a method to estimate the pose of swimmers by using the DCNN representation of DPM. HOG feature was abandoned and AlexNet was employed to extract visual features and complete the classification of components directly, then a deformable part model was created to complete pose estimation. This work achieved pose estimation based on graph structure, as a result, this method still retains the limitations brought by using template matching. In fact, DCNNs obtain multiscale and multiclass human node features under various receptive fields and contextual information of each feature, there is no need to adopt deformable part and constrain the relationship between body parts.

Heatmaps accumulate the contextual information of human body. By using heatmaps as transitions to estimate key points, CNNs establish the spatial structure between nodes implicitly, the deformable model based on prior knowledge is no longer needed. Heatmaps were firstly employed [6] in convolutional pose machines (CPMs). The large receptive field is beneficial for the model to learn spatial structure of various parts of human body. The network of CPMs consists of four sequentially connected phases. In each stage, the classical VGG network was taken to extract belief maps. Deep supervision is added between stages to ensure the quality of deep net training.

CPMs [4] were combined with temporal sequence models to solve the problem of pose estimation of swimmers. The importance of spatial information and temporal information is emphasized. Various styles of swimming posture information are integrated into the network so as to boost the network better and obtain the contextual information of body parts. CNN is proffered to learn temporal information and optimize the output of the CPMs network. The approach successfully improved the performance of the baseline of this model.

The ways for CPMs to increase the receptive field are max pooling and multistep large convolution kernels. However, max pooling will reduce the resolution of images and result in the loss of details. The high-performance networks have been proposed after CPMs, most of them obey the high-to-low and the low-to-high structures. High-to-low structures are employed to increase the receptive field by reducing the resolution of the inputs which is known as downsampling. The low-to-high structure restores the low-resolution representation to the original resolution, which is known as upsampling.

In order to improve the performance, the networks carry out multiple downsampling or upsampling. These processes are sequentially connected to form the entire network. For example, a symmetrical structure [7] concatenates the high-to-low structure and the low-to-high structure. The Cascaded Pyramid Networks [12] and the Simple Baselines Network [14] took use of cascade heavyweight of the high-to-low structures with the lightweight

low-to-high structure. However, the performance of these networks is also constrained by the information loss caused by downsampling.

The low-level features have higher resolution and contain location, but with lower semantics and more noises. The receptive field of the high-level network is relatively large, the high-level features have stronger semantic information, but the resolution is low and the ability to perceive details is poor, the fusion of these two features improves the performance of the model. A skip path [7] was added to preserve the high-resolution representation before downsampling the image and fuse the feature with the corresponding images in the way of connections and add respectively before and after each upsampling. Similarly, visual features are fused with the same resolution in the process of upsampling and downsampling by means of channel splicing [13]. However, these networks only have an additional unidirectional feature fused from high-resolution to low-resolution or from low-resolution to high-resolution.

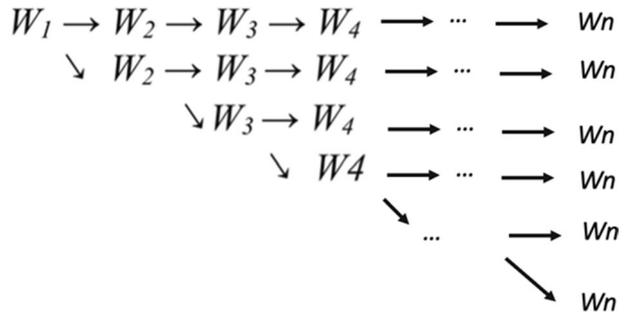
3 Approaches

Swimming poses are specific human poses for swimmers; hence, we detect the joints of swimmers by generating heatmaps. For each group of key points, we generate the corresponding heatmap which includes the possibility of each pixel in the graph to be this class of key points. We calculate the coordinates of point according to the probability. Heatmaps are regressed by using the feature maps, output by HRNet. In this section, we describe the network structure of HRNet and the estimated coordinates of key points of single swimmer [9]. The backbone of HRNet consists of four parallel branch networks. The resolution of the parallel network is halved one by one from top to bottom. The branches exchange information through feature fusion. The network is divided into four stages horizontally. The first stage includes a high-resolution sub-network, a low-resolution network will be added in each subsequent stage. Except the fourth phase, the transition layer is added to the end of each phase of the network to adjust the number of channels.

We take Fig. 1 as an example of the network structure, where W_n represents the sub-network, n indicates the stage of the network. In Fig. 1, subnetwork W_i is related to sub-network W_{i-1} , $i=1, \dots, n$. Fig. 2 shows the changes in the resolution of the network, where REn is used to represent the resolution of the network, n is the stage to which the network belongs to, REi is based on $1/2^{i-1}$ where i is the row number in Fig. 2.

The input of this fusion layer is a collection of feature maps of stage n which has s sub-network $\{F_1, F_2, \dots, F_s\}$. The input maps are aggregated to produce each output map. The

Fig. 1 The proposed network structure



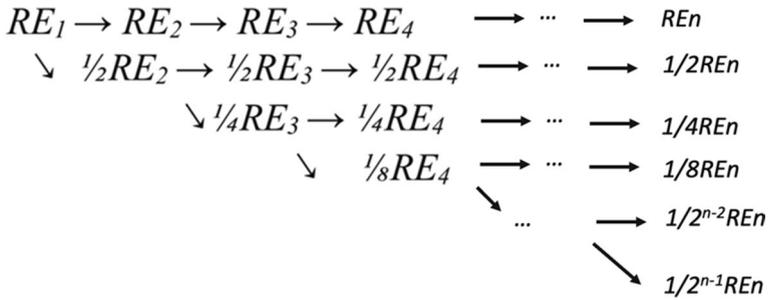


Fig. 2 The changes of resolution in HRNet

collection of output maps are represented as $\{O_1, O_2, \dots, O_s\}$, Eq. (1) shows the specific calculation process, where f refers to the resolution of the output feature image O , if resolution of the input feature map F is lower than f , A represents the upsampling calculation, if the resolution of the input image F is higher than f , which shows the downsampling calculation. If the resolution of the input feature image X is equal to f , which indicates direct replication. The input feature map of the subnet is as same as the resolutions of the output feature map.

$$O_f = \sum_{i=1}^s A(F_i, f) \tag{1}$$

In Fig. 3, the input of this transition layer, after the fusion layer, is the output of the subnetwork with the lowest resolution at this stage. A new channel O_{s+1} is generated in this layer. Eq. (2) shows the specific calculation process, where f_s refers to the resolution O_s .

$$O_{s+1} = A\left(O_s, \frac{1}{2}f_s\right) \tag{2}$$

The first stage contains only one branch network, which mainly includes four residual units. The structure of this residual unit is as same as that of the bottleneck in ResNet-50 (i.e., Residual Neural Network), its width is 64, including a 1×1 convolution for dimensionality reduction, a 3×3 convolution for feature extraction, a 1×1 convolution for

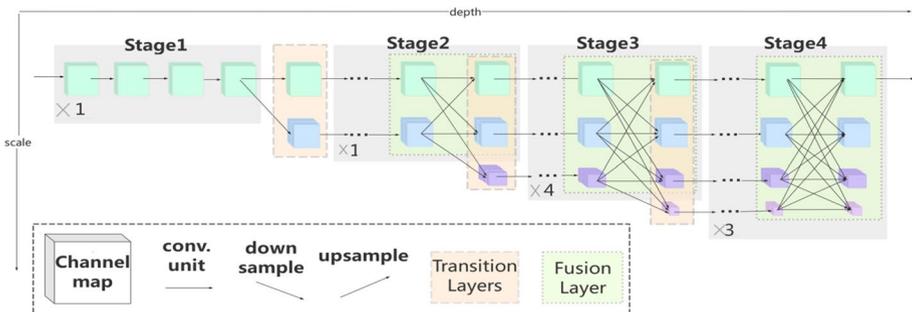


Fig. 3 An overview of HRNet, the duplicated channel maps are represented

dimensionality reduction. The first two convolutions are followed by using Batch Normalization (BN) and Nonlinear Activation Function (ReLU), the last 1×1 convolution without using ReLU, in order to maintain the diversity of visual features. The structure of the jump connection includes two possibilities: If the number of input feature channels is as same as the output, it is added directly; Otherwise, a 1×1 convolution is joined to match the dimension of the output feature, which usually occurs after the resolution is reduced. The structure of this connection is to avoid the vanishing gradient and degradation problems caused by the increase of network layers. The first phase contains only one branch, the fusion module is not included in this phase.

The second, third, and fourth stages are similar, which are all composed of repetitive modules with similar structures. The second stage encapsulates one module, which is composed of two parallel subnetworks. The third stage contains four repetitive modules, each module includes three parallel subnetworks. The fourth stage contains three repetitive modules, the module is composed of four parallel subnetworks. The subnetworks in the modules make up each stage with the same structure, that is, four elements of residual network. The unit of residual network structure is as same as that of the basic block in ResNet-50, which includes two 3×3 convolutions, the first is followed by batch normalization and nonlinear activation ReLU. The second convolution is followed by only one batch normalization, the second, third and fourth stages all include multiple network branches with multiple resolutions, the end of each module in this stage is with fusion layer, to fuse the features at various resolutions. In total, multiscale fusions were carried out for eight times in the second, third, and fourth stages.

Taken the fusion layer of the third stage as shown in Fig. 3, it spans all parallel networks in this stage and merges the characteristic graphs of all sub-networks. The output of each sub-network after passing through the fusion layer is the aggregation of all sub-network inputs. Before aggregation, the characteristics of the same network branch are copied directly. The low-resolution feature map can improve the resolution by using the nearest neighbor upsampling and a 1×1 convolution to obtain the same number of channels as the high-resolution. An appropriate number of stride 3×3 convolution with the stride 2 is applied to reduce the resolution and increase the number of channels of high-resolution feature maps to match that of the low-resolution subnetwork. We do not take use of max pooling for downsampling to avoid the loss of information during dimensionality reduction. Downsampling through stride 3×3 convolutions is able to reduce the loss of visual information.

After the last fusion layer in each stage, we add a transition layer to generate a new subnetwork with more channels. In the transition layer, 3×3 convolutions are applied to adjust the channels and resolution of the network, the resolution of the new subnetwork is halved. In the experiment of this paper, we implemented a small network named HRNet-W32 and a large network named HRNet-W48. The main difference between them is the width of the sub-network. The widths of four parallel subnetworks of HRNet-W32 are 32, 64, 128, and 256, respectively. The widths of the four parallel subnetworks of HRNet-W48 are 48, 96, 192, and 384, respectively [9].

The fourth stage consists of four network branches and outputs four feature maps of multiple resolutions. We abandon the three lower resolution outputs and only take use of the representation of the highest resolution sub-network output to return the heatmap. This hardly affects the performance of the net that can reduce the computational complexity [9] as shown in Fig. 4.

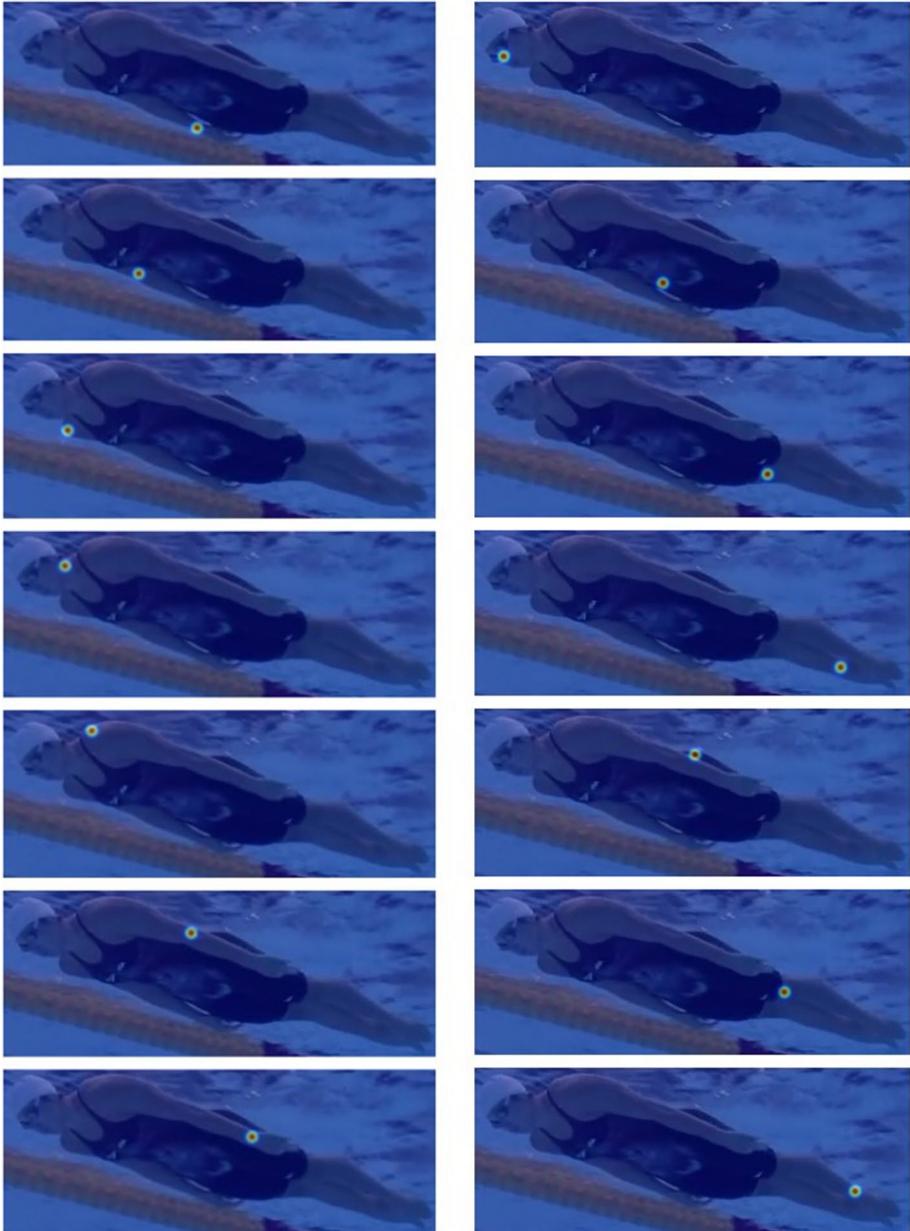


Fig. 4 Heatmap visualization of the 14 key points

4 Experiments

The dataset in this paper was collected by using video footages of swimming events, which recorded the swimmers from an underwater view. The dataset includes human

bodies of both genders, including breaststroke, freestyle, butterfly, and backstroke. In the dataset, there are 2,500 images with 3,615 swimmers that were labelled by using COCO-annotator [22]. Almost all angles of underwater views are covered. Compared with the angle of side view, we observe a complete swimmer's body.

We increase amount of image data in the dataset through data augmentation, including three ways: Randomly rotating the image between -35° and 35° , randomly scaling the image to 0.65 to 1.35 times, and flipping the image horizontally randomly. In order to make the model better adaptive to noises, we added 20.00% noisy images, including water reflection, bubbles, dim light, occlusion, and other factors.

We annotated all image data, in each dataset, only swimmers are labeled. In the dataset, we labelled the classes and locations of all bounding boxes for each image. In the experiment, each dataset was split into training set and test set with the ratio 9:1, then the training set was separated as training set and validation set with the ratio 9:1.

In addition, we also added empty swimming lanes as negative samples. In the pose estimation for swimmers, the facial key points are lower than that of key points on the arms, hence we reduce the number of facial key points of the skeleton. Finally, the proposed model includes 14 key points: Nose, neck, left shoulder, right shoulder, left elbow, right elbow, left wrist, right wrist, left hip, right hip, left knee, right knee, left ankle, right ankle.

We use of the following three different losses for experiments: (1) We define loss function as the Mean Squared Error (MSE) which were applied to compare the predicted heatmaps with the ground truth. (2) With online hard key points mining (OHKM) [14], we sorted the results of MSE output, and screened the top eight key points with the largest loss as the cases for key regression. (3) Bone loss [15] was employed to compare the distance between predicted key points and the distance between ground truth points so as to ensure the model reflects spatial characteristics between key points, such as the length of bone. In the experiment, we took use of bone loss and MSE for joint training.

We define Object Keypoint Similarity (OKS) by using Eq. (3). The main purpose is to calculate the similarity between the true points and the key points predicted by the model.

$$OKS = \frac{\sum_i [\exp(-d_i^2/2s^2k_i^2)\delta(v_i > 0)]}{\sum_i [\delta(v_i > 0)]} \quad (3)$$

where Euclidean distance between the true points and the detected key points is expressed. The scale of this target is represented as s , which shows the square root of the area occupied by the target in the ground truth that is calculated by using the coordinates of bounding box, k_i denotes the normalization factor of the i -th skeletal point. The higher the value k , the worse the annotation effect of this point in the whole dataset, which means, the annotation of this dimension point is much difficult. On the contrary, this point is easier to be marked, v_i is the visibility, the detector does not predict v_i , only v_i in the set. Combined with the function $\delta(\cdot)$, we filter out the marked key points by using the value of v .

Similar to IOU, we were use of OKS to calculate AP (i.e., average precision) and AR (i.e., average recall). We gave a threshold K . If OKS of the target is greater than this threshold, it means that the skeletal point of the current target has been correctly detected. If the OKS of the target is less than the threshold, it means that there is a false detection or omission of the skeletal point of the current target. AP is equal to the proportion of OKS or greater than T . In the experiment, we calculated the following indicators to evaluate the performance of the model: $AP^{0.50}$ and $AR^{0.50}$ if the threshold K is 0.50; $AP^{0.75}$ and $AR^{0.75}$ if the threshold K is 0.75, the average value of AP and AR at ten locations if the threshold K

is 0.50, 0.55, ..., 0.95. The average AR at ten positions if K is 0.50, ..., 0.95. AP^M and AR^M for medium objects, AP^L and AR^L for large objects.

We were use of the Percentage of Correct Keypoints (PCK) to evaluate the detection ability of the proposed model. We define PCK by using Eq. (4), where TP means the number of correctly predicted points. If the distance between the detected key point and the truth point is less than the threshold, the detected point is considered to be correct, the distance is normalized by using the scale of a person [12], T refers to the number of true key points. In the experiment of this paper, we set the threshold as 0.50.

$$PCK = \frac{TP}{T} \quad (4)$$

The dataset is split into a training set and a test set at the proportion 9:1. We set the aspect ratio of the bounding box of a swimmer as 4:3 and cropped the detected swimmer out from the image, we adjusted its size to a fixed resolution 384×288 before importing it into the network. We increased the amount of data in the dataset through data augmentation, including three ways: Rotating the image between -35° and 35° , scaling the image from 0.65 to 1.35, and flipping the image horizontally, all randomly.

Before training, we converted the key points marked in the dataset into heatmaps for subsequent calculation. We took use of Gaussian distribution to generate the heat maps, we set the variance σ as 2. Therefore, in order to speed up the calculations, we only carried out Gaussian distribution with 3σ . The Gaussian variance is 6, the mean of Gaussian distribution is 13.

In the training process, the initial learning rate is set as 1.00×10^{-3} , the decay Gamma is set as 0.10, the decay steps are set as 170 to 200 epochs, the batch size is set as 16. We trained the proposed model based on NVIDIA GeForce RTX2080 super GPU. The training process was ended after the 200-th iteration, with a total time around 12 hours for the main network and around 8 hours for the sub-net.

In the testing part, we harnessed bounding boxes including single swimmer cropped from the input images as the input of the proposed model. We calculated the heatmap by averaging the original image and the flipped image. We trained the net three times by using three loss functions, respectively. We evaluated the performance of the model using indicators such as AP, APM, APL, AR, the results are shown in Table 1. The results of two models for each class of joint points after trained with various loss functions are shown in Fig. 5, where we utilized PCK@0.50 as an evaluation index.

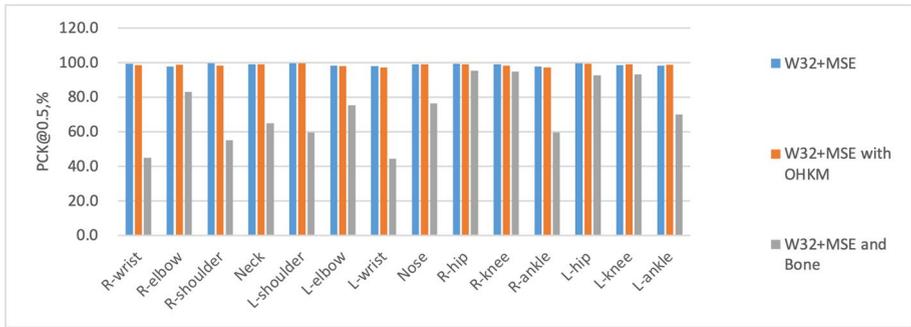
As shown in Table 1, both of two nets were trained with MSE loss that achieved the most ideal outcome. Compared with that of using other loss functions, the large net has obtained the best value with all indexes except AP^M . For example, it obtained AP score 95.30% and AR score 96.60%, but these values are only slightly superior to the results of the small network. The sub-network that was trained with MSE loss obtained AP score 95.20% as well as AR score 96.50%. The sub-nets are ahead of large networks by 1.50 in AP^M .

In addition, in terms of parameter quantity of the model, the parameter quantity of the small network is 28.50M while the large network is 63.60M, which means that the small model has less spatial complexity and thus the small network has less requirement for the data quantity than the large net.

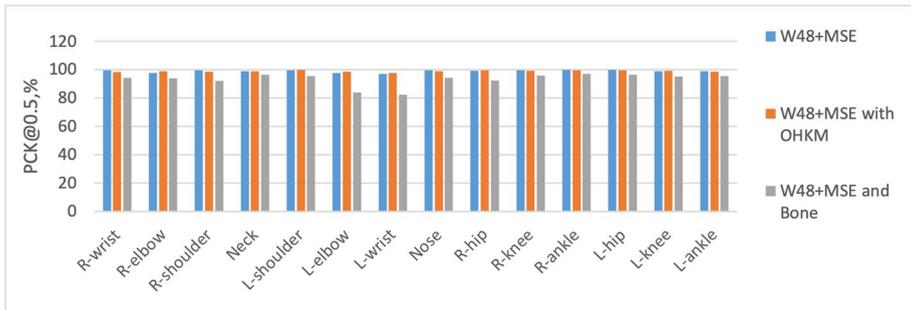
In terms of computational complexity of the proposed model, GFLOPs (i.e., Giga Floating Point Operations Per Second) were 16.00 for the small network and 32.90 for the large network. This illustrates the low computational complexity of the sub-nets, which means

Table 1 The results on test dataset for HRNet-W32 and HRNet-W48

Backbone	#Params	GFLOPs	Loss	AP(%)	AP ⁵⁰ (%)	AP ⁷⁵ (%)	AP ^M (%)	AP ^L (%)	AR(%)	AR ^M (%)	AR ^L (%)
HRNet-W32	28.50M	16.00	MSE	95.20	99.00	98.00	92.40	95.80	96.50	95.70	96.60
HRNet-W32			MSE+OHKM	87.40	99.00	98.00	84.20	88.20	91.20	90.40	91.30
HRNet-W32			MSE+BONE	42.00	95.40	24.80	36.20	44.30	48.60	47.80	48.70
HRNet-W48	63.60M	32.90	MSE	95.30	99.00	99.00	90.90	96.10	96.60	94.10	97.10
HRNet-W48			MSE+OHKM	95.00	99.00	97.00	91.80	95.50	96.40	95.30	96.50
HRNet-W48			MSE+BONE	76.40	99.00	85.00	68.30	78.20	80.90	76.70	81.60



(a)



(b)

Fig. 5 The performance of deep nets for each type of key points using PCK@0.5 metric. (a) HRNet-W32 was trained with MSE, MSE with OHKM, MSE and Bone loss, respectively. (b) HRNet-W48 was trained with MSE, MSE with OHKM, MSE and Bone loss, respectively

that small nets possess lower temporal complexity. Thus, the small network is faster in training and prediction.

Figure 5 shows the performance of HRNet-W48 and HRNet-W32 for each type of key points using PCK@0.50 as the metric. Taken the large network trained with MSE as an example, the net has a strong ability for most of joint points. For example, the scores of large nets based on joint points of shoulders, noses and hips are more than 99.00%. The recognition accuracy of this net for elbow and wrist joints is weak, only a score of about 97.00% is obtained. The reason is that, firstly, the hand joints need to enter and exit the water frequently to complete the strokes, which make the hand joints disappear intermittently in the underwater perspective. Therefore, the number of hand key points in the dataset is less than that of other joint points. Secondly, the bubbles brought by swimmer's arms in the stroke will block the hand joint points, which blurs the hand joints.

The loss of the proposed model was calculated based on the validation set during the training process. As shown in Fig. 6, though the net was trained with MSE loss achieved the best performance, its value loss fluctuated in the training process. This is due to the use of small size of epochs and big learning rate. Therefore, we adjusted the learning rate and retrain the HRNet-W48. We set the initial learning rate as 5.00×10^{-4} , the learning rate is attenuated at 50, 80, 110, 140, 170 and 190 epochs with 0.70 as the decay rate. The

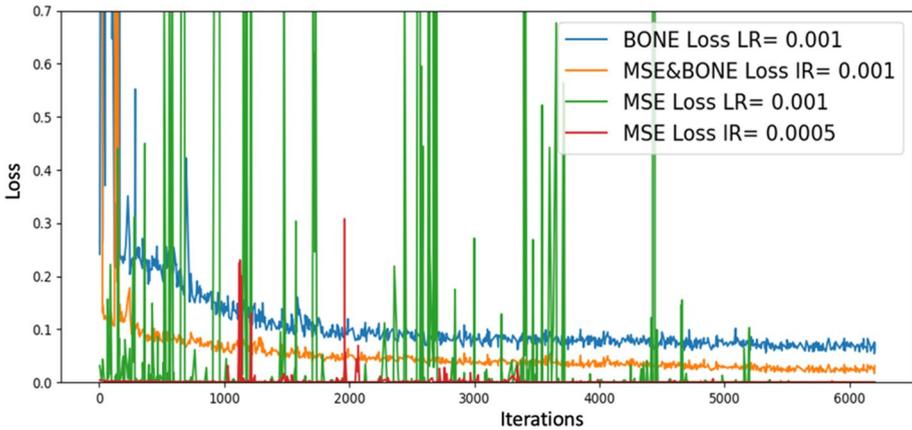


Fig. 6 The loss curves of HRNet-W48 during the training time with various training parameters, where LR refers to initial learning rate

Table 2 The performance of HRNet-W48 with various learning rates

Backbone	HRNet-W48	
Loss	MSE	
Learning rate	1.00×10^{-3}	5.00×10^{-4}
AP	95.30%	95.60%
AP50	99.00%	99.00%
AP75	99.00%	99.00%
AP ^M	90.90%	92.50%
AP ^L	96.10%	96.20%
AR	96.60%	97.10%
AR50	99.70%	99.70%
AR75	99.20%	99.20%
AR ^M	94.10%	95.90%
AR ^L	97.10%	97.30%

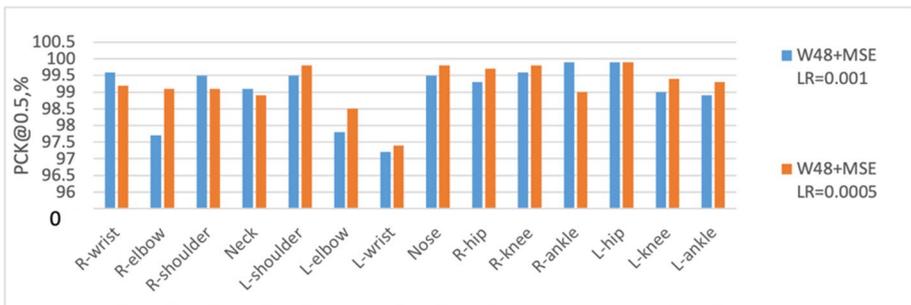


Fig. 7 The performance of model HRNet-W48 that was trained with various learning rates for each type of key points, where LR refers to initial learning rate

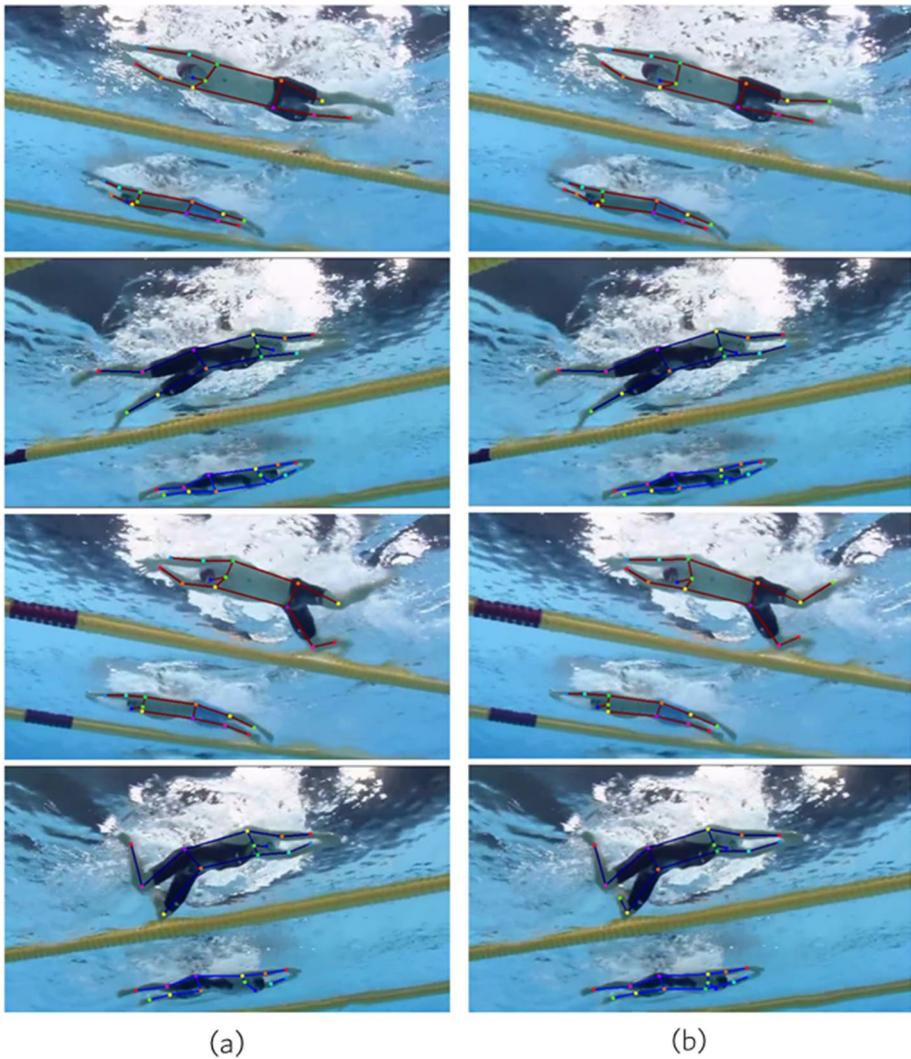


Fig. 8 The result of swimming events (a) The HRNet-W48 model with 1.00×10^{-3} as the initial learning rate (b) The HRNet-W48 model with 5.00×10^{-4} as the initial learning rate

red curve in Fig. 6 shows the changes of value loss during training time, compared to the results using 1.00×10^{-3} as the initial learning rate, which converges better.

The results of HRNet-W48 are shown in Table 2, which express that the new model that was trained with a smaller learning rate that performs better on most evaluation indicators. Figure 7 shows the performance of the two models for each type of key points, the new model obtains stronger capability for identifying the key points, such as elbow, wrist, and ankle. As shown in Fig. 8, the new model accurately detects missing joint points of the previous model on ankles and wrists. In general, the proposed model still achieved ideal detection results after training with a small

amount of data. This proves that the proposed method is very practical in swimmer pose estimation.

5 Conclusion

In this paper, we implemented a swimmer pose estimation model based on deep learning. HRNet is the core net of the proposed model which innovatively changed the link between high and low-resolution from series to parallel, thus retained the high-resolution representation in the whole network structure. Compared with the swimmer pose estimation implemented by using traditional methods in the past, our proposed model supports parallel computing such as GPU processing, which greatly shortens the running time. The proposed network has a strong recognition ability for most of joint points. The scores of large networks based on joint points of shoulders, noses and hips are more than 99.00%.

In conclusion, the proposed model has achieved excellent results in speed and accuracy, which deploys a great foundation for other machine vision tasks. Compared to conventional methods for human pose estimation, the deep learning methods overcame the inherent limitations of a machine learning classification approach and realized the end-to-end design purpose in deep learning.

In the future, we plan to adjust the structure of the existing models so as to obtain higher detection speed and combine target detection to estimate the pose of multiple swimmers [16–21]. The pose estimation of single person is the basis of machine vision algorithms. This plays a pivotal role in multiplayer scenarios. In addition, the swimmer's pose correction based on pose estimation is also an important research direction of the future work. In future, we may apply this methodology to other movement analyses of other sport games. It is also very helpful for VR and AR industries, especially for swimming simulations and computer-aided coaching.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions

Data availability Data sharing not applicable to this article as no datasets were generated or analyzed during the current study

Declarations This work has not any funding support, it has not any conflicts of interests or competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Lienhart R, Einfalt M, Zecha D (2018) Mining automatically estimated poses from video recordings of top athletes. *Int J Comput Sci Sport* 17(2):4–112

2. Zecha D, Greif T, Lienhart R (2012) Swimmer detection and pose estimation for continuous stroke-rate determination. Research Report, Institut für Informatik
3. Zecha D, Eggert C, Lienhart R (2017) Pose estimation for deriving kinematic parameters of competitive swimmers. *Electron Imaging* 2017(16):21–29
4. Einfalt M, Zecha D, Lienhart R (2018) Activity-conditioned continuous human pose estimation for performance analysis of athletes using the example of swimming. In: *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp 446–455
5. Zheng C et al (2020) Deep learning-based human pose estimation: A survey. *ACM Comput Sur* 56(1):1–37
6. Wei S, Ramakrishna V, Kanade T, Sheikh Y (2016) Convolutional pose machines. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp 4724–4732
7. Newell A, Yang K, Deng J (2016) Stacked hourglass networks for human pose estimation. In: *European Conference on Computer Vision*, Springer, pp 483–499
8. Çiçek Ö, Abdulkadir A, Lienkamp S, Brox T, Ronneberger O (2016) 3D U-Net: Learning dense volumetric segmentation from sparse annotation. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, pp 424–432
9. Sun K, Xiao B, Liu D, Wang J (2019) Deep high-resolution representation learning for human pose estimation. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp 5693–5703
10. Felzensz W et al (2010) Object detection with discriminatively trained part-based models. *IEEE Trans Pattern Anal Mach Intell* 32(9):1627–1645
11. Toshev A, Szegedy C (2014) DeepPose: Human pose estimation via deep neural networks. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp 1653–1660
12. Xiao B, Wu H, Wei Y (2018) Simple baselines for human pose estimation and tracking. In: *European Conference on Computer Vision (ECCV)*, pp 466–481
13. Ronneberger O, Fischer P, Brox T (2015) U-Net: Convolutional networks for biomedical image segmentation. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, pp 234–241
14. Chen Y, Wang Z, Peng Y, Zhang Z, Yu G, Sun J (2018) Cascaded pyramid network for multi-person pose estimation. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp 7103–7112
15. Kulon D, Guler R, Kokkinos I, Bronstein M, Zafeiriou S (2020) Weakly-supervised mesh-convolutional hand reconstruction in the wild. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp 4990–5000
16. Cao X (2022) Pose estimation of swimmers from digital images using deep learning. Auckland University of Technology, New Zealand, Master's Thesis
17. Zhang F, Zhu X, Wang C (2021) Single person pose estimation: A survey. *ACM Comput Surv* 56(1):1–37
18. Yu Z, Yan W (2020) Human action recognition using deep learning methods. In: *Proceedings of IEEE IVCNZ*
19. Parekh P, Patel A (2021) Deep learning-based 2D and 3D human pose estimation: A survey. In: *International Conference on Computing, Communications, and Cyber-security*, Springer, pp 541–556
20. Yan W (2023) *Computational methods for deep learning*. Springer
21. Yan W (2019) *Introduction to intelligent surveillance*. Springer
22. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Zitnick CL (2014) Microsoft COCO: Common objects in context. In: *European Conference on Computer Vision*. Springer, pp 740–755

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.