Pattern Recognition 147 (2024) 110111

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Vital information is only worth one thumbnail: Towards efficient human pose estimation

Zian Zhang[1], Yongqiang Zhang [*,1], Yin Zhang, Rui Tian, Mingli Ding

School of Instrumentation Science and Engineering, Harbin Institute of Technology, Harbin 15001, China

ARTICLE INFO

Keywords:
Human pose estimation
Small-size input
Knowledge distillation
Network compression and acceleration

ABSTRACT

In pursuit of impressive performance, existing DCNN-based approaches of human pose estimation usually use massive networks and large-size images to train a deep model. When applying these deep based methods in real-time systems, current works try to compress the deep network by reducing the number of layers and channels, but such approaches are complex and poorly generalized since they require elaborate design of small-scale network structures. Based on the fact that large-size images contain redundant information, in this paper, we explore the influence of image-size on system complexity and propose a novel framework called **ThumbPose** to accelerate and compress deep models by inferring on **thumbnail representations** in the task of human pose estimation. In our framework, we first propose a **style supervised online downscaler** to reduce an input image into a thumbnail image. Furthermore, a training strategy of **dual-branch auto-encoding** is designed to obtain effective and accurate thumbnail representation in a knowledge distillation manner, which is further used to maintain the performance of thumbnail images as the original-size input images. For heat-map based human pose estimation, ThumbPose is an orthogonal and implementation-friendly method, that can not only compress and accelerate the inference network but also obtain an image downscaler in a supervised manner that can be used in other high-level tasks (*e.g.* detection, segmentation, *etc.* in practical applications). Extensive experiments on MS COCO dataset demonstrate the effectiveness of our proposed method, and ThumbPose achieves superior performance (+ 1.3% AP and + 0.7% AR) with negligible additional cost (<0.2 GFLOPs) compared to previous state-of-the-art methods when using small-size images as inputs. Moreover, experiments on MPII show that our model achieves higher accuracy (+ 0.2% Mean@0.5) with minimal computation (2.5 GFLOPs) compared to superior lightweight models obtained by the network compression methods.

1. Introduction

Human pose estimation is a fundamental problem in computer vision, since it is the basic technology for some advanced tasks such as action recognition [1–3], human motion prediction [4–6] and intelligent monitoring [7]. Recent researches of human pose estimation have achieved superior accuracy by employing massive fully convolutional networks (FCN) to infer on larger-size images. Despite impressive performance has been achieved, heavy consumption of computation and memory makes it difficult to be used in real-time applications.

To accelerate the network of human pose estimation, existing works [8,9] compress a deep or wide model into a narrow (with fewer channels) or shallow (with fewer layers) network, and knowledge distillation [10] is another type of method to maintain the accuracy. While encouraging results have been observed, these methods are limited in the following aspects: (1) *Simplicity*. The powerful fitting ability of an FCN is highly coupled to its ingenious structure, so it is necessary to elaborately design a shrinking strategy to retain its structural characteristics [11,12]. (2) *Universality*. Existing lightweight models are obtained by reducing the number of stacked blocks of Hourglass [8,9,13], as shown in Fig. 1(a). Such methods can be effectively applied to specific networks, but cannot be generalized to other advanced models, *e.g.* HRNet [14], *etc*.

Actually, a promising method for reducing the computational complexity of deep models is to infer on a small-size image. As the spatial size of feature maps decreases, the required computation is diminished, and the memory required to accommodate those feature maps will also be reduced. We believe that the above viewpoint can address the limitations of network compression. First, this method only needs to simply reduce the size of input images, which avoids weakening the representation ability of networks. Moreover, this method can be

* Corresponding author.
E-mail address: zhangyongqiang@hit.edu.cn (Y. Zhang).
[1] Equal contribution.

https://doi.org/10.1016/j.patcog.2023.110111
Received 23 July 2022; Received in revised form 31 October 2023; Accepted 9 November 2023
Available online 11 November 2023

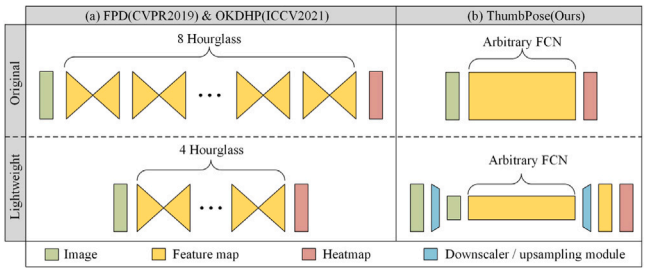


Fig. 1. Methods of obtaining lightweight models, (a) FPD [8] and OKDHP [9] compress an 8-stage Hourglass into a 4-stage Hourglass. (b) The proposed ThumbPose reduces the size of the input image by a downscaler, and enlarges the size of the heat-map by an up-sampling module.

generalized to advanced models since FCN allows arbitrary-size input. Nevertheless, directly reducing the image size will inevitably lose some key information, which leads to a sharp decline in the final estimation performance as reported in [15,16].

As the above analyzed, the network-compression based methods are effective but usually limited in both simplicity and universality. By contrast, the input-reduction based methods are simple and generalizable but lead to inferior performance. Some efforts based on input-reduction have been attempted in classification [17] and object detection [18], hence we try to extend this technique for the task of efficient human pose estimation. To accelerate the model of human pose estimation in an effective, simple and generalizable manner, in this paper, we propose a novel framework called **ThumbPose** to accelerate and compress FCN-based networks by inferring on the **thumbnail representation**, as shown in Fig. 1(b). Specifically, we present a **style supervised online downscaler** to reduce the size of an input image into a thumbnail image as lossless as possible. The downscaler removes the redundant information in the original-size image and therefore allows us to represent all key information in a thumbnail image. Meanwhile, in order to obtain effective and accurate thumbnail representations, we design a training strategy of **dual-branch auto-encoding** to explicitly incorporate knowledge distillation into ThumbPose. This strategy enables the thumbnail representation contains the same information as its original-input counterpart by aligning feature maps, so that the prediction accuracy can be further retained.

Contributions. For heat-map based human pose estimation, ThumbPose is an orthogonal and implementation-friendly method of model compression and acceleration, that can not only address the limitations of network compression in simplicity and universality, but also achieve an impressive performance. Moreover, the well-trained image downscaler obtained by ThumbPose can be used in other high-level tasks (*e.g.* detection [19,20] and segmentation [21]) in practical applications. To sum up, we make the following contributions:

(1) For efficient human pose estimation, we propose a novel framework ThumbPose to decrease computation and memory consumption by inferring on thumbnail representations, which addresses the inherent limitations in simplicity and universality of previous network compression approaches.

(2) We adopt a style supervised online downscaler to adaptively filter out the redundant information in an original-size image, and compress the image into a thumbnail image as low-distortion as possible.

(3) We design a dual-branch auto-encoding strategy to further understand thumbnail information in a knowledge distillation manner. Furthermore, a novel distribution-aware distillation method is exploited to guide the learning of background knowledge.

(4) Extensive experiments on MS COCO [22] demonstrate that ThumbPose achieves superior performance (+1.3% AP and +0.7% AR) with negligible additional cost (<0.2 GFLOPs) as compared to previous state-of-the-art methods when using small-size images as inputs. Moreover, when conducting experiments on MPII [23], our model achieves higher accuracy (+0.2% Mean@0.5) with minimal computation cost (2.5 GFLOPs) compared to existing superior lightweight models obtained by network compression methods.

The rest of the paper is organized as follows. We review the related work in Section 2. In Section 3, the architecture of our ThumbPose, the principle and implementation of the style supervised online downscaler, and the details of the dual-branch auto-encoding strategy are described. In Section 4, we present some experiments and ablation studies to validate the effectiveness of our proposed method on two widely used human pose estimation benchmarks (*i.e.* MS COCO and MPII). Finally, the conclusions and further work are provided in Section 5.

2. Related work

2.1. Human pose estimation

Convolutional neural networks (CNNs) are widely used in the task of human pose estimation [13–15,24–31], and they have greatly improved prediction accuracy in recent years. The methods of human pose estimation can be divided into two categories: heat-map based methods [14,15,26–29,32,33] and regression-based methods [24,34–37]. Heat-map based methods first predict pixel-wise heat-maps, and then calculate the joint coordinates according to the heat-maps. Regression-based methods directly regress the coordinates of joints from the outputs of backbones, and these methods are faster but less accurate than heat-map based approaches.

Currently, most existing SOTA methods are heat-map based methods, which tend to improve the performance by stacking deeper and wider network architectures. For instance, Hourglass [13] stacks high-to-low and low-to-high blocks to enhance the heat-map estimation quality. Simple Baseline [27] stacks convolutional and transposes convolutional layers to a simple single-stage architecture, and achieves an impressive performance. HRNet [14] proposes a high-resolution parallel representation network in order to provide precise heat-maps. DARK [15] proposes a principled method of coordinate representation for significantly improving the performance of existing models. UDP [38] explores an unbiased data processing method for accurate human pose estimation. Graph-PCNN [16] designs a graph-based refining strategy to boost the accuracy of key-points detection. TokenPose [29] uses tokens to represent each joint entity and a transformer architecture is introduced to achieve competitive performance. UniFormer [30] integrates the merits of convolution and self-attention in a concise transformer format, and it also achieves competitive accuracy in human pose estimation.

Although encouraging performance has been achieved in existing methods, however these approaches are limited when using in real-time systems due to the heavy consumption of resources. In order to meet the needs of practical applications, in this paper, we design a novel framework called ThumbPose, which enables networks to detect joints in an efficient and effective manner.

2.2. Model compression and acceleration

Methods of model compression and acceleration are proposed to reduce the network size and computation cost, because of the increasing demand of real-time applications in artificial intelligence. Early methods yielded sparse weights by network pruning [39–41], which defined some mechanisms to prioritize the parameters and set unimportant ones to zeros. However, these methods usually require specific implementations for acceleration. Hereby, some works are proposed to prune filters from a whole perspective [42,43], and these strategies achieve efficient inference speed in common conditions. Beyond pruning, quantization methods [39,44] use smaller bit number of weight to reduce consumption, where the extreme case is binarization [45,46]. Moreover, some methods are proposed to reduce the scale of filters

by low-rank approximation [47–49]. In addition, neural architecture search is also used to automatically search a lightweight network architecture [50] for model compression and acceleration, and it is effectively applied to human pose estimation [51]. However, the above methods require complex processing or hand-crafted designing, which is not suitable for practical applications.

In recent years, knowledge distillation has been widely used in model compression for various visual tasks [10,52–55], which is a strategy to guide a compressed network (*i.e.* student network) to learn the knowledge extracted from a larger teacher network. For example in human pose estimation, FPD [8] compresses an 8-stage Hourglass into a 4-stage network and first uses knowledge distillation to maintain its accuracy. OKDHP [9] further improves the accuracy of the lightweight Hourglass by online knowledge distillation. However, it is difficult to generalize these targeted works (*e.g.* FPD and OKDHP) to advanced models (*e.g.* HRNet [14], *etc.*), since it is needed to carefully designed a network shrinking strategy. Actually, all the above mentioned approaches neglect a fact that the spatial size of the feature map plays a significant role in the overall complexity of deep networks.

A hypothesis proven in classification [17,56–58] and object detection tasks [18,59] is that as the spatial size of feature maps decreases, the required computation is diminished, and the memory required to accommodate those feature maps will also be reduced. For classification, ThumbNet [17] first compresses the input images and utilizes knowledge distillation to improve the accuracy of small-input-sized model. For object detection, ThumbDet [18] exploits an image down-sampling module to accelerate the detection model and performs knowledge distillation by aligning both transformer tokens and predicted logits of teacher and student models. For human pose estimation, some works [8,9] try to reduce computation and memory consumption by using small-size images as the network inputs, but the performance of models is sharply decreased. We believe that this type of approach is advanced since it only needs to reduce the size of the input image, which avoids elaborate designing of compression strategies. To improve the accuracy of low-resolution human pose estimation, CAL [60] proposes a confidence-aware learning method, but this work greatly increases the computation of the model.

In order to obtain a high-accuracy lightweight model of human pose estimation in a simple manner, using thumbnail images as inputs is a promising way. However, it is difficult to directly use the method of ThumbNet [17] or ThumbDet [18] in the task of human pose estimation. When inputting original size images into the teacher model and inputting thumbnail images into the inference model, the heat-map dimensions of these two networks are different, so we propose a strategy of dual-branch auto-encoding for aligning the output dimensions. In addition, previous works rarely discuss the underlying principles of image down-scaling, hence we rethink the essential reasons for the performance degradation when using small-size images as inputs, and then propose a style supervised online downscaler to generate the thumbnail inputs for the model of efficient human pose estimation.

2.3. Auto-encoder

Auto-encoder is a popular framework, that is widely used in image reconstruction [61], restoration [62] and super-resolution [63], *etc.* In the architecture of auto-encoder, an encoder maps an input image to a low-dimensional latent code, and a decoder reconstructs a pre-processed (*e.g.* scaled, filtered, colored, *etc.*) input from the latent code. Early auto-encoder methods usually use a fully connected structure for distribution mapping [64], however they are unfriendly to visual tasks. To infer on images efficiently and effectively, researchers use a series of convolutional layers to build a convolutional auto-encoder for different tasks [65,66]. In this paper, inspired by the convolutional auto-encoder, we design a strategy of dual-branch auto-encoding to explicitly incorporate knowledge distillation into network training, which enables us to obtain effective thumbnail representation and further maintain the thumbnail-input performance as the original-size inputs.

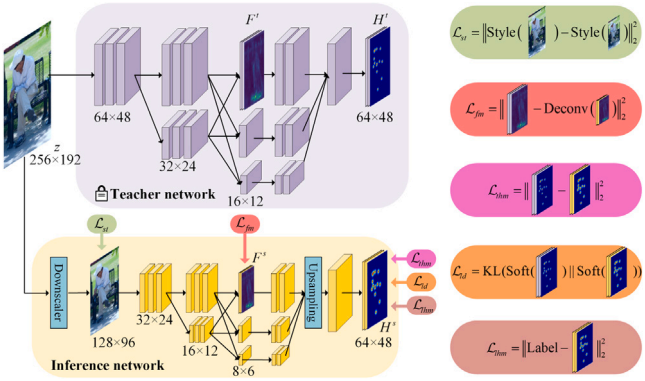


Fig. 2. The architecture of ThumbPose. Teacher network: the architecture of the well-trained teacher network. Inference network: the architecture of the inference network. Downscaler and Up-sampling: the modules for scale transformation. The numbers below each image and feature map are their spatial resolutions. The symbol of lock indicates the parameters are fixed. Rounded rectangles with arrows represent the four losses and subjects of their supervision, *i.e.* supervise style of thumbnail image by $\mathcal{L}_{st}$ (green), supervise intermediate feature maps by $\mathcal{L}_{fm}$ (light red), supervise heat-maps by $\mathcal{L}_{thm}$ (magenta), $\mathcal{L}_{ld}$ (orange) and $\mathcal{L}_{lhm}$ (brown). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3. Proposed method

In this section, we will introduce our proposed efficient human pose estimation method in detail. First, we present the overview of our proposed ThumbPose, and instantiate this framework for clearer expression. Then, the principle and implementation of the style supervised online downscaler are described. In addition, we represent each part of the dual-branch auto-encoding strategy in detail, *i.e.* a regularization approach for training, a distribution-aware heat-map reconstruction method and the overall training process.

3.1. Overview of ThumbPose

Most heat-map based human pose estimation methods utilize FCN as the backbone, and an inference head (1×1 convolution) is used to predict the final results, *i.e.* pixel-wise heat-maps. In these methods, the ground-truth labels of joints are represented by K heat-maps (where K is the number of human joints, $K \in \mathbb{Z}$) $H^l = \{H^l_1, H^l_2, \ldots, H^l_K\}$. Heat-map H^l_k indicates the location confidence of the kth labeled joint, and it is a confidence map obtained by centering a Gaussian kernel around the labeled joint position (x_k, y_k), which can be formulated as:

$$H^l_k(x, y) = \exp \frac{-\left((x - x_k)^2 + (y - y_k)^2\right)}{2\sigma_h^2}, \tag{1}$$

where (x, y) denotes a pixel location, and σ_h is the standard deviation of the Gaussian kernel. Heat-maps H^l are used as the ground-truths to supervise the training of the network.

In this paper, we choose an original-size inputted HRNet as a teacher network to guide the training of an inference network. The overall architecture of our method is shown in Fig. 2, which consists of the following components: a teacher network, an inference network, a supervision $\mathcal{L}_{st}$ from the style of original images, a supervision $\mathcal{L}_{fm}$ from intermediate feature maps of the teacher network, a supervision $\mathcal{L}_{thm}$ from heat-maps of the teacher network, a supervision $\mathcal{L}_{ld}$ from a probability distribution which is generated by softening and normalizing the heat-maps of the teacher network, and a supervision $\mathcal{L}_{lhm}$ from heat-maps of GT labels.

Specifically, when training the inference network, the original-size (*i.e.* 256×192 pixels) image z is input into the teacher and inference network simultaneously for online guidance, where the teacher network is well-trained on original-size images and the parameters are

fixed during training. For the first stream (*i.e.* the teacher network), ThumbPose passes z through the stacked convolutional layers, and then the generated intermediate feature maps F^t (*i.e.* 64 × 48 pixels) and heat-maps H^t (*i.e.* 64 × 48 pixels) are extracted for online guidance to train the inference network. For the second stream (*i.e.* the inference network), the downscaler is employed to reduce z into a thumbnail image (*i.e.* 128 × 96 pixels), and then ThumbPose passes the thumbnail image through convolutional layers and an up-sampling module of the inference network to generate the final heat-maps H^s (*i.e.* 64 × 48 pixels). In addition, intermediate feature maps F^s (*i.e.* 32 × 24 pixels) are extracted from the inference network for calculating $\mathcal{L}_{fm}$, hence a de-convolutional layer is designed to re-size F^s to the size of F^t.

3.2. Style supervised online downscaler

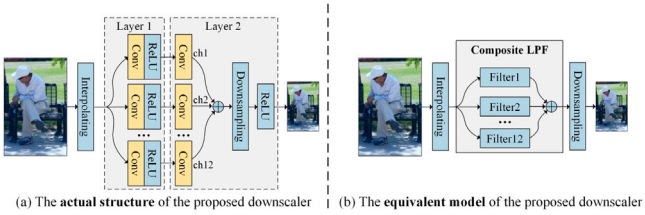
A mainstream method of image down-scaling is the affine transformation, and it can be decomposed into an interpolating and a down-sampling operation, and the factorization of down-scaling $\mathbf{T_s}$ is as follows:

$$\begin{aligned}\mathbf{T_s} &= \begin{bmatrix} S_w & 0 & 0 \\ 0 & S_h & 0 \\ 0 & 0 & 1 \end{bmatrix}\Bigg|_{0<S_*<1|S_*\in\mathbb{Q}} = \begin{bmatrix} \frac{I_w}{D_w} & 0 & 0 \\ 0 & \frac{I_h}{D_h} & 0 \\ 0 & 0 & 1 \end{bmatrix}\Bigg|_{0<I_*<D_*|I_*,\,D_*\in\mathbb{Z}} \\ &= \begin{bmatrix} \frac{1}{D_w} & 0 & 0 \\ 0 & \frac{1}{D_h} & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} I_w & 0 & 0 \\ 0 & I_h & 0 \\ 0 & 0 & 1 \end{bmatrix} = \mathbf{T_d}\,\mathbf{T_i},\end{aligned} \tag{2}$$

where h and w denote the height and width of an image respectively, S_* is the scale coefficient, $\mathbf{T_i}$ and I_* denote the processing operation and coefficient of interpolation, and $\mathbf{T_d}$ and D_* denote the processing operation and coefficient of down-sampling. In the processing of S_*-fold down-scaling, the inputted image is first interpolated by an I_* factor, and then is D_*-fold down-sampled. It has been proven by the Nyquist-Shannon sampling theorem [67] that spectrum aliasing will lead to image distortion in down-sampling, thus affine-based down-scaling methods may lose key information of original images, which is not conducive to obtaining accurate predictions. For aliasing distortion, a conventional solution is proposed to add an anti-aliasing low-pass filter (LPF) before down-sampling, but this is complicated because it requires careful selection of filters and design of its parameters. In this paper, we instead exploit CNN to adaptively filter out the aliased components and then down-sample a large-scale image into a thumbnail image.

To this end, we propose a style supervised online downscaler, as shown in Fig. 3(a), which comprises an interpolating operation and two convolutional layers, where each convolutional layer contains a convolutional operation followed by an ReLU, and the interpolating operation employs traditional methods, *e.g.* bi-linear and bicubic [68]. In order to strengthen the filtering ability, we use multiple convolution kernels to expand a 3-channel interpolated image into 12-channel feature maps in the first convolutional layer, and then the second convolutional layer is used to reconstruct a 3-channel thumbnail image. Note that the second convolutional layer can realize image down-sampling by setting the stride parameters. In the downscaler, the interpolated image passes through 12 channels in parallel, which essentially is a structure of composite filter, *i.e.* Fig. 3(b). Moreover, the trainable characteristic enables the composite filter to adaptively evolve advanced low-pass filtering ability, and it can further retain the information of the original-size image as low-distortion as possible by effectively filtering out those aliasing components.

We hope that the information in the color channels of the generated small-size image should not be destroyed or misaligned, since that: (1) As a substitute of small-size images obtained by traditional methods, the visual characteristics of small-size images generated by our proposed downscaler should be similar to the original images. (2)



(a) The **actual structure** of the proposed downscaler (b) The **equivalent model** of the proposed downscaler

Fig. 3. The structure of our proposed style supervised online downscaler. (a) The downscaler comprises an interpolating operation and 2 convolutional layers, and each convolutional layer is a convolutional operation (with 5 × 5 convolution kernel) followed by a ReLU. The numbers of convolution kernels in the first layer and the second layer are 12 and 3 respectively, and scale reduction occurs in the second layer. (b) Through an equivalent transformation to execution order, the proposed downscaler can be viewed as the equivalent model illustrated in this figure. Here, 2 convolutional layers can be regarded as a composite filter (*i.e.* a combination of 12 parallel adaptive filters) followed by a down-sampling operation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The model performance may be decreased when using non-RGB images as inputs, since the backbones of human pose estimation are usually pre-trained on a large-scale RGB image dataset (*e.g.* ImageNet [69]). Therefore, we aim to explicitly guarantee the visual authenticity of the generated thumbnail images by adding a supervision for the color similarity between the thumbnail and original images. To quantify this color similarity, we follow the task of style transfer [70–72] to use the style, *i.e.* the mean and variance of the pixel distribution in each color channel, of an image as the representation for its visual characteristic. Specifically, we supervise the training of the proposed downscaler via a specially designed loss function $\mathcal{L}_{st}$, which is formulated as follows:

$$\mathcal{L}_{st} = \frac{1}{3}\sum_{i=1}^{3}\left(\|\mu(z_i) - \mu(\phi_s(z)_i)\|_2^2 + \lambda\|\sigma^2(z_i) - \sigma^2\left(\phi_s(z)_i\right)\|_2^2\right), \tag{3}$$

where i denotes the serial number of channel, $\phi_s(\cdot)$ denotes all of the nested operations in our downscaler, $\mu(\cdot)$ and $\sigma^2(\cdot)$ denote the mean and variance respectively which are computed across spatial dimensions independently for each batch and feature channel, and λ is a hyper-parameter that balances the two statistics. The proposed style supervision $\mathcal{L}_{st}$ encourages that the distribution in each color channel of the thumbnail image stays close to its counterpart in the original image, leading to a visually realistic thumbnail image. Apart from the supervision by $\mathcal{L}_{st}$, the downscaler is incorporated into the whole architecture of ThumbPose and trained together with other components in an end-to-end manner. Such an online training approach provides results-driven supervision, which instructs the downscaler to filter out noise and then generate thumbnail images with discriminative information for accurate prediction.

3.3. Dual-branch auto-encoding for effective thumbnail representation

Although a low-distortion thumbnail image is obtained by using our style supervised downscaler, it is not guaranteed that all of the key information is contained in such a small-size image. In order to maintain performance, it is taken for granted that the thumbnail feature maps of the inference network contain the same information as their original counterparts. To this end, we employ a teacher network to guide the inference network in a knowledge distillation manner.

Specifically, the intermediate feature maps F^s of the inference network are first selected as a monitoring site, and then we design a dual-branch auto-encoding strategy to incorporate all five supervisions (*i.e.* $\mathcal{L}_{st}$, $\mathcal{L}_{fm}$, $\mathcal{L}_{thm}$, $\mathcal{L}_{ld}$ and $\mathcal{L}_{lhm}$) into the training process of the inference network, as shown in Fig. 4. In the architecture of the dual-branch auto-encoder, the Encoder maps the original-size image z to F^s, the FDecoder maps F^s to a space with the same dimension as their counterparts F^t in the teacher network, and F^s is also mapped to the predicted heat-maps H^s by the HDecoder. In order to enhance

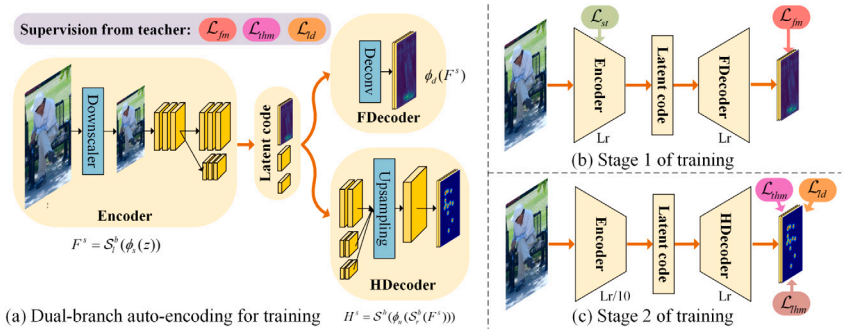


Fig. 4. Architecture of the dual-branch auto-encoder. (a) During training, the inference network is essentially a dual-branch auto-encoder, where F^s is a latent code, $\mathcal{S}_l^b(\phi_s(\cdot))$ is an encoder named Encoder, $\phi_d(\cdot)$ is a decoder named FDecoder, $\mathcal{S}_r^b(\cdot)$ denotes the nested functions in the inference backbone in addition to $\mathcal{S}_l^b(\cdot)$ and $\mathcal{S}^h(\phi_u(\mathcal{S}_r^b(\cdot)))$ is another decoder named HDecoder. The teacher network guides the inference network by $\mathcal{L}_{fm}$ and $\mathcal{L}_{thm}$ losses. (b) In stage 1, the Encoder-FDecoder network is supervised by $\mathcal{L}_{st}$ and $\mathcal{L}_{fm}$ losses. (c) In stage 2, the Encoder-HDecoder network is supervised by $\mathcal{L}_{thm}$, $\mathcal{L}_{ld}$ and $\mathcal{L}_{thm}$ losses. In the stage 1 and stage 2, Lr is the base learning rate, and it is dropped into Lr/10 for the Encoder.

the thumbnail representation, we divide the training process into two stages: stage 1 regularizes the Encoder, and stage 2 trains the Encoder-HDecoder network for reconstructing distribution-aware heat-maps in an end-to-end manner.

3.3.1. Regularization for the encoder

In the Encoder-FDecoder network, a de-convolutional layer (*i.e.* a de-convolutional operation followed by a ReLU) is used to enlarge F^s into the same dimension as their counterparts F^t in the teacher network, as shown in Fig. 4(a), so that the information gap of feature maps between the inference and teacher networks can be narrowed by aligning the enlarged F^s to F^t, and the corresponding loss $\mathcal{L}_{fm}$ is computed by:

$$\mathcal{L}_{fm} = \frac{1}{N}\sum_{i=1}^{N} \| \mathcal{T}_l(z)_i - \phi_d\left(\mathcal{S}_l^b\left(\phi_s(z)\right)\right)_i \|_2^2, \tag{4}$$

where N denotes the number of channels in the intermediate feature maps, $\phi_d(\cdot)$ denotes the nested functions in the de-convolutional layer, $\mathcal{T}_l(\cdot)$ denotes the mapping from z to F^t, and $\mathcal{S}_l^b(\cdot)$ maps the thumbnail image into F^s. This stage (*i.e.* stage 1) is essentially a regularization for the Encoder, which enables the Encoder to extract most of the same information as contained in F^t. To assist the information extracted from the Encoder, the style based loss $\mathcal{L}_{st}$ is also used to supervise the downscaler in this stage.

3.3.2. Distribution-aware heat-map reconstruction

Although the regularization in stage 1 enables the thumbnail representation to contain almost the same information as the original large-scale counterpart in the teacher network, some key information may still be lost since the capacity of F^s is much lower than F^t (*i.e.* quarter in our setting). To achieve more valuable information in the thumbnail representation, we further refine information by enabling the HDecoder to reconstruct the heat-maps H^t of the teacher network from F^s. In particular, an up-sampling module is inserted between the backbone and the head of the inference network (as shown in Fig. 4(a)) to enlarge H^s to the same size as H^t, where the up-sampling module comprises three layers: the first layer is a bi-linear interpolating, the second layer is a convolutional operation followed by batch normalization (BN), and the third layer is a convolutional operation followed by BN and ReLU. The loss $\mathcal{L}_{thm}$ is used to supervise the reconstruction of H^t by the Mean Square Error (MSE) function, which is formulated as:

$$\mathcal{L}_{thm} = \frac{1}{K}\sum_{k=1}^{K} \| \mathcal{T}(z)_k - \mathcal{S}^h\left(\phi_u\left(\mathcal{S}^b\left(\phi_s(z)\right)\right)\right)_k \|_2^2, \tag{5}$$

where $\mathcal{T}(\cdot)$ denotes the mapping from z to H^t, $\mathcal{S}^h(\cdot)$ and $\mathcal{S}^b(\cdot)$ denote the nested functions in the head and backbone of the inference network respectively, and $\phi_u(\cdot)$ denotes nested operations in the up-sampling module.

We experimentally find that there are several regions with small response values in the background of H^t, and we believe that these regions contain dark knowledge which is helpful to guide the inference network to extract vital information. However, smaller response values than the target joints make the dark knowledge difficult to be perceived by $\mathcal{L}_{thm}$. In [73], the localization distillation is proposed to alleviate a similar problem in object detection by adaptively enhancing the values of smaller responses. To apply a similar idea in heat-maps reconstruction, we design a distribution-aware heat-maps distillation, which is illustrated in Fig. 5. The distribution-aware heat-maps distillation method first softens the distribution of H^t by a softmax function with a temperature coefficient $\tau|_{\tau>1}$. The above softening operation reduces the difference between the responses belonging to the background and the joints respectively, and each channel of $H^t \in \mathbb{R}^{K\times W\times L}$, where W and L are the width and height of H^t respectively, is normalized to a probability distribution. In order to properly transfer the dark knowledge to the inference network, the $H^s \in \mathbb{R}^{K\times W\times L}$ are also softened by the same softmax function. The softening of H^t and H^s are respectively formulated as follows:

$$P_{k,x,y}^t = \frac{\exp\left(\mathcal{T}(z)_{k,x,y}/\tau\right)}{\sum_{i=1}^{W}\sum_{j=1}^{L}\exp\left(\mathcal{T}(z)_{k,i,j}/\tau\right)}, \tag{6}$$

$$P_{k,x,y}^s = \frac{\exp\left(\mathcal{S}^h\left(\phi_u\left(\mathcal{S}^b\left(\phi_s(z)\right)\right)\right)_{k,x,y}/\tau\right)}{\sum_{i=1}^{W}\sum_{j=1}^{L}\exp\left(\mathcal{S}^h\left(\phi_u\left(\mathcal{S}^b\left(\phi_s(z)\right)\right)\right)_{k,i,j}/\tau\right)}, \tag{7}$$

where $P^t \in \mathbb{R}^{K\times W\times L}$ and $P^s \in \mathbb{R}^{K\times W\times L}$ are the softened and normalized probability distributions of H^t and H^s respectively.

Significantly, the smaller response regions do not affect the prediction accuracy since the final joint coordinates are calculated via an argmax function. Finally, KL divergence is employed to measure the distance between these two distributions, and the loss function $\mathcal{L}_{ld}$ is computed by:

$$\mathcal{L}_{ld} = \frac{1}{K}\sum_{k=1}^{K}\mathrm{KL}(P_k^t \parallel P_k^s) = \frac{1}{K}\sum_{k=1}^{K}\sum_{i=1}^{W}\sum_{j=1}^{L} P_{k,i,j}^t \log\frac{P_{k,i,j}^t}{P_{k,i,j}^s}. \tag{8}$$

3.3.3. Overall training process

The dual-branch auto-encoding strategy incorporates all five supervisions into the training process of the inference network. Besides these losses mentioned above (*i.e.* $\mathcal{L}_{st}$, $\mathcal{L}_{fm}$, $\mathcal{L}_{thm}$ and $\mathcal{L}_{ld}$), the Encoder-HDecoder network is also supervised by the alignment error $\mathcal{L}_{lhm}$ between H^s and the GT labels of joints H^l, which is formulated as:

$$\mathcal{L}_{lhm} = \frac{1}{K}\sum_{k=1}^{K} \| H_k^l - \mathcal{S}^h\left(\phi_u\left(\mathcal{S}^b\left(\phi_s(z)\right)\right)\right)_k \|_2^2. \tag{9}$$

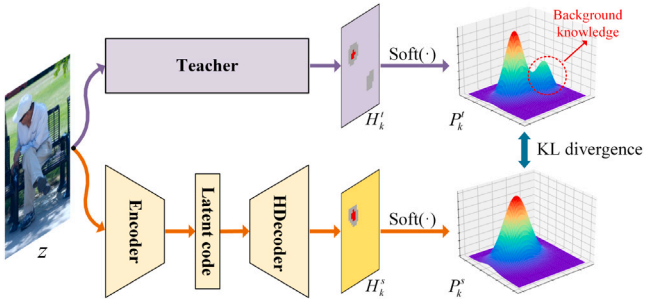


Fig. 5. Illustration of the distribution-aware heat-maps distillation. For predicted heat-maps of a joint from the teacher and the inference networks, we first transform them into two probability distributions by the Soft function (*i.e.* a softmax function with a temperature coefficient) separately. Then, we calculate the Kullback–Leibler (KL) divergence between two distributions as a supervision for training the heat-maps reconstruction. Then the amplitude difference between the joint response region and the smaller response region are reduced via the Soft function, which makes it easy to transfer the background knowledge of the smaller response region from the teacher network to the inference network.

We summarize the overall training strategy as follows:

$$\mathcal{L} = (\mathcal{L}_{st} + \alpha\mathcal{L}_{fm})\mathbb{I}(Flag = 1) + \left(\beta\mathcal{L}_{lhm} + (1-\beta)(\mathcal{L}_{thm} + \gamma\mathcal{L}_{ld})\right)\mathbb{I}(Flag = 2), \tag{10}$$

where $\mathcal{L}$ denotes the overall loss function, $\mathbb{I}(\cdot)$ is an indicator function, $Flag$ indicates the training stage. In the first stage of training, *i.e.* $Flag = 1$, we train the Encoder-FDecoder network for the regularization of the Encoder, as shown in Fig. 4(b), and a hyper-parameter α is set to balance $\mathcal{L}_{st}$ and $\mathcal{L}_{fm}$. In the second stage of training, *i.e.* $Flag = 2$, we train the Encoder-HDecoder network to generate the final predicted heat-maps H^s, as shown in Fig. 4(c). We fine-tune the regularized Encoder by dropping its learning rate by a 10-fold, and then balance $\mathcal{L}_{thm}$, $\mathcal{L}_{ld}$ and $\mathcal{L}_{lhm}$ by hyper-parameters β and γ, respectively.

4. Experiments

In this section, we report the performance of our proposed method in terms of accuracy and resource requirements. Moreover, we also provide experiments to verify the effectiveness of each component in our framework.

4.1. Datasets and evaluation metrics

MS COCO2017 [22] contains 250,000 person samples, and each person is labeled with 17 joints. We choose the train set for training and use the val and test-dev sets for testing. For evaluation, the standard metric based on object keypoint similarity (OKS) [14] is used, OKS $= \frac{\sum_i \exp(-d_i^2/2s^2k_i^2)\mathbb{I}(v_i>0)}{\sum_i \mathbb{I}(v_i>0)}$, where d_i is the Euclidean distance between the detected joint and the corresponding ground truth, v_i is the visibility flag of the ground truth, s is the object scale, k_i is a per-joint constant that controls falloff, and $\mathbb{I}(\cdot)$ is an indicator function. We report the standard average precision and recall scores: AP.50 and AP.75 (AP at OKS = 0.50 and 0.75 respectively), AP (the mean of AP scores at 10 positions, OKS = 0.50, 0.55, …, 0.90, 0.95), AP(M) for medium objects, AP(L) for large objects, and AR at OKS = 0.50, 0.55, …, 0.90, 0.95.

MPII [23] contains 40,000 person samples, and each person is labeled with 16 joints. We choose the train set for training and use the val set for testing. For evaluation, the standard metric PCKh [23] score is used. A joint is correct if it falls within ηl pixels of the ground truth position, where η is a constant and l is the head size that corresponds to 60% of the diagonal length of the ground-truth head bounding box. We report the PCKh scores of each joint, head, shoulder, elbow, wrist, hip, knee and ankle at $\eta = 0.5$, and the mean PCKh scores at $\eta = 0.5$ and $\eta = 0.1$ respectively.

Despite both floating point operations (FLOPs) and model parameters are reported in our results, we only utilize FLOPs to evaluate the computational efficiency of the model, since the processing speed is more important than the storage consumption in most embedded systems.

4.2. Implementation details

We use Adam as the optimizer in our method, and the training strategy is shown in Fig. 4. The hyper-parameter σ_h in Eq. (1) is set to 2, λ in Eq. (3) is set to 0.1, τ in Eqs. (6) and (7) is set to 4. α, β and γ in Eq. (10) are set to 1, 0.5 and 0.1 respectively. We set the scale coefficients both S_w and S_h in Eq. (2) to 0.5, so the interpolating need not be used in our downscaler and the stride of the second convolutional layer in the downscaler is set to 2. For experiments on MS COCO2017, we set the original size of input to 256 × 192, hence the size of thumbnail images is 128 × 96. For experiments on MPII, we set the size of original inputs as 256 × 256, hence the size of thumbnail image is 128 × 128. In our method, the structures of teacher and inference networks are similar, and the differences are that the downscaler $\phi_s(\cdot)$ and up-sampling module $\phi_u(\cdot)$ are inserted into the inference network for generating thumbnail images and scaling up the size of heat-maps H^s respectively. In addition, the FDecoder $\phi_d(\cdot)$ is only activated in the training stage, which is bypassed in the inference stage. We present the details of our designed modules, *i.e.* the downscaler $\phi_s(\cdot)$, the FDecoder $\phi_d(\cdot)$ and the up-sampling module $\phi_u(\cdot)$, in Table 1.

In Section 3, we employ HRNet [14] as an example to describe our method. Actually, our ThumbPose is designed to apply on arbitrary FCN-based networks, *e.g.* Hourglass [13], SimpleBaseline [27] and HRNet *etc*. To further verify the generality of ThumbPose, we conduct experiments based on Hourglass, SimpleBaseline, HRNet and DARK [15].

For experiments based on Hourglass, two 8-stage Hourglass networks are used as the backbone of teacher and inference networks respectively. In the inference network, we insert 7 FDecoders into the output positions of the first 7 Hourglass modules individually, and insert our up-sampling module between the final Hourglass module and head. The total number of training epochs is 140, where the loss functions and modules used in the stage 1 and stage 2 are activated in all epochs. The base learning rate Lr is set to 2.5e−4, and it is dropped to 2.5e−5 and 2.5e−6 at the 90th and 120th epochs respectively.

As for experiments based on SimpleBaseline, two SimpleBaseline networks are used as the backbone of teacher and inference networks respectively. The specific architectures of the networks used in our experiments include SimpleBaseline-R50 and SimpleBaseline-R101, and both of them are consisted sequentially by 4 convolutional modules and 3 deconvolutional layers. In the inference network, we insert one FDecoder into the output position of the second convolutional module, and insert our up-sampling module between the final deconvolutional layer and head. The total number of training epochs is 166, where the first 26 epochs for the stage 1 and the rest epochs for the stage 2. The base learning rate Lr is set to 1e−3, and it is dropped to 1e−4 and 1e−5 at the 116th and 146th epochs respectively.

For experiments based on HRNet and DARK (which is an HRNet-based model), two HRNet networks are used as the backbone of teacher and inference networks respectively. The specific architectures of the networks used in our experiments include HRNet-W32 and HRNet-W48, and both of them are consisted sequentially by 4 multi-resolution subnetworks. In the inference network, we insert one FDecoder into the output position of the second multi-resolution subnetwork, and insert our up-sampling module between the final multi-resolution subnetwork and head. The total number of training epochs is 236, where the first 26 epochs for the stage 1 and the rest epochs for the stage 2. The base

Table 1
The details of the designed modules in our method. Where St. denotes the structure of layers, K. indicates the kernel size, $C_{\cdot in}$ and $C_{\cdot out}$ denote the channel number of input and output of feature maps respectively, S. denotes the stride and P. indicates the number of padding. $N \in \mathbb{Z}$ is the channel number of any intermediate feature maps.

Layer	Downscaler $\phi_s(\cdot)$						FDecoder $\phi_d(\cdot)$						Up-sampling module $\phi_u(\cdot)$					
	St.	K.	$C_{\cdot in}$	$C_{\cdot out}$	S.	P.	St.	K.	$C_{\cdot in}$	$C_{\cdot out}$	S.	P.	St.	K.	$C_{\cdot in}$	$C_{\cdot out}$	S.	P.
Layer-1	Conv-ReLU	5	3	12	1	2	Deconv-ReLU	3	N	N	2	1	2 × Up-sample	–	N	N	–	–
Layer-2	Conv-ReLU	5	12	3	2	2	None	–	–	–	–	–	Conv-BN	3	N	N	1	1
Layer-3	None	–	–	–	–	–	None	–	–	–	–	–	Conv-BN-ReLU	3	N	N	1	1

Table 2
Comparison of our proposed method with the state-of-the-art human pose estimation methods using small-size image as input on MS COCO2017 val set, where 'SimBa' denotes 'SimpleBaseline'.

Method	Backbone	Input size	GFLOPs	#Params	AP	AP.50	AP.75	AP(M)	AP(L)	AR
Hourglass [13]	4-stage Hourglass	128 × 96	2.7	**13.0M**	66.2	87.6	75.1	63.8	71.4	72.8
Hourglass [13]	8-stage Hourglass	128 × 96	4.9	25.1M	67.6	88.3	77.4	65.2	73.0	74.0
SimBa [27]	ResNet-50	128 × 96	2.3	34.0M	59.3	85.5	67.4	57.8	63.8	66.6
SimBa [27]	ResNet-101	128 × 96	3.1	53.0M	58.8	85.3	66.1	57.3	63.4	66.1
SimBa [27]	ResNet-152	128 × 96	3.9	68.6M	60.7	86.0	69.6	59.0	65.4	68.0
HRNet [14]	HRNet-W32	128 × 96	**1.8**	28.5M	66.9	88.7	76.3	64.6	72.3	73.7
HRNet [14]	HRNet-W48	128 × 96	3.6	63.6M	68.0	88.9	77.4	65.7	73.7	74.7
DARK [15]	HRNet-W32	128 × 96	1.8	28.5M	70.7	88.9	78.4	67.9	76.6	76.7
DARK [15]	HRNet-W48	128 × 96	3.6	63.6M	71.9	89.1	79.6	69.2	78.0	77.9
Graph-PCNN [16]	HRNet-W32	128 × 96	>1.8	>28.5M	71.5	89.0	79.0	68.4	77.6	77.3
Graph-PCNN [16]	HRNet-W48	128 × 96	>3.6	>63.6M	72.8	89.2	80.1	69.9	79.0	78.6
TokenPose-L/D24 [29]	HRNet-W48-stage3	128 × 96	2.3	26.1M	71.1	88.8	78.5	68.4	77.0	77.0
UniFormer-base [30]	UniFormer-base	128 × 96	–	–	61.9	87.2	71.1	60.3	67.1	69.3
SimBa + ThumbPose	ResNet-50	128 × 96	2.5	34.1M	64.4	86.7	72.7	61.8	70.2	70.8
SimBa + ThumbPose	ResNet-101	128 × 96	3.4	53.1M	63.8	87.1	71.8	61.4	69.3	70.2
DARK + ThumbPose	HRNet-W32	128 × 96	1.9	28.6M	72.4	89.6	79.9	68.7	79.3	77.8
DARK + ThumbPose	HRNet-W48	128 × 96	3.8	63.6M	**74.1**	**89.8**	**81.3**	**70.4**	**81.1**	**79.3**

learning rate Lr is set to 1e−3, and it is dropped to 1e−4 and 1e−5 at the 196th and 226th epochs respectively.

Following the same way as most works of human pose estimation, the used backbones are pre-trained on the large-scale RGB image dataset ImageNet [69]. Our models follow the top-down paradigm in pose estimation, which first obtains the bounding boxes of human instances in images, and then single person pose estimation is performed on the human body in each box separately. There are different ways to obtain the bounding boxes of human instances for MS COCO2017 and MPII, since the images of MS COCO2017 usually contain multiple human instances but each image of MPII contains only one human instance. In this paper, for MS COCO2017, we employ the human bounding boxes of official annotations to crop human instances in the train set, and adopt the same human instance detector as HRNet [14] to obtain the bounding boxes of human instances for inference, which achieves 56.4% and 60.9% AP on the val and test-dev sets respectively. For MPII, we directly use the whole image as the network input of human pose estimation. In all of our experiments, we utilize PyTorch to implement our method, and run it on two Quadro RTX6000 GPUs.

4.3. Comparison to the state-of-the-arts

We compare our method with other state-of-the-art methods using small-size (*i.e.* 128 × 96 pixels obtained by affine transformation) images as inputs. For the training of ThumbPose (HRNet-W32/HRNet-W48), the teacher network is DARK (HRNet-W32/HRNet-W48) with original-size (*i.e.* 256 × 192 pixels) images as inputs. Table 2 shows the results of our ThumbPose and other SOTA methods, including Hourglass [13], SimpleBaseline [27], HRNet [14], DARK [15], Graph-PCNN [16], TokenPose [29] and UniFormer [30], on COCO val set.

From Table 2, we have the following observations: (1) DARK-W48 with ThumbPose absolutely surpasses all the previous approaches in terms of accuracy and recall, which achieves the best score of 74.1% AP and 79.3% AR with competitive efficiency (*i.e.* 3.8 GFLOPs). (2) DARK-W48 with ThumbPose exceeds the previous state-of-the-art method Graph-PCNN-W48 (*i.e.* 72.8% AP and 78.6% AR) by about 1.3% AP and 0.7% AR. In addition, DARK-W48 with ThumbPose outperforms Graph-PCNN-W48 on all the sub-items, *i.e.* AP.50, AP.75, AP(M) and AP(L), and the GFLOPs gap between DARK-W48 with ThumbPose and Graph-PCNN-W48 is lower than 0.2 because both models are based on the same backbone(*i.e.* HRNet-W48). All of these comparisons demonstrate that, in the case of using small-size representation, our method is superior than the most advanced graph-based refining strategy. (3) SimpleBaseline with ThumbPose comprehensively surpasses the performance of its baseline SimpleBaseline on all the evaluation items whether using ResNet-50 or ResNet-101 as the backbone. Specifically, SimpleBaseline-50/101 with ThumbPose improves respectively the AP and AR of SimpleBaseline-50/101 from 59.3%/58.8% and 66.6%/66.1% to 64.4%/63.8% and 70.8%/70.2% with only 0.2/0.3 GFLOPs of extra cost. (4) DARK with ThumbPose comprehensively surpasses the performance of its baseline DARK on all the evaluation items whether using HRNet-W32 or HRNet-W48 as the backbone. Specifically, DARK-W32 with ThumbPose improves the AP and AR of DARK-W32 from 70.7% and 76.7% to 72.4% and 77.8% respectively, and DARK-W48 with ThumbPose exceeds DARK-W48 (*i.e.* 71.9% AP and 77.9% AR) of about 2.2% AP and 1.4% AR respectively. Moreover, DARK-W32/W48 with ThumbPose requires tiny extra computation, which increases the GFLOPs of DARK-W32/W48, from 1.8/3.6 to 1.9/3.8, by only 0.1/0.2 respectively. It can be proven by these comparisons that our informative thumbnail representation can effectively improve model performance with negligible attached consumption. (5) We represent key attributes (*i.e.* AP, AR and GFLOPs) of several SOTA models in a bubble chart form, as shown in Fig. 6(a), where DARK-W32 with ThumbPose reaches the best performance in the application of small-scale networks, and DARK-W48 with ThumbPose achieves the best AP and AR absolutely with competitive operation requirements compared with all of other methods.

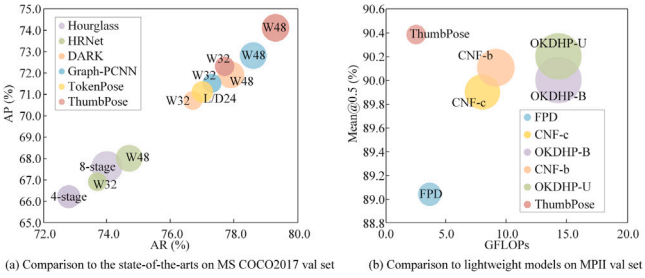
Table 3
Comparison of our proposed method with the state-of-the-art human pose estimation methods using small-size images as inputs on MS COCO2017 test-dev set.

Method	Backbone	Input size	GFLOPs	#Params	AP	AP.5	AP.75	AP(M)	AP(L)	AR
SimpleBaseline [27]	ResNet-50	128 × 96	2.3	34.0M	60.3	88.1	67.8	58.6	62.8	64.4
SimpleBaseline [27]	ResNet-101	128 × 96	3.1	53.0M	59.7	88.2	67.5	58.6	61.7	64.1
SimpleBaseline [27]	ResNet-152	128 × 96	3.9	68.6M	61.1	88.2	69.7	59.9	63.8	65.5
HRNet [14]	HRNet-W32	128 × 96	**1.8**	**28.5M**	60.0	90.2	70.1	57.4	64.9	67.2
HRNet [14]	HRNet-W48	128 × 96	3.6	63.6M	61.2	90.6	71.4	58.6	66.3	68.4
UDP [38]	HRNet-W32	128 × 96	1.8	28.6M	67.9	90.6	76.2	65.9	72.3	74.3
UDP [38]	HRNet-W48	128 × 96	3.7	63.7M	69.7	90.9	78.1	67.6	74.2	75.9
DARK [15]	HRNet-W32	128 × 96	1.8	28.5M	70.0	90.9	78.5	67.4	75.0	75.9
CAL [60]	HRNet-W32	128 × 96	2.5	49.3M	71.4	91.3	79.6	69.2	76.1	77.3
CAL [60]	HRNet-W48	128 × 96	5.3	110.3M	72.3	91.6	80.5	69.9	77.2	78.0
DARK + ThumbPose	HRNet-W32	128 × 96	1.9	28.6M	71.7	91.3	79.7	68.6	77.3	77.2
DARK + ThumbPose	HRNet-W48	128 × 96	3.8	63.6M	**73.4**	**91.9**	**81.3**	**70.1**	**79.1**	**78.6**

Table 4
Comparison with lightweight methods of human pose estimation on MPII val set.

Method	Input size	GFLOPs	#Params	Hea	Sho	Elb	Wri	Hip	Kne	Ank	Mean@0.5	Mean@0.1
8-stage Hourglass[a] [13]	256 × 256	26.2	25.6M	97.2	96.4	90.8	86.5	90.0	86.8	82.7	90.5	38.3
4-stage Hourglass [13]	256 × 256	**3.6**	3.3M	96.8	95.2	87.7	82.9	87.9	82.6	78.3	87.9	34.6
4-stage Hourglass + FPD [8]	256 × 256	**3.6**	3.3M	96.9	95.6	89.0	84.4	88.9	84.0	80.7	89.0	36.1
4-stage Hourglass + OKDHP-B [9]	256 × 256	14.3	13.0M	97.0	96.1	90.8	85.9	89.5	85.4	81.6	90.0	–
4-stage Hourglass + OKDHP-U [9]	256 × 256	14.3	13.0M	–	–	–	–	–	–	–	**90.2**	–
8-stage Hourglass [13]	128 × 128	6.5	25.6M	95.8	93.9	86.5	79.3	86.4	81.4	77.4	86.4	25.4
8-stage Hourglass + ThumbPose	128 × 128	6.9	25.7M	96.7	96.0	89.6	84.2	90.1	86.0	82.1	89.7	34.9
HRNet-W32[a] [14]	256 × 256	9.5	28.6M	97.1	95.9	−90.3	86.5	89.1	87.1	83.3	90.3	37.7
HRNet-W32 + CNF-c [51]	256 × 256	8.3	12.9M	–	–	–	–	–	–	–	89.9	39.5
HRNet-W32 + CNF-b [51]	256 × 256	9.4	16.4M	–	–	–	–	–	–	–	**90.1**	39.4
HRNet-W32 [14]	128 × 128	**2.3**	28.6M	96.4	94.7	87.4	81.3	87.2	82.6	79.5	87.6	29.6
HRNet-W32 + ThumbPose	128 × 128	2.5	28.6M	96.8	96.1	90.0	85.2	89.8	86.7	82.9	**90.1**	38.4
DARK-W32[a] [15]	256 × 256	9.5	28.6M	97.2	95.9	91.2	86.7	89.7	86.7	84.0	90.6	42.0
DARK-W32 [15]	128 × 128	**2.3**	28.6M	96.4	94.7	87.7	87.7	87.6	82.9	79.6	87.8	34.6
DARK-W32 + ThumbPose	128 × 128	2.5	28.6M	97.0	96.3	90.4	85.9	90.1	86.8	83.0	**90.4**	39.9

[a] Denotes the high-accuracy large-consumption baselines of lightweight models.



(a) Comparison to the state-of-the-arts on MS COCO2017 val set (b) Comparison to lightweight models on MPII val set

Fig. 6. Comparison of our proposed ThumbPose with other SOTA models. (a) The x-axis, y-axis, and bubble size indicate AR, AP, and GFLOPs respectively, where ThumbPose is instantiated on DARK-w32 and DARK-W48. (b) The x-axis, y-axis, and bubble size indicate GFLOPs, Mean@0.5, and GFLOPs respectively, where ThumbPose is instantiated on DARK-w32. Best viewed in color and zoomed in.

In order to further demonstrate the effectiveness of our proposed method, we compare ThumbPose with the state-of-the-art models, including SimpleBaseline [27], HRNet [14], UDP [38], DARK [15] and CAL [60], on COCO test-dev set when using small-size images as inputs, as shown in Table 3. From Table 3, we can observe that the proposed DARK-W48 with ThumbPose still comprehensively exceeds all of other methods, which achieves the best 73.4% AP and 78.6% AR with competitive efficiency (*i.e.* 3.8 GFLOPs). Compared to the previous most advanced method of low-resolution human pose estimation, *i.e.* CAL-W48, which reaches 72.3% AP and 78.0% AR, DARK-W48 with ThumbPose further improves the performance by 1.1% AP and 0.6% AR. When employing the small-scale HRNet-W32 as the backbone, the AP of DARK-W32 with ThumbPose (*i.e.* 71.7%) also exceeds CAL-W32 (*i.e.* 71.4%) by 0.3%. Significantly, DARK-W32/W48 with ThumbPose not only surpasses the previous most advanced CAL(W32/W48) on predicted performance, but also drastically reduces the network consumption, *i.e.* reduces GFLOPs from 2.5/5.3 to 1.9/3.8 and decreases model parameters from 49.3M/110.3M to 28.6M/63.6M. The large margin improvements of predicted performance and efficiency prove that when processing low resolution images, our method is more advanced than the previous superior confidence-aware learning method. All of the above comparisons of SOTA methods and our method demonstrate that the proposed ThumbPose achieves superior accuracy and competitive efficiency on different backbones compared to state-of-the-art methods when using small-size images as inputs.

4.4. Comparison to lightweight methods

To compare our method with other SOTA lightweight methods and analyze the trade-offs between accuracy and computational efficiency, we compare our proposed ThumbPose with some lightweight methods, *i.e.* FPD [8], OKDHP [9], CNF [51], and the vanilla methods that directly compress the network or reduce the input size. For fair comparison, this series of experiments is conducted on MPII dataset, and the model performances are reported in Table 4.

First, we compare ThumbPose with FPD and OKDHP based on the Hourglass network, as shown in the top part of Table 4. Here, the high-accuracy large-consumption baseline is an 8-stage Hourglass model with the input size of 256 × 256 pixels. Although directly reducing the input size of the baseline to 128 × 128 or compressing the baseline

Table 5
Ablation studies of ThumbPose on MS COCO2017 val set, where *Dow.* and *Ups.* denote our proposed downscaler and up-sampling module respectively.

Method	Input size	GFLOPs	*Dow.*	*Ups.*	$\mathcal{L}_{st}$	$\mathcal{L}_{fm}$	$\mathcal{L}_{thm}$	$\mathcal{L}_{ld}$	AP	AP.50	AP.75	AP(M)	AP(L)	AR
(A) Baseline by DARK [15]	256 × 192	7.12							75.6	90.5	82.1	71.8	82.8	80.8
(B) Bi-linear by DARK [15]	128 × 96	**1.78**							70.7	88.9	78.4	67.9	76.6	76.7
(C) Downscaler	128 × 96	1.83	✓						71.1	89.1	79.3	68.3	77.4	77.0
(D) Style supervision	128 × 96	1.83	✓		✓				71.3	88.7	79.0	68.0	77.8	77.0
(E) Up-sampling module	128 × 96	1.89	✓	✓	✓				71.1	89.1	78.8	68.0	77.5	76.8
(F) Regularization	128 × 96	1.89	✓	✓	✓	✓			71.7	89.2	79.4	68.4	78.3	77.4
(G) Heat-map distillation	128 × 96	1.89	✓	✓	✓		✓		71.5	89.1	79.1	68.2	77.9	77.2
(H) Heat-map distillation[a]	128 × 96	1.89	✓	✓	✓		✓	✓	71.6	89.2	79.2	68.4	77.9	77.2
(I) ThumbPose	128 × 96	1.89	✓	✓	✓	✓	✓	✓	**72.4**	**89.6**	**79.9**	**68.7**	**79.3**	**77.8**

[a] Denotes the background knowledge of the H^t is further distilled by the $\mathcal{L}_{ld}$ loss.

network into a 4-stage Hourglass model can reach better computational efficiency (*i.e.* 6.5 and 3.6 GFLOPs respectively), the prediction accuracy is sharply decreased by 4.1% and 2.6% Mean@0.5 (from 90.5% to 86.4% and 87.9%). Previous SOTA methods FPD/OKDHP-U follow the lightweight method of compressing network, and improve the mean accuracy of 4-stage Hourglass model (256 × 256) from 87.9% to 89.0%/90.2% Mean@0.5 with a consumption of 3.6/14.3 GFLOPs. Our ThumbPose aims to enhance the performance of small-size-input model, and achieves 89.7% Mean@0.5 with 6.9 GFLOPs. Compared to FPD and OKDHP, our ThumbPose achieves a moderate balance between accuracy and computational efficiency. Significantly, the performance gain caused by our method (+3.3% Mean@0.5, *i.e.* from 86.4% to 89.7%) is higher than FPD and OKDHP-U (+1.1% and +2.3% Mean@0.5, *i.e.* from 87.9% to 89.0% and 90.2%, respectively). This comparison demonstrates that our method is competitive with FPD and OKDHP on Hourglass network.

Then, we compare the proposed method with CNF based methods on the vanilla HRNet network, as shown in the middle part of Table 4. Here, the high-accuracy large-consumption baseline is a HRNet-W32 model with the input size of 256 × 256 pixels. The existing SOTA method CNF-c/b exploits the method of neural architecture search to automatically search a lightweight substitute of HRNet-W32, which accelerates the baseline from 9.5 to 8.3/9.4 GFLOPs but decreases the Mean@0.5 from 90.3% to 89.9%/90.1%. When reducing the input size of the baseline to 128 × 128, the GFLOPs and Mean@0.5 are decreased to 2.3 and 87.6%, separately. Using our ThumbPose can improve the mean accuracy of 128 × 128-input-sized HRNet-W32 to 90.1% Mean@0.5 with +0.2 GFLOPs of extra cost (from 2.3 to 2.5). Compared to CNF-c/b, our model is superior since ThumbPose not only achieves faster operational speed (2.5 *vs.* 8.3/9.4), but also reaches a similar prediction accuracy (90.1% *vs.* 90.1%/89.9% Mean@0.5, 38.4% *vs.* 39.5%/39.4% Mean@0.1). Furthermore, the lightweight method of ThumbPose (*i.e.* reducing input size) is easier than CNF. This comparison demonstrates that our method of compressing the feature maps is more effective than the method of neural architecture search on the HRNet network.

Finally, we execute ThumbPose on the SOTA model (*i.e.* DARK [15]) of human pose estimation, to further validate the acceleration ability of our method, as shown in the bottom part of Table 4. Here, the high-accuracy large-consumption baseline is a DARK-W32 model with the input size of 256 × 256 pixels. When merely reducing the input size to 128 × 128, the GFLOPs and Mean@0.5/0.1 of the baseline are decreased from 9.5 and 90.6%/42.0% to 2.3 and 87.8%/34.6%, separately. However, using ThumbPose can improve the performance of small-input-sized model to 90.4%/39.9% Mean@0.5/0.1 with only 0.2 GFLOPs (from 2.3 to 2.5) of extra cost. The ThumbPose-based lightweight model of DARK-W32 achieves the highest mean accuracy and the fastest processing speed compared to FPD, OKDHP and CNF, and a more intuitive representation is shown in Fig. 6(b).

To obtain small-scale models, OKDHP and FPD rely on specific structure of networks (*e.g.* Hourglass), and CNF requires the complex operation of neural architecture search. Excitingly, our proposed ThumbPose not only can be easily applied on arbitrary FCN-based models (*e.g.* Hourglass, HRNet and DARK in this subsection), but also achieves competitive performance both on prediction accuracy and computational efficiency compared to previous SOTA lightweight methods. It is worth noted that processing speed is more important than storage consumption in most embedded systems, so our method is more suitable for real-time applications even though the network parameters (#Params) are not compressed.

4.5. Ablation studies

4.5.1. Effect of major components in ThumbPose

We experimentally validate and analyze the effectiveness of each component in our proposed ThumbPose, *i.e.* a downscaler, an up-sampling module, $\mathcal{L}_{st}$, $\mathcal{L}_{fm}$ and $\mathcal{L}_{thm}$, *etc*. Our experiments are conducted on COCO val set, as shown in Table 5, and we analyze them in the following.

(A) Baseline. Our baseline is DARK(HRNet-W32) [15] using original-size images (*i.e.* 256 × 192) as inputs. We inherit the networks and all hyper-parameters as in DARK unless otherwise stated, and the performance is copied from the published paper, as shown in Table 5(A).

(B) Bi-linear. A simple method [15] to generate small-size images is bi-linear based affine transformation, and the generated small-size image is used as the network input to perform pose estimation. We copy the reported performance from the published paper and show it in Table 5(B), Bi-linear setting reduces the GFLOPs from 7.12 to 1.78 when compared with baseline (A), but the accuracy and recall are sharply declined by 4.9% (from 75.6% to 70.7%) and 4.1% (from 80.8% to 76.7%) respectively, which demonstrates that small-size input can accelerate network operation, but the model performance will be decreased dramatically.

(C) Downscaler. In order to maintain the performance when using small-size images as inputs, we first use our proposed downscaler to reduce the size of original images (*i.e.* 256 × 192 pixels) into thumbnail images (*i.e.* 128 × 96 pixels). Compared to the bi-linear based affine transformation (B) method, using thumbnail images as inputs, as shown in Table 5(C), improves the accuracy and recall by 0.4% (from 70.7% to 71.1%) and 0.3% (from 76.7% to 77.0%) respectively with negligible growth of execution cost (0.05 GFLOPs). These results prove that using the thumbnail image generated by our downscaler, instead of the small-size image obtained by the affine transformation, as model inputs can effectively improve the model accuracy.

(D) Style supervision. We improve the visual authenticity of the generated thumbnail images by supervising the training of downscaler with $\mathcal{L}_{st}$, as shown in Table 5(D). Despite the performance gets worse in AP.50, AP.75 and AP(M), the effectiveness of the proposed style supervision is demonstrated by the improvement on the mean prediction accuracy (*i.e.* from 71.1% to 71.3% AP). Moreover, the visual analysis for the style supervised thumbnail will be conducted in Section 4.6.

(E) Up-sampling module. To enable the thumbnail-input network can be guided by the teacher network, we align the size of the H^s and

Table 6
Prediction accuracy with different methods of image down-scaling on MS COCO2017 val set. Here HE denotes the processing method of histogram equalization.

Image down-scaling	Input size	AP	AP.50	AP.75	AP(M)	AP(L)	AR
(a) None	256 × 192	75.6	90.5	82.1	71.8	82.8	80.8
(b) Bi-linear	128 × 96	70.7	88.9	78.4	67.9	76.6	76.7
(c) Bi-linear + HE	128 × 96	70.0	88.8	78.0	67.1	76.0	76.0
(d) Our downscaler w/o $\mathcal{L}_{st}$	128 × 96	71.1	**89.1**	**79.3**	**68.3**	77.4	**77.0**
(e) Our downscaler w/ $\mathcal{L}_{st}$	128 × 96	**71.3**	88.7	79.0	68.0	**77.8**	**77.0**

Table 7
The performance of the temperature coefficient τ in the $\mathcal{L}_{ld}$ for background knowledge transfer on MS COCO2017 val set.

τ	0.1	1.0	2.0	4.0	10.0	w/o $\mathcal{L}_{ld}$
AP	72.1	72.2	72.3	**72.4**	72.0	72.3
AR	77.7	77.7	77.7	**77.8**	77.6	77.7

H^t by inserting an up-sampling module. As shown in Table 5(E), the up-sampling module decreases both the accuracy and recall by 0.2% (71.3% *vs.* 71.1%, 77.0% *vs.* 76.8% respectively), while with additional consumption of 0.06 GFLOPs (*i.e.* from 1.83 to 1.89). Thus, we conclude that the under-refined information contained in the thumbnail representation is not enough to reconstruct H^t.

(F) Regularization. We further enable F^s to contain the same information as their original-input counterparts F^t by using $\mathcal{L}_{fm}$ to regularize the Encoder network. As shown in Table 5(F), the 0.4% (from 71.3% to 71.7%, and from 77.0% to 77.4% respectively) gain of both AP and AR demonstrates the effectiveness of the regularization when compared to the style supervised downscaler (D). This experiment demonstrates that our regularization strategy can greatly improve the performance of thumbnail-input networks.

(G) Heat-map distillation. We further refine the valuable information contained in the thumbnail representation by using $\mathcal{L}_{thm}$ to distill the knowledge of the heat-map, when comparing the improvement brought by the style supervised downscaler (D), both AP and AR are increased by 0.2% (71.3% *vs.* 71.5%, and 77.0% *vs.* 77.2%). This experimental result proves that MSE based Heat-map distillation is conductive to refining the vital information in thumbnail feature maps.

(H) Heat-map distillation‡. Moreover, we enhance the ability of heat-map distillation by exploiting the distribution-aware loss $\mathcal{L}_{ld}$ to transfer the background knowledge. As shown in Table 5(H), the AP is further improved from 71.5% (G) to 71.6%. It can be demonstrated that transferring of the background knowledge can further enhance the model ability for processing thumbnail representation.

Finally, all of our proposed components are integrated into ThumbPose (I), compared to the original-input baseline (A), ThumbPose sharply reduces the computation cost from 7.12 GFLOPs to 1.89 GFLOPs as shown in Table 5(I). Compared to DARK using small-size images as inputs (B), we dramatically improve the AP and AR by 1.7% (from 70.7% to 72.4%) and 1.1% (from 76.7% to 77.8%) respectively, and the execution cost is only increased by 0.11 GFLOPs. The above comparison demonstrates that all components in the framework of our proposed ThumbPose are effective for pose estimation on low-resolution images.

4.5.2. Effect of different methods for image down-scaling

To analyze the effect of different methods for image down-scaling, we conduct some experiments on MS COCO2017 dataset. The used baseline is an DARK-W32 [15] model, which directly utilizes the original-size image (*i.e.* 256 × 192 pixels) as the input, as shown in Table 6(a). The adopted down-scaling methods include (b) bi-linear based down-scaling, (c) using the method of histogram equalization to process the small-size image generated by bi-linear based down-scaling, (d) reducing the image size via the proposed downscaler without style supervision and (e) exploiting the proposed downscaler to generate

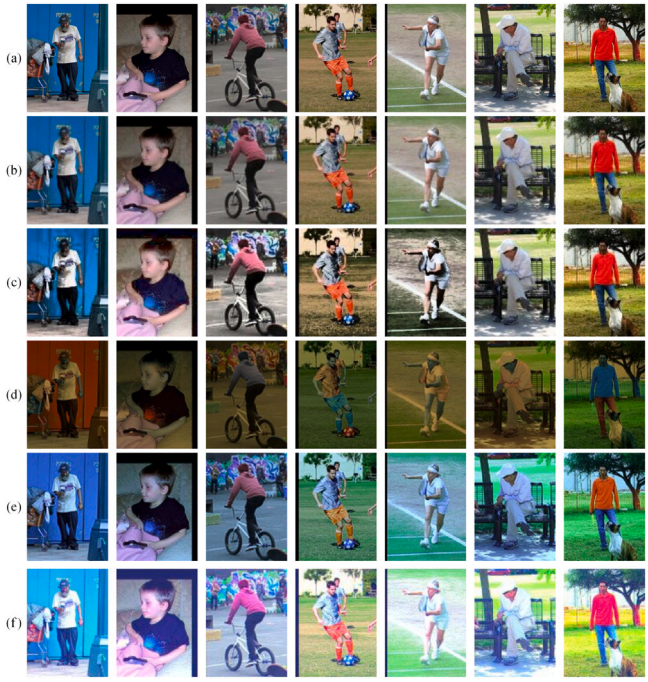


Fig. 7. Visualization comparison of the small-size images obtained by different methods, *i.e.* (a) original large-size images, (b) small-size images generated by bi-linear based down-scaling, (c) using histogram equalization to process the small-size images (b), (d) thumbnail images generated by our downscaler trained with only GT supervision $\mathcal{L}_{lhm}$, (e) thumbnail images generated by our downscaler trained with both GT ($\mathcal{L}_{lhm}$) and style ($\mathcal{L}_{st}$) supervisions, (f) thumbnail images generated by our downscaler trained under the whole procedure of ThumbPose. The original images (a) are 256 × 192 pixels, and the others are 128 × 96 pixels. The serial numbers (a) to (e) between this figure and Table 6 are one-to-one correspondence. All images are extracted from MS COCO2017 dataset. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

thumbnail with style supervision. From Table 6, we can observe that, when reducing the input size by the method of (b), the AP and AR of original-input baseline (a) are reduced separately from 75.6% and 80.8% to 70.7% and 76.7% due to the loss of vital information. Despite the image contrast can be improved by executing histogram equalization (c) on the small-size image obtained by (b), the performance gets worse, *i.e.* decreasing from 70.7%/76.7% to 70.0%/76.0% AP/AR. We think it is because that these kind of features enhanced by histogram equalization cannot contribute to the improvement of DCNN-based model, and even the valuable information in images is weakened by the operation of equalization. However, our proposed downscaler, whether supervised by style function $\mathcal{L}_{st}$ or not, achieves higher performance than the conventional methods both (b) and (c). Although the gain from the proposed style supervision is marginal, *i.e.* from (d) 71.1%/77.0% to (e) 71.3%/77.0% AP/AR, the restriction (introduced by $\mathcal{L}_{st}$) for color features can make a dramatic difference to the visual effect of thumbnail images. More visual results for small-size images obtained by above methods are shown in Fig. 7, and we will discuss them in Section 4.6.

Fig. 8. Qualitative evaluation results for joints detection on MS COCO2017 val set. The red and green skeletons denote the estimated results of the small-input DARK and our ThumbPose, respectively. Obviously, our proposed method can suppress some wrong detection joints. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.5.3. Analysis of background knowledge transfer

In our distribution-aware heat-map distillation $\mathcal{L}_{ld}$, the boundary of transferred response regions is related to the temperature coefficient τ. To prove the effectiveness of background knowledge transfer and select a suitable value of τ, we test the model performance under different τ. Our experiments are conducted on COCO val set with an HRNet-W32 backbone, as shown in Table 7. Note that all of the hyper-parameters except for τ are fixed when conducting this ablation experiment. From Table 7, we can see that both too small and too large temperatures, *i.e.* 0.1 and 10.0, lead to poor model performance, and the best AP and AR (*i.e.* 72.4% in AP and 77.8% in AR) are reached when set τ as 4.0. These experiments demonstrate that the learning of background knowledge is helpful to improve the performance of the inference network, and we set τ to 4.0 in all our experiments.

4.6. Qualitative results

For better understanding the achieved performance, we visualize some results predicted by our proposed ThumbPose-DARK-W32 and the vanilla DARK-W32 using small-size images as inputs on COCO dataset.

Fig. 7 shows the small-size images generated by different methods of down-scaling. First, compared to small-size images (b) and (c), which are generated by bi-linear based down-scaling methods, our thumbnails, *i.e.* (d), (e) and (f), have more noticeable edges. This is because the composite LPF in our downscaler effectively filters out aliasing components. Second, the style supervised thumbnails, *i.e.* (e) and (f), are more realistic than their no-style-supervision counterparts (d). This phenomenon proves that the proposed style-based supervision $\mathcal{L}_{st}$ helps our downscaler to generate RGB-like thumbnail images. Third, although executing conventional processing method (*i.e.* histogram equalization) on resized images can increase the image contrast, as shown in Fig. 7(c), the enhanced features are not benefit to improve prediction accuracy as proved in Section 4.5.2. Last, the generated thumbnail images both (d) and (e) are darker than the original images (a), but the thumbnail images (f) are brighter. The reason is that our proposed downscalers are trained in a supervised way with the consideration of the human pose estimation losses. We believe that these types of styles are helpful for network inference.

Fig. 8 shows some estimated skeletons of vanilla DARK-W32 (red) and our ThumbPose-DARK-W32 (green) when using small-size images as inputs, from these results we can observe that joint coordinates of ThumbPose are more accurate than DARK. The above qualitative results further demonstrate the effectiveness of our proposed method for pose estimation on low-resolution images. However, as shown in the last row of Fig. 8, there are some missed and false joint detections in crowded scenes. We think the reason is that compressing the crowded multi-person images significantly weakens the recognizability of some human bodies and limbs, which only occupy a small area in the original images. In the future, to improve the performance of ThumbPose in crowded scenes, we plan to design more effective methods under the transformer architecture to generate high quality thumbnail images and learn more discriminative features.

5. Conclusions and future work

In this paper, we propose a novel thumbnail representation learning method, namely ThumbPose, to accelerate the model of human pose estimation in an effective, simple and generalizable manner, which based on the fact that the computation and memory costs of an FCN-based network can be decreased by reducing the size of input images. In particular, we present a style supervised online downscaler to downscale an input image into a thumbnail image, and then design a dual-branch auto-encoding training strategy to retain the performance of thumbnail images as the original-input model in a knowledge distillation manner. Extensive experiments on MS COCO and MPII benchmarks demonstrate the effectiveness of our proposed method, and our proposed ThumbPose outperforms both previous state-of-the-art methods when using small-size images as input and lightweight models by a large margin. To the best of our knowledge, we are the first to thoroughly explore and analyze thumbnail representation for the task of human pose estimation, and this study can provide good insights for further application of depth models in real-time systems.

Despite moderate improvement has been achieved by our proposed method, the obtained thumbnail images are not sufficiently realistic. Moreover, thumbnail-based prediction accuracy is still below the original-size-input model. Hence further exploration following our work is valuable. On the other hand, detecting human key-points in crowded scenes, especially when joints overlap is aggravated due to image downscaling, is still an open challenge. In the future, for excellent human pose estimation, we plan to design more effective methods to generate the stronger thumbnail images and learn more discriminative features.

Declaration of competing interest

We would like to note that in the manuscript no conflict of interest exits in the submission of this manuscript, and manuscript is approved by all authors for publication.

Data availability

No data was used for the research described in the article.

Acknowledgments

This work is supported by the China Postdoctoral Science Foundation (Grant No. 259822), the National Postdoctoral Program for Innovative Talents (Grant No. BX20200108), the National Science Foundation of China (Grant No. 62206077), and the Science Foundation of Heilongjiang Province (LH2021F024).

References

[1] H. Duan, Y. Zhao, K. Chen, D. Shao, D. Lin, B. Dai, Revisiting skeleton-based action recognition, 2021, arXiv preprint arXiv:2104.13586.

[2] L. Shi, Y. Zhang, J. Cheng, H. Lu, Adasgn: Adapting joint number and model size for efficient skeleton-based action recognition, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 13413–13422.

[3] W. Peng, X. Hong, G. Zhao, Tripool: Graph triplet pooling for 3D skeleton-based action recognition, Pattern Recognit. 115 (2021) 107921.

[4] M. Dong, C. Xu, Skeleton-based human motion prediction with privileged supervision, IEEE Trans. Neural Netw. Learn. Syst. (2022).

[5] C. Zhong, L. Hu, Z. Zhang, Y. Ye, S. Xia, Spatial-temporal gating-adjacency GCN for human motion prediction, 2022, arXiv preprint arXiv:2203.01474.

[6] J. Yang, Y. Ma, X. Zuo, S. Wang, M. Gong, L. Cheng, 3D pose estimation and future motion prediction from 2D images, Pattern Recognit. 124 (2022) 108439.

[7] R.J. Cotton, E. McClerklin, A. Cimorelli, A. Patel, T. Karakostas, Transforming gait: Video-based spatiotemporal gait analysis, 2022, arXiv preprint arXiv:2203.09371.

[8] F. Zhang, X. Zhu, M. Ye, Fast human pose estimation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 3517–3526.

[9] Z. Li, J. Ye, M. Song, Y. Huang, Z. Pan, Online knowledge distillation for efficient pose estimation, in: IEEE/CVF ICCV, 2021, pp. 11740–11750.

[10] G. Hinton, O. Vinyals, J. Dean, et al., Distilling the knowledge in a neural network, arXiv preprint arXiv:1503.02531, 2 (7).

[11] C. Yu, B. Xiao, C. Gao, L. Yuan, L. Zhang, N. Sang, J. Wang, Lite-hrnet: A lightweight high-resolution network, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 10440–10450.

[12] Z. Zhang, Y. Jiang, J. Jiang, X. Wang, P. Luo, J. Gu, Star: A structure-aware lightweight transformer for real-time image enhancement, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 4106–4115.

[13] A. Newell, K. Yang, J. Deng, Stacked hourglass networks for human pose estimation, in: European Conference on Computer Vision, Springer, 2016, pp. 483–499.

[14] K. Sun, B. Xiao, D. Liu, J. Wang, Deep high-resolution representation learning for human pose estimation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 5693–5703.

[15] F. Zhang, X. Zhu, H. Dai, M. Ye, C. Zhu, Distribution-aware coordinate representation for human pose estimation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 7093–7102.

[16] J. Wang, X. Long, Y. Gao, E. Ding, S. Wen, Graph-pcnn: Two stage human pose estimation with graph pose refinement, in: European Conference on Computer Vision, Springer, 2020, pp. 492–508.

[17] C. Zhao, B. Ghanem, ThumbNet: one thumbnail image contains all you need for recognition, in: Proceedings of the 28th ACM International Conference on Multimedia, 2020, pp. 1506–1514.

[18] Y. Zhang, Y. Zhang, R. Tian, Z. Zhang, Y. Bai, W. Zuo, M. Ding, ThumbDet: One thumbnail image is enough for object detection, Pattern Recognit. 138 (2023) 109424.

[19] Y. Zhang, Y. Bai, M. Ding, Y. Li, B. Ghanem, Weakly-supervised object detection via mining pseudo ground truth bounding-boxes, Pattern Recognit. 84 (2018) 68–81.

[20] Y. Zhang, M. Ding, Y. Bai, B. Ghanem, Detecting small faces in the wild based on generative adversarial network and contextual information, Pattern Recognit. 94 (2019) 74–86.

[21] J. Bragantini, A.X. Falcão, L. Najman, Rethinking interactive image segmentation: feature space annotation, Pattern Recognit. (2022) 108882.

[22] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft coco: Common objects in context, in: European Conference on Computer Vision, Springer, 2014, pp. 740–755.

[23] M. Andriluka, L. Pishchulin, P. Gehler, B. Schiele, 2D human pose estimation: New benchmark and state of the art analysis, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 3686–3693.

[24] A. Toshev, C. Szegedy, Deeppose: Human pose estimation via deep neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1653–1660.

[25] S.-E. Wei, V. Ramakrishna, T. Kanade, Y. Sheikh, Convolutional pose machines, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4724–4732.

[26] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, J. Sun, Cascaded pyramid network for multi-person pose estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7103–7112.

[27] B. Xiao, H. Wu, Y. Wei, Simple baselines for human pose estimation and tracking, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 466–481.

[28] W. Li, Z. Wang, B. Yin, Q. Peng, Y. Du, T. Xiao, G. Yu, H. Lu, Y. Wei, J. Sun, Rethinking on multi-stage networks for human pose estimation, 2019, arXiv preprint arXiv:1901.00148.

[29] Y. Li, S. Zhang, Z. Wang, S. Yang, W. Yang, S.-T. Xia, E. Zhou, Tokenpose: Learning keypoint tokens for human pose estimation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 11313–11322.

[30] K. Li, Y. Wang, J. Zhang, P. Gao, G. Song, Y. Liu, H. Li, Y. Qiao, UniFormer: Unifying convolution and self-attention for visual recognition, 2022, arXiv preprint arXiv:2201.09450.

[31] Y. Li, K. Li, X. Wang, R.Y. Da Xu, Exploring temporal consistency for human pose estimation in videos, Pattern Recognit. 103 (2020) 107258.

[32] L. Tian, P. Wang, G. Liang, C. Shen, An adversarial human pose estimation network injected with graph structure, Pattern Recognit. 115 (2021) 107863.

[33] Y. Bin, Z.-M. Chen, X.-S. Wei, X. Chen, C. Gao, N. Sang, Structure-aware human pose estimation with graph convolutional networks, Pattern Recognit. 106 (2020) 107410.

[34] J. Li, S. Bian, A. Zeng, C. Wang, B. Pang, W. Liu, C. Lu, Human pose regression with residual log-likelihood estimation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 11025–11034.

[35] X. Zhou, D. Wang, P. Krähenbühl, Objects as points, 2019, arXiv preprint arXiv:1904.07850.

[36] F. Wei, X. Sun, H. Li, J. Wang, S. Lin, Point-set anchors for object detection, instance segmentation and pose estimation, in: ECCV, Springer, 2020, pp. 527–544.

[37] X. Nie, J. Feng, J. Zhang, S. Yan, Single-stage multi-person pose machines, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 6951–6960.

[38] J. Huang, Z. Zhu, F. Guo, G. Huang, The devil is in the details: Delving into unbiased data processing for human pose estimation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 5700–5709.

[39] S. Han, H. Mao, W.J. Dally, Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding, 2015, arXiv preprint arXiv:1510.00149.

[40] Y. LeCun, J. Denker, S. Solla, Optimal brain damage, Adv. Neural Inf. Process. Syst. 2 (1989).

[41] S. Han, J. Pool, J. Tran, W. Dally, Learning both weights and connections for efficient neural network, Adv. Neural Inf. Process. Syst. 28 (2015).

[42] H. Li, A. Kadav, I. Durdanovic, H. Samet, H.P. Graf, Pruning filters for efficient convnets, 2016, arXiv preprint arXiv:1608.08710.

[43] J.-H. Luo, J. Wu, W. Lin, Thinet: A filter level pruning method for deep neural network compression, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 5058–5066.

[44] J. Wu, C. Leng, Y. Wang, Q. Hu, J. Cheng, Quantized convolutional neural networks for mobile devices, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4820–4828.

[45] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, Y. Bengio, Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1, 2016, arXiv preprint arXiv:1602.02830.

[46] M. Rastegari, V. Ordonez, J. Redmon, A. Farhadi, Xnor-net: Imagenet classification using binary convolutional neural networks, in: European Conference on Computer Vision, Springer, 2016, pp. 525–542.

[47] M. Jaderberg, A. Vedaldi, A. Zisserman, Speeding up convolutional neural networks with low rank expansions, 2014, arXiv preprint arXiv:1405.3866.

[48] E.L. Denton, W. Zaremba, J. Bruna, Y. LeCun, R. Fergus, Exploiting linear structure within convolutional networks for efficient evaluation, Adv. Neural Inf. Process. Syst. 27 (2014).

[49] X. Zhang, J. Zou, X. Ming, K. He, J. Sun, Efficient and accurate approximations of nonlinear convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1984–1992.

[50] Q. Guo, X.-J. Wu, J. Kittler, Z. Feng, Differentiable neural architecture learning for efficient neural networks, Pattern Recognit. 126 (2022) 108448.

[51] S. Yang, W. Yang, Z. Cui, Searching part-specific neural fabrics for human pose estimation, Pattern Recognit. 128 (2022) 108652.

[52] B. Zhao, Q. Cui, R. Song, Y. Qiu, J. Liang, Decoupled knowledge distillation, 2022, arXiv preprint arXiv:2203.08679.

[53] Z. Yang, Z. Li, X. Jiang, Y. Gong, Z. Yuan, D. Zhao, C. Yuan, Focal and global knowledge distillation for detectors, 2021, arXiv preprint arXiv:2111.11837.

[54] T. Wang, L. Yuan, X. Zhang, J. Feng, Distilling object detectors with fine-grained feature imitation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4933–4942.

[55] W. Park, D. Kim, Y. Lu, M. Cho, Relational knowledge distillation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 3967–3976.

[56] Y. Wang, F. Sun, D. Li, A. Yao, Resolution switchable networks for runtime efficient image recognition, in: European Conference on Computer Vision, Springer, 2020, pp. 533–549.

[57] T. Yang, S. Zhu, C. Chen, S. Yan, M. Zhang, A. Willis, Mutualnet: Adaptive convnet via mutual learning from network width and resolution, in: European Conference on Computer Vision, Springer, 2020, pp. 299–315.

[58] D. Li, A. Yao, Q. Chen, Learning to learn parameterized classification networks for scalable input images, in: European Conference on Computer Vision, Springer, 2020, pp. 19–35.

[59] L. Qi, J. Kuen, J. Gu, Z. Lin, Y. Wang, Y. Chen, Y. Li, J. Jia, Multi-scale aligned distillation for low-resolution detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 14443–14453.
[60] C. Wang, F. Zhang, X. Zhu, S.S. Ge, Low-resolution human pose estimation, Pattern Recognit. 126 (2022) 108579.
[61] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, R. Girshick, Masked autoencoders are scalable vision learners, 2021, arXiv preprint arXiv:2111.06377.
[62] K. Purohit, M. Suin, A. Rajagopalan, V.N. Boddeti, Spatially-adaptive image restoration using distortion-guided networks, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 2309–2319.
[63] G. Bhat, M. Danelljan, F. Yu, L. Van Gool, R. Timofte, Deep reparametrization of multi-frame super-resolution and denoising, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 2460–2470.
[64] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, science 313 (5786) (2006) 504–507.
[65] J. Masci, U. Meier, D. Cireşan, J. Schmidhuber, Stacked convolutional auto-encoders for hierarchical feature extraction, in: International Conference on Artificial Neural Networks, Springer, 2011, pp. 52–59.
[66] V. Turchenko, E. Chalmers, A. Luczak, A deep convolutional auto-encoder with pooling-unpooling layers in caffe, 2017, arXiv preprint arXiv:1701.04949.
[67] M. Sonka, V. Hlavac, R. Boyle, Image Processing, Analysis, and Machine Vision, Cengage Learning, 2014.
[68] T.M. Lehmann, C. Gonner, K. Spitzer, Survey: Interpolation methods in medical image processing, IEEE Trans. Med. Imaging 18 (11) (1999) 1049–1075.
[69] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: IEEE CVPR, Ieee, 2009, pp. 248–255.
[70] X. Huang, S. Belongie, Arbitrary style transfer in real-time with adaptive instance normalization, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1501–1510.
[71] H. Chen, L. Zhao, H. Zhang, Z. Wang, Z. Zuo, A. Li, W. Xing, D. Lu, Diverse image style transfer via invertible cross-space mapping, in: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), IEEE Computer Society, 2021, pp. 14860–14869.
[72] H. Nam, H. Lee, J. Park, W. Yoon, D. Yoo, Reducing domain gap by reducing style bias, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 8690–8699.
[73] Z. Zheng, R. Ye, P. Wang, D. Ren, W. Zuo, Q. Hou, M.-M. Cheng, Localization distillation for dense object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 9407–9416.

Zian Zhang received the B.S. and M.S. degrees in instrument science and technology from Harbin Institute of Technology (HIT), Harbin, China, in 2018 and 2020, respectively. Currently, he is a doctoral student in the School of Instrumentation Science and Engineering at HIT. His research areas are computer vision, pattern recognition, machine learning, and deep learning. His research interests mainly include human pose estimation, object detection, action recognition, gait recognition, image and video understanding in the monitoring system.

Yongqiang Zhang received the M.S. and Ph.D. degrees in instrument science and technology from Harbin Institute of Technology (HIT), Harbin, China, in 2015 and 2020, respectively. He worked at King Abdullah University of Science and Technology (KAUST) as a visiting student from 2017 to 2018. Currently, he is an assistant professor in the School of Instrumentation Science and Engineering at Harbin Institute of Technology. His research areas are computer vision, pattern recognition, machine learning, and deep learning. His research interests mainly include face detection, weakly/fully supervised object detection, activity detection, image and video understanding in the real-world.

Yin Zhang received the MS degree in instrument science and technology from Harbin Institute of Technology (HIT), Harbin, China, in 2021. He is currently a Ph.D. student from the Harbin Institute of Technology (HIT). His research areas are computer vision, pattern recognition, machine learning, and deep learning. His research interests mainly include object detection, knowledge distillation, image super-resolution.

Rui Tian received the bachelor's degree from Harbin University of Science and Technology in 2021. He is studying for a master's degree from Harbin Institute of Technology. His research areas are computer vision, pattern recognition, deep learning and weakly/fully supervised object detection.

Mingli Ding received the B.S., M.S. and Ph.D. degrees in instrument science and technology from Harbin Institute of Technology (HIT), Harbin, China, in 1996, 1997 and 2001, respectively. He worked as a visiting scholar in France from 2009 to 2010. Currently, he is a professor in the School of Instrumentation Science and Engineering at Harbin Institute of Technology. Prof. Ding's research interests are intelligence tests and information processing, automation test technology, computer vision, and machine learning. He has published over 40 papers in peer-reviewed journals and conferences.