Estimating Human Pose Efficiently by Parallel Pyramid Networks

Lin Zhao, Nannan Wang, *Member, IEEE,* Chen Gong, *Member, IEEE,* Jian Yang, *Member, IEEE,* and Xinbo Gao, *Senior Member, IEEE*

***Abstract*—Good performance and high efficiency are both critical for estimating human pose in practice. Recent state-of-the-art methods have greatly boosted the pose detection accuracy through deep convolutional neural networks, however, the strong performance is typically achieved without high efficiency. In this paper, we design a novel network architecture for human pose estimation, which aims to strike a fine balance between speed and accuracy. Two essential tasks for successful pose estimation, preserving spatial location and extracting semantic information, are handled separately in the proposed architecture. Semantic knowledge of joint type is obtained through deep and wide sub-networks with low-resolution input, and high-resolution features indicating joint location are processed by shallow and narrow sub-networks. Because accurate semantic analysis mainly asks for adequate depth and width of the network and precise spatial information mostly requests preserving high-resolution features, good results can be produced by fusing the outputs of the sub-networks. Moreover, the computational cost can be considerably reduced comparing with existing networks, since the main part of the proposed network only deals with low-resolution features. We refer to the architecture as "parallel pyramid" network (PPNet), as features of different resolutions are processed at different levels of the hierarchical model. The superiority of our network is empirically demonstrated on two benchmark datasets: the MPII Human Pose dataset and the COCO keypoint detection dataset. PPNet outcompetes all recent methods by using less computation and memory to achieve better human pose estimation results.**

***Index Terms*—PPNet, hierarchical representation, high efficiency, human pose estimation.**

I. Introduction

GETTING the pixel location of important joints of human body plays a key role to understand people in images and videos. Accurate 2D human poses offer great convenience for high-level tasks such as 3D pose estimation [1], [2], motion prediction [3], [4], and action recognition [5], [6]. In addition, human pose estimation can be a fundamental tool utilized in multiple applications like human-computer interaction. Thus strong performance is the basic requisite for a good pose estimation model. What's more as pose estimation systems may be deployed on various platforms with limited computing capability and memory capacity, high efficiency is also very important for an usable pose estimation algorithm.

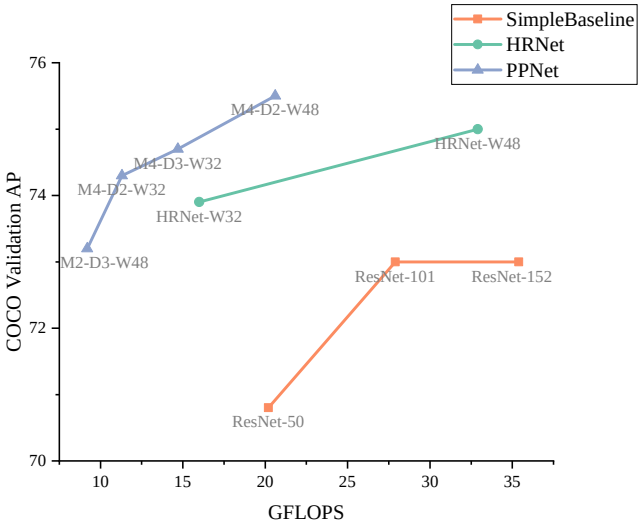


Fig. 1. A comparison of computational efficiency between PPNet, HRNet, and SimpleBaseline with the input size 384×288. Whether for small networks or big networks, our PPNet can always deliver better performance with fewer GFLOPs.

Great progress has been made by recent developed methods, particularly those adopting deep convlutional neural networks [7], [8]. Several state-of-the-art models are able to produce accurate results on both the MPII [9] and the COCO [10] benchmark datasets. Better performance is still an overwhelming desire to design more sophisticated pose estimation systems, however, as exhibited on the leaderboard of the MPII[1] and the COCO[2] benchmarks, the leading models usually make limited improvement at the price of much heavier computation. Considering the quality of poses delivered by recent methods already meets the requirements in most application scenarios, it may be more desirable to develop a pose estimation model, that gives the performance on a par with the state-of-the-art methods but uses less computation and memory, as shown in Fig. 1.

A direct idea to achieve high efficiency is to use light-

L. Zhao is with PCA Lab, Key Lab of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, and Jiangsu Key Lab of Image and Video Understanding for Social Security, School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, P.R. China, also with the State Key Laboratory of Integrated Services Networks, Xidian University. (e-mail: linzhao@njust.edu.cn)

N. Wang is with the State Key Laboratory of Integrated Services Networks, School of Telecommunications Engineering, Xidian University, Xian 710071, China. (e-mail: nnwang@xidian.edu.cn)

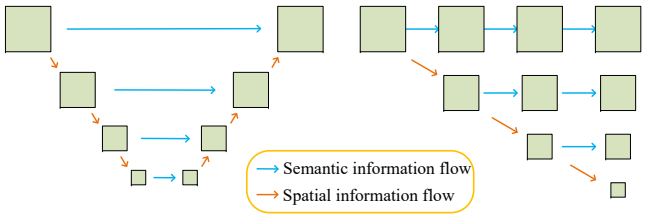
C. Gong and J. Yang are with PCA Lab, School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, P.R. China. (e-mail: chen.gong@njust.edu.cn, csjyang@njust.edu.cn)

X. Gao is with the Chongqing Key Laboratory of Image Cognition, Chongqing University of Posts and Telecommunications, Chongqing 400065, P.R. China. (e-mail: gaoxb@cqupt.edu.cn)

Corresponding authors: Lin Zhao (linzhao@njust.edu.cn) and Jian Yang (csjyang@njust.edu.cn).

[1]http://human-pose.mpi-inf.mpg.de/#results

[2]http://cocodataset.org/#keypoints-leaderboard



(a) The architecture of Hourglass (b) The architecture of HRNet

Fig. 2. Illustration of representative networks for human pose estimation that couple spatial information preserving with semantic information acquisition. For both Hourglass in (a) and HRNet in (b), a high-to-low process is adopted to acquire semantics and the high-resolution features in low levels are preserved by skip connections or additional convolutions.

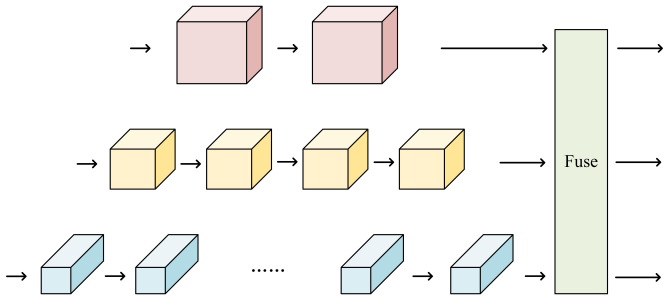


Fig. 3. An illustration of a single "parallel pyramid" module. The module is constructed by several parallel branches, and each branch takes in and processes features of different resolutions. The bottom branch containing deep and wide sub-networks only processes low-resolution features, and high-resolution features are only dealt with in the top branch containing narrow and shallow networks. At the end of each module, the features of different resolutions are fused to give the output. Multiple modules can be stacked to form a parallel pyramid network.

weight backbones like MobileNets [11] for pose estimaiton. Nevertheless, the results produced by these networks can be fairly poor. Alternatively, network compression techniques like knowledge distillation [12] can be resorted to, but a big teacher network is still needed to be trained ahead. This paper proposes a novel network architecture, which increases efficiency without reducing accuracy. Learning from the-state-of-the-art systems [13], [14], accurate semantic acquisition and precise spatial information preserving are two key factors in the success of keypoint localization. The existing deep models of pose estimation [13], [14], [15], [16] broadly refer to the classic networks such as ResNet [8] and VGG [17] developed for ImageNet classification [18]. Specifically, the network takes high-resolution inputs and gradually decreases resolution and increases width while going deep. To keep spatial location in the process, skip connections [13] or additional convolution layers [14] are utilized to retain low-level high-resolution features. That is to say, preserving spatial location is coupled with extracting semantic features in these networks. We show the architectures of the representative networks of pose estimation in Fig. 2.

Aiming to strike a fine balance of strong performance and high efficiency for human pose estimation, we propose to separate spatial information preserving from semantic acquisition. In our architecture, only low-resolution features are taken as input and processed to conduct semantic analysis. For accurate semantic extraction, we make the network deep and keep a big width throughout. A separate sub-network is utilized to hold spatial locations with high-resolution input, and the network is shallow and narrow. We fuse the outputs of sub-networks to produce the final representations for pose estimation. Fig. 3 illustrates the network structure. Convolutional layers of different depths and widths build a "parallel pyramid" network (PPNet), and features of different resolutions are computed in parallel within the pyramids. For easier training and better performance, we introduce intermediate supervision after the output of each pyramid.

The design feature of separating semantic and spatial information enjoys substantial benefits. (1) Strong performance can be guaranteed. Because the robustness of semantic features mainly relies on the depth and width of the network, and precise spatial location is maintained by high-resolution features. The proposed network meets both requirements to produce accurate results. (2) High efficiency is able to be achieved. Comparing to the existing networks [13], [14], [15], [16] for pose estimation, our network mainly makes calculation on low-resolution features. Hence to give the same level of performance, less computation and memory is required. We conduct extensive experiments on two widely-used benchmark datasets: the MPII Human Pose dataset [9] and the COCO keypoint detection dataset [10]. The proposed network demonstrates the consistent superiority of efficiency over the-state-of-the-art models, using about two thirds of the computation cost and half of the parameter size to deliver better keypoint localization performance.

II. Related Work

Estimating human pose efficiently in images or videos is a long-standing ambition in computer vision. Traditional methods utilize hand-crafted features like HOG [19] and Classifiers such as SVM [20] to detect body parts in images, then probabilistic graphical models [21], [22], [23] depicting the constrains of body structure are adopted to infer the most probable part locations. The pictorial structure models [24], [25], [26] and the deformable part models [27], [28], [29] are the representative methods. With the great power shown by deep convolutional neural networks (ConvNets) [30] on many computer vision tasks, the research nowadays on human pose estimation has been dominated by deep models. Toshev et al. [31] propose "DeepPose" to directly regress the coordinates of joints from images. Though it is a brutal utilization of deep networks on pose estimation, the performance surpasses most traditional methods. Inspired by the pictorial structure model, Tompson et al. [32], [33] replace HOG features and SVM classifiers with ConvNets, and generate heatmaps of joints by conducting convolutions at multiple image scales. Another feature of their method is jointly using a graphical model with a ConvNet to learn spatial relationships between joints. The final x, y coordinates of joints are obtained by inferring on the heatmaps with the graphical model. Several other works [34], [35] follow the framework and make refinements mainly on how to better combine graphical models with ConvNets. As the

ability of ConvNets is further improved by the smart designs like VGGNet [17] and ResNet [8], recent works [13], [14], [15], [16], [36] no longer make use of graphical models to help infer the coordinates of joints from heatmaps. Because a very deep ConvNet can have a large receptive field to perceive abundant information in an image, the spatial relationships of joints are also able to be learned by ConvNets. The latest methods [37], [38], [39], [40] all produce impressive results, and only simple post-processing techniques are utilized to get joint locations.

Recent research on human pose estimation mainly focuses on how to design the network, aiming to produce more precise heatmaps. There are two mainstream paradigms to obtain poses of multiple people in an image, two-stage top-down [14], [15] and one-stage bottom-up [41], [42], [43], [44], the difference is whether bounding boxes of persons are used. Nevertheless, the network for generating heatmaps is always shared, for example both the state-of-the-art top-down [40] and bottom-up methods [44] use the same network HRNet [14]. Hence we review closely-related network design techniques of human pose estimation developed mainly under the top-down paradigm, *i.e*. methods for single-person pose estimation. A successful network generally needs to acquire semantic and spatial information about joints to serve pose estimation, we talk about related works from these two aspects.

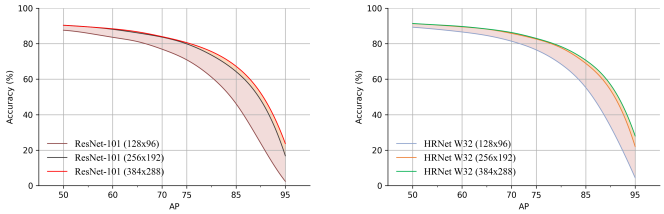
Semantic information acquisition. From early networks like AlexNet [7] for ImageNet [18] classification competition, accurate semantic information acquisition is always one of the most important goals of network design. Typically, a network outputs the final low-resolution high-level features to depict semantic information. ResNet [8] gradually decreases feature resolution and increases channels while the network goes deeper. This strategy demonstrates its effectiveness to produce good high-level representations on various tasks such as object detection [45] and semantic segmentation [46]. Utilizing RseNet as the backbone, both Chen et al. [16] and Xiao et al. [15] successfully estimate human pose in images. Hourglass [13] adopts an encoder-decoder architecture to do human pose estimation, and semantic information is explored in its encoder structure. Different from ResNet, because hourglass uses a more symmetric topology to conduct bottom-up and top-down processing, multiple hourglasses are stacked to make the network deep enough to produce excellent high-level representations. As hourglass provides an elegant framework for human pose estimation, various works make refinements based on it. Chu et al. [47] propose a multi-context attention mechanism to make hourglass explore more precise semantics. Yang et al. [48] introduces a feature pyramid module to replace the basic residual block of hourglass, which hopes to enlarge receptive field. Chen et. al [49] utilize adversarial learning [50] to better train the Hourglass network. All these works make Hourglass generate better results. Like ResNet, HRNet [14] also progressively decreases feature resolution and increases channels to get the final high-level representation. Increasing the width of the sub-networks in HRNet is demonstrated to be able to acquire more accurate semantic information, making better results [51]. Mutli-scale attention is also introduced by Jiang et al. [52] into HRNet to further boost the performance.

Spatial information preserving. Like semantic segmentation [46], human pose estimation requires pixel-level understanding of the input image to give the coordinates of joints. Thus high-resolution representations are needed to be recovered from low-resolution semantics. Xiao et al. [15] utilize a light upsample process with just dilated convolutions to obtain high-resolution heatmaps. Though this simple strategy produces acceptable results, recent studies [14], [16], [53] show fusing mutli-scale high-level and low-level features can bring benefits, since spatial information is preserved in the low-level high-resolution features. Learning from PSPNet [54] and DeepLab [55] for semantic segmentation, Chen et al. [16] combine the pyramid features at the different stages of ResNet in the low-to-high upsample process. Hourglass [13] uses skip connections to copy the high-resolution features in low-level layers to the mirrored layers in its symmetric architecture. This fusion technique also can be found in U-Net [56] and encoder-decoder [57]. With the goal to produce excellent high-resolution representations, HRNet [14] proposes to maintain the resolution of low-level features by extra convolutional layers while the high-to-low downsample process proceeds to acquire semantics. And after every several convolutional layers, features in multiple scales are fused. Inspired by deep fusion [53], the whole network repeats multi-scale fusion quite a lot times to give the final high-resolution representations.

Our parallel pyramid network. Our approach learns from successful experience of the above works for human pose estimation, and tries to deliver the same level performance with less memory usage and computational cost. The main difference lies in how to efficiently acquire semantic and spatial information. The existing networks such as Hourglass [13] and HRNet [14] all utilize a high-to-low downsample process to get high-level representations, spatial information preserving is coupled with semantic acquisition. Our network handles these two tasks separately, semantic acquisition is achieved only with low-resolution features and high-resolution spatial information is independently taken care of. We adopt multi-scale fusion to give the final results like other works for dense prediction [14], [16], [54], [55]. From "DeepPose" [31] to the work of Tompson et al. [32] and follow-ups [13], [14], [40], the mutli-channel heatmap representation of joints disentangles semantics from spatial coordinates, which secures significant improvement of performance. In this paper, we bring the idea of separating spatial information preserving from the acquisition of semantics into the design of network architecture, hoping this can promote efficiency.

III. Parallel Pyramid Networks

We follow the widely-used pipeline by contemporary state-of-the-art networks [13], [14], [15], [36] to do human pose estimation. Taking an image $\mathbf{I}$ of size $W \times H \times 3$ as input, the network outputs K heatmaps $\{\mathbf{H}_1, \mathbf{H}_2, \cdots, \mathbf{H}_K\}$, where $\mathbf{H}_k$ is the heatmap of the kth joint. The final coordinate x, y of each joint is obtained from its corresponding heatmap by simple post-processing operations. This paper mainly pays attention on the efficiency of human pose estimation with the aim of designing a new network, achieving the same-level



(a) SimpleBaseline (b) HRNet

Fig. 4. The performance variation caused by different resolutions of the input image. The detailed AP values obtained by two networks SimpleBaseline [15] and HRNet [14] from AP^{50} to AP^{95} are shown in the figure. The COCO validation set is used for evaluation.

performance as current state-of-the-art methods by using less computation and memory.

A. Motivation

The performance of nowadays human pose estimation algorithms [14], [42], [58] has reached the level which can provide reliable results for high-level computer vision tasks like action recognition [59]. And trying to further improve the performance of the famous networks such as Hourglass [13] and HRNet [14] usually needs to pay the cost of considerably more computation. Because the efficiency of a method is also very important for practical applications, this drives us to present a low-cost but good-performance network for human pose estimation.

A direct idea to increase efficiency is to use a lower-resolution input. However, the difficulty lies in maintaining strong performance. Fig. 4 shows standard average precision (AP) obtained by two recent networks SimpleBaseline [15] and HRNet [14], tested on the COCO [10] validation dataset. Clearly for both networks, the APs decrease with lower-resolution inputs. But if we only look at the APs at relatively low OKS (Object Keypoint Similarity, $\mathrm{OKS} < 0.70$), for both networks, the performance degradation caused by lower resolution is not obvious. This indicates an interesting direction to increase efficiency. Because the APs at relatively low OKS like AP^{50} (AP at $\mathrm{OKS} = 0.50$) depict the accuracy of keypoint detections with not very high localization precision, *i.e.* the semantics of keypoint detections is right and just the spatial coordinates are not extremely close to the ground truth, we can have a conjecture that semantics is able to be acquired using a moderately low resolution input. It also accords with our intuition, as a human body in a low-resolution image still holds its shape and appearance. On the other hand, semantics is mainly represented by high-level features in a deep model. Hence we can put forward a hypothesis that semantics can be obtained through a deep and wide network but with a low-resolution input.

Except for accurate semantic information acquisition, p-reserving the spatial location of features is also essential for human pose estimation. And it asks for high resolution features to keep spatial information, which seems contrary to the expectation of using low-resolution features throughout the model. Nevertheless, learning from successful models such as Hourglass [13] and HRNet [14], spatial information can be well obtained in low-level features, *i.e.* we can utilize a shallow network to process high-resolution features for the preservation of spatial location. This means the requirements of semantic and spatial information acquisition are not highly correlated. It motivates us to introduce a novel network architecture that takes care of semantics and the spatial location of features separately, and we propose a "parallel pyramid" network (PPNet) with the aim of increasing the efficiency of human pose estimation.

B. Parallel Pyramid Network Design

Coupled spatial information with semantics. Existing networks all use low-level high-resolution features to preserve spatial location. And high-level low-resolution features representing semantics are also obtained from low-level features through a sequence of convolutions, decreasing the resolution gradually in the process. Let $\mathbf{F}^{s_i}$ be the feature of resolution s_i, this process (*e.g.* containing three resolutions s_1, s_2, and s_3) can be denoted as:

$$\begin{array}{ccccccc} \mathbf{F}^{\mathrm{in}} \rightarrow \mathbf{F}^{\mathrm{s}_1} \rightarrow & \dashrightarrow & \dashrightarrow & \dashrightarrow \rightarrow & \oplus \rightarrow \mathbf{F}^{\mathrm{out}}. \\ \searrow \ \mathbf{F}^{\mathrm{s}_2} \rightarrow & & & \dashrightarrow \nearrow & \\ \searrow \ \mathbf{F}^{\mathrm{s}_3} & & & \nearrow & \end{array} \tag{1}$$

The resolution of features $\mathbf{F}^{in}$ and $\mathbf{F}^{out}$ is also s_1. Typically down-sample layers are utilized to halve the resolution, thus $s_2 = \frac{1}{2}s_1$ and $s_3 = \frac{1}{2}s_2$. It is obvious that all the features, especially $\mathbf{F}^{s_1}$ and $\mathbf{F}^{s_2}$, need to present both semantic and spatial information well. To give good performance, widely-used networks such as ResNet [8] and HRNet [14] all put heavy computation on the features of the relatively high resolutions *i.e.* s_1 and s_2.

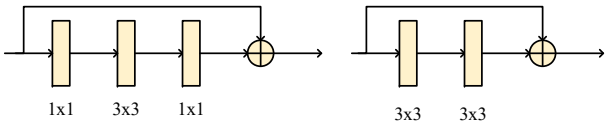
Separating Spatial information from semantics. We use separate subnetworks to deal with semantic acquisition and the preserving of spatial locations. The input feature $\mathbf{F}^{in}$ is directly downsampled to several features of relatively low resolutions. The feature of the lowest resolution is processed to get semantics, while the feature of the highest resolution is taken care of to keep spatial information. We also make use of the features of the middle resolutions to provide more information for strong performance. An illustration of our design, using three resolutions, is given as follows:

$$\begin{array}{ccccc} \mathbf{F}^{\mathrm{in}} \rightarrow & \mathbf{F}^{\mathrm{s}_1} & \rightarrow \oplus \rightarrow & \mathbf{F}^{\mathrm{out}}. \\ \searrow & \mathbf{F}^{\mathrm{s}_2} & \nearrow & \\ \searrow & \mathbf{F}^{\mathrm{s}_3} & \nearrow & \end{array} \tag{2}$$

In this structure, most computation will be put on semantic acquisition from the features of the relatively low resolutions, and only limited computation will be made to preserve spatial location on the features of the highest resolution .

C. Parallel Pyramid Network Instantiation

Following the common network structure of human pose estimation [13], [15], [32], the proposed parallel pyramid network consists of three parts, a stem bringing the resolution of features to a quarter of the input resolution, a main body outputting the high-level features with both semantic and



(a) Residual module (b) Residual block

Fig. 5. Illustrating the residual module and the residual block used in PPNet.

spatial information of keypoints, and a regressor producing the heatmaps. We mainly put effort into the design of the main body, presenting a novel architecture to increase the efficiency while keeping strong performance without any bells and whistles.

Stem. The stem network takes an image as input and provides preliminary features for the main body. It starts with two consecutive 3×3 convolutional layers with stride 2, decreasing the resolution down to a quarter of the input. Like the ResNet-50 [8] and Hourglass [13], four subsequent residual modules (shown in Fig. 5(a)) with the width 64 are utilized for more convolutions. Finally, several 3×3 convolutions with stride 1 or 2 are followed to change the resolution and width of feature maps, meeting the demands that the parallel pyramid module makes on its input.

Parallel Pyramid Module. The parallel pyramid modules form the main body of the network. There are three parallel branches (indexed by $i = 1, 2, 3$ from top to bottom) in a parallel pyramid module, each branch is an independent sub-network and processes the feature maps of one resolution throughout. The resolution of feature maps in the branches decreases from top to bottom. Let s_i be the resolution of feature maps in each branch, for simplicity and easy implementation, we set

$$s_{i+1} = \frac{1}{2} s_i. \tag{3}$$

s_1 is equal to a quarter of the resolution of the input image. In contrast, the depth (the number of convolutional layers) and width (the number of channels) of the sub-networks increase from top to bottom. Following a common practice, both the ratio of depth and width between two adjacent branches is 2. Using D_i and W_i to denote the depth and width of each sub-network, the relation can be written as:

$$\begin{aligned} D_{i+1} &= 2D_i, \\ W_{i+1} &= 2W_i. \end{aligned} \tag{4}$$

Hence both the shape of feature maps and the architecture of sub-networks in a module resemble a pyramid (as shown in Fig. 3), which inspires us to name the design a parallel pyramid module.

We make extensive use of residual blocks to construct the network. Fig. 5(b) gives an illustration of the block, there are two 3×3 convolutions in each block, and each convolution is followed by batch normalization and the non-linear activation ReLU. The convolutions are with the stride 1 and the number of channels (the width) is kept the same. We use n residual blocks with the width C to make up the top branch of a parallel pyramid module, thus the middle and bottom branch contain $2n$ and $4n$ blocks with the width $2C$ and $4C$ respectively. The size of a parallel pyramid module is represented as Dn-WC, where n and C are the depth and width of the top sub-network. For example, D2-W32 means there are 2, 4, and 8 residual blocks in the three sub-networks and the width are 32, 64, and 128 respectively.

The output of the sub-networks are fused at the end of a parallel pyramid module. Like the mutli-scale fusion in HRNet [14], the output of each branch is aggregated with the outputs of the other two branches, and the aggregated feature maps are still with the same resolution and width of its corresponding branch. Denoting the outputs of the sub-networks before fusion as: $\{\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3\}$, the outputs after fusion still contain 3 feature maps: $\{\mathbf{Y}_1, \mathbf{Y}_2, \mathbf{Y}_3\}$, where the resolution and width of $\mathbf{Y}_i$ are the same as $\mathbf{X}_i$. The fusion is conducted as follows:

$$\mathbf{Y}_i = \sum_{j=1}^{3} f(\mathbf{X}_j, i), \quad i, j = 1, 2, 3. \tag{5}$$

$f(\mathbf{X}_j, i)$ changes the resolution and width of $\mathbf{X}_j$ to that of $\mathbf{X}_i$. If $j = i$, it is just an identify connection. When $j > i$, we simply use nearest neighbour interpolation for upsampling and a 1×1 convolution to align the number of channels. 3×3 convolutions with the stride 2 are adopted in case of $j < i$, $2\times$ downsampling needs one strided convolution and $4\times$ downsampling asks for two.

Heatmap regressor. Heatmaps are predicted only from the feature maps of the highest resolution, *i.e.* $\mathbf{Y}_1$. Two consecutive rounds of 3×3 convolutions are applied to further process the feature maps, then a 1×1 convolution is utilized to produce the heatmaps.

Intermediate supervision. The main body of our network is built by connecting multiple parallel pyramid modules end-to-end, feeding the fused outputs of one module as input into the next. Intermediate supervision is exercised after each module, when the prediction of intermediate heatmaps are able to be made and a loss can be applied on. As demonstrated by early works [13], [36], stacking multiple modules is able to make subsequent modules process features at both local and global context and reconsider the overall coherence of the features for more accurate predictions. Intermediate supervision allows for multiple re-evaluation of the high-level features partway through the full network, which makes the training easier and more stably.

The intermediate heatmaps are re-integrated back into the feature space by using a 1×1 convolution to align the number of channels. Same to Hourglass [13], we also add these intermediate features along with the input of the current parallel pyramid module. And all these integrated features make up the input of the following parallel pyramid module. We illustrate this process of intermediate supervision in Fig 6.

An illustration of the proposed parallel pyramid network is given in Fig. 7. For the sake of simplicity, all parallel pyramid modules used in our network are of the same size and structure, but the weights are not shared across the modules. And each parallel pyramid module produces the predictions of heatmaps. We apply a loss to the predicted heatmaps of each module with the same ground truth. Let m be the number of

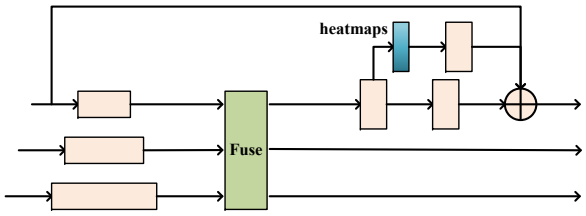


Fig. 6. Illustration of the intermediate supervision process in PPNet. Intermediate heatmaps are obtained on the top branch with the highest feature resolution, and a loss is applied. The intermediate heatmap will be re-integrated into the feature space, and these features will be added with the input of the current parallel pyramid module to make up the input of the next module.

parallel pyramid modules used in the network, Mm-Dn-WC indicates the size of the whole network. For example, M4-D2-W32 depicts a parallel pyramid network of small size, which we will discuss in the experiments.

D. Analysis

The proposed parallel pyramid network draws upon several valuable experience of previous works to deliver a good performance on human pose estimation. Our main contribution is the proposal of disentangling semantic and spatial information in the design of the network, and this idea promotes efficiency while doing no harm to the performance. Here we give some analysis about the similarities and differences between our network and previous models, especially two most famous and successful ones including Hourglass [13] and HRNet [14]. All three networks are able to produce high-level features with accurate semantics and preserve the spatial location of features in the process, which makes all models give promising results without big gap. The main difference lies in the mechanism and thus the efficiency to achieve these. We compare the proposed network with the other two separately in detail below. The architectures of the three networks are shown in Fig. 2 and Fig. 3.

The structure of the stacked Hourglass network [13] is a single pipeline, and an encoder-decoder architecture with symmetric topology is utilized. It preserves spatial information by skip layers during semantic acquisition. Our model adopts several parallel pipelines to construct the network, spatial and semantic information is processed with different resolutions in separate pipelines. Because most computation in our model is carried out on low-resolution features, less time is taken to get results of similar accuracy. Both models leverage modular design to develop a flexible network structure, and the intermediate supervision is able to be adopted for stable training of the networks. In addition, the shapes of the modules in two networks resemble a hourglass and a pyramid respectively, which makes both networks look quietly elegant.

The high-resolution network [14] explores the feasibility of dealing with features in different resolutions by multiple parallel pipelines. It still utilizes a high-to-low process to generate high-level and low-resolution representations, the high-to-low resolution sub-networks are added gradually one by one and connected in parallel. Because the features of each resolution are always maintained using convolutional layers in the corresponding pipeline, there is no explicit low-to-high process to recover high-resolution representations. Nevertheless the convolutions on high-resolution features have to be conducted from start to end, which can be resource intensive. We also adopt multiple pipelines to build the network, however, there is no high-to-low process either in our architecture. High-level semantics are obtained only using the features of low resolution, and only limited resources are put on the features of high resolution to preserve the spatial location. This design idea of separating spatial location from semantic information makes our algorithm more efficient. Moreover, it can be much more convenient for our model to adjust its size. Because modular design is utilized in the network, we can easily have small or big models just by modifying the size and number of modules, but not only changing the width.

IV. Experiments

Datasets. The Experiments are conducted on two most widely-used datasets for human pose estimation, the MPII Human Pose dataset [9] and the 2017 Microsoft COCO keypoint dataset [10]. No additional datasets like AI Challenger [60] will be used for training a better model, as we want to simply demonstrate the efficiency of the proposed algorithm.

The state-of-the-art networks for comparison. We compare our Parallel Pyramid Networks (PPNet) to three representative and the best models in the literature, described as follows:

- SimpleBaseline [15]: This model utilizes ResNet [8] as its backbone, and the predictions of heatmaps are obtained only by several deconvolutional layers. Although the structure is surprisingly simple, its performance surpasses other works like CPN [16] based on ResNet. Hence SimpleBaseline is the representative model of works using ResNet as the backbone network, and makes the baseline of human pose estimation.
- Hourglass [13]: With its elegant architecture and strong performance, Hourglass may be the most popular network for human pose estimation. Numerous best-performing methods on the MPII dataset [9] are based on Hourglass.
- HRNet [14]: It is the current state-of-the-art network of human pose estimation. The latest works such as DARK [61] and UDP [62] that report the best results on the COCO dataset [10] all use HRNet as the backbone model.

Although various methods has been proposed to push the boundaries of human pose estimation, the backbone model of these works is generally one of the three networks. We are interested in a more efficient backbone network for human pose estimation, experiments are carried out thoroughly to demonstrate the superiority of the proposed PPNet over the above three networks. PPNet can certainly benefit from the model-agnostic refinement techniques, such as attention mechanism [63], distribution-aware coordinate representation [61], and unbiased data processing [62]. Nonetheless, pursuing the best results on the MPII or COCO benchmark is not the goal of this paper.

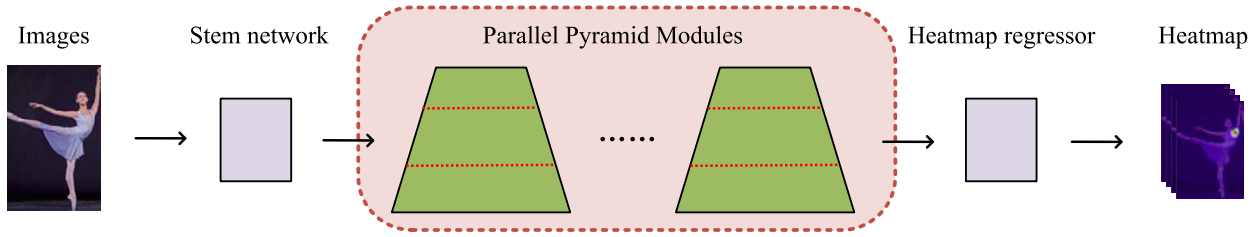


Fig. 7. The framework of using PPNet to do human pose estimation. Our main work is the design of the architecture of the proposed parallel pyramid network, which makes the main body of the framework.

Instantiated models of PPNet. We are able to build a variety of models by setting different depth and width of the parallel pyramid module. Besides, the number of modules can be adjusted to change the model size. As described in Section III-C, we use Mm-Dn-WC to express the size of a PPNet model, where m, n, and C are the parameters. In the experiments, eight different networks of PPNet are constructed by taking two different values for each of the three parameters, *i.e.* $m = 2, 4$, $n = 2, 3$, and $C = 32, 48$. We will mainly study several of them. M2-D2-W32 is the smallest network providing the highest efficiency, and the biggest one is M4-D3-W48 delivering the best performance.

Experimental settings. Model-agnostic improvements may significantly affect the results of human pose estimation. Some examples demonstrated effectiveness include more data augmentation [64], specialized data-processing [62], or multi-scale testing [47]. To reveal the pure model performance and make fair comparisons, we train and test all networks under the same standard settings, excluding any discrepancies except for the model that can potentially account for differences in the reported accuracy. All compared networks, including our PPNet, are implemented in Python with Pytorch 1.0, and executed on a machine with one NVIDIA 2080Ti GPU. For SimpleBaseline [15] and HRNet [14], we directly use their public code. For Hourglass [13] and our PPNet, we implement them exactly according to the description in the paper. All models share the same code for data augmentation and post-processing. ImageNet pre-training is well-known to bring benefits, however, pre-training all the compared models on such a large dataset is beyond the capacity of our resource. In order to be fair and truly reflect the power of the models, we train all networks by initializing the weights from normal distribution. The source code and trained models of PPNet will be public available at https://github.com/sharling-lz/ppnet.

A. Evaluation on COCO Keypoint Detection

The performance on the COCO keypoint dataset [10] is the current touchstone of a good model. The dataset presents naturally challenging imagery data with unconstrained environments, different scales, and various occlusion patterns. Each person instance is labelled with 17 keypoints. We train all models on COCO train2017 dataset, which contains $57K$ images and $150K$ person instances. The evaluation of our approach is made on both the val2017 set and test-dev2017 set, containing $5K$ and $20K$ images respectively.

Evaluation metric. The standard average precision (AP) and average recall (AR) scores based on Object Keypoint Similarity (OKS) are utilized to evaluate the performance. OKS is defined as a similarity measure of poses mimicking IoU in object detection, which is formulated as:

$$\mathrm{OKS} = \frac{\sum_{\mathrm{i}} \exp(-\mathrm{d}_{\mathrm{i}}^2/2\mathrm{s}^2\mathrm{k}_{\mathrm{i}}^2)\delta(\mathrm{v}_{\mathrm{i}} > 0)}{\sum_{\mathrm{i}} \delta(\mathrm{v}_{\mathrm{i}} > 0)}, \tag{6}$$

where d_i is the Euclidean distance between each detected keypoint and its corresponding ground truth, s is the object scale, k_i is a keypoint-specific constant that controls falloff, v_i is the visibility flag of the ground truth. We report AP (the mean of AP scores at $\mathrm{OKS} = 0.50, 0.55, \ldots, 0.90, 0.95$), AP^{50} (AP at $\mathrm{OKS} = 0.50$), AP^{75}, AP^{M} (medium objects), AP^{L} (large objects) and AR (the mean of AR scores at $\mathrm{OKS} = 0.50, 0.55, \ldots, 0.90, 0.95$). To compare the complexity of a model, we give the number of parameters and FLOPs. In addition, the speed of training $(\mathrm{samples/s})$ is presented to directly reflect the efficiency.

Training. The image of each person instance is cropped using the corresponding bounding box, and the aspect ratio of height and width is fixed as $4:3$. The image is resized to 256×192 or 384×288 after cropping. We use the standard data augmentation including random horizontal flipping, scaling ($[.65, 1.35]$), and rotation ($[-45°, 45°]$).

All models are trained following the same learning schedule of HRNet [14]. The total number of epochs is 210, the base learning rate is decayed by a factor of 10 at the 170th and 200th epochs, respectively. For SimpleBaseline[15] and HRNet [14], the Adam optimizer [65] is utilized and the base learning rate is set as $1\mathrm{e}-3$. For Hourglass [13] and PPNet, we adopt RMSProp [66] for optimization with a learning rate of $2.5\mathrm{e}-4$. The batch size of 32 is used for training, except for the very large models, we decrease the batch size to 24 or 16 as the memory of GPU will exceed.

Testing. The common two-stage top-down paradigm [14], [15], [16] is used during testing. For both validation set and test-dev set, we use the person detections released by SimpleBaseline [15]. Following the standard post-processing protocol of testing, the heatmaps from the original and flipped images are averaged to compute the heatmap. And the keypoint location is predicted by applying a quarter offset to the location of the highest heatvalue in the direction from the highest response to the second highest response.

Results on the validation set. All the compared models are divided into four categories according to the complexity

and efficiency, and we compare the results of the models in each category. We want to see whether our PPNet can achieve a better balance between accuracy and efficiency comparing to the existing models. Table I presents the results produced by models of mini-size. The models in this category can process around 200 images per second with the input size 256×192 on our machine (one NVIDIA 2080Ti GPU). Among the models of the compared networks, only SimpleBaseline using ResNet-50 as the backbone reaches the standard. Our PPNet using both the backbones of M2-D2-W32 and M2-D3-W32 can be trained at a higher speed, and the number of parameters and GFLOPs are much lower. The smallest one PPNet (M2-D2-W32) reaches a speed of $235\,\mathrm{samples/s}$ and uses 4.8x fewer parameters and 2.7x fewer GFLOPs than SimpleBaseline (ResNet-50). Nonetheless, it improves AP by 1.0 and 1.5 points under the input size 256×192 and 384×288 respectively. Our a little bigger model PPNet (M2-D3-W32) further increases accuracy, while its complexity and training speed are still competitive.

The second category contains models of small size, including SimpleBaseline (ResNet-101), Hourglass (2-stage), and PPNet (M4-D2-W32). We give the results of these models in Table II. Compared to SimpleBaseline (ResNet-101), our PPNet (M4-D2-W32) shows absolute superiority. Though the number of parameters and GFLOPs is 4.2x fewer and 2.5x fewer respectively, the improvement in AP is 1.5 points and 1.3 points under the input size 256×192 and 384×288, respectively. Compared to Hourglass (2-stage), PPNet (M4-D2-W32) has slightly more parameters, but uses less GFLOPs and trains faster. For the input size 256×192, AP obtains 1.0 points gain from 71.7 to 72.7. The input size 384×288 is not trained and tested on Hourglass, because the Hourglass network needs a 64x down-sampling to its lowest resolution, and it can not be achieved on this resolution.

The results of the models in medium size are compared in Table III. Except for SimpleBaseline, the other three methods (*i.e*. Hourglass, HRNet, and PPNet) deliver a performance on a par with each other under the input size 256×192. Nevertheless, our PPNet (M4-D3-W32) can be trained at a speed of $130\,\mathrm{samples/s}$, which is 1.7x faster than Hourglass (4-stage) and 1.3x faster than HRNet (W32). For the input size 384×288, PPNet (M4-D3-W32) achieves a score of 74.7 AP, outperforming HRNet (W32) by 0.8 AP. At the same time, PPNet (M4-D3-W32) still maintains its higher efficiency.

We compare the large models finally in the fourth category. Models of this size are used to compete for the state-of-the-art performance, but the cost of computation is usually huge. Table IV reports the results for comparison. Hourglass (8-stage) obtains the highest score of 74.8 AP for the input size 256×192[3]. The result of PPNet (M4-D3-W48) is slightly worse, however, its efficiency far surpasses Hourglass (8-stage). For the training speed, PPNet (M4-D3-W48) can maintain a speed of $95\,\mathrm{samples/s}$, but the speed of Hourglass (8-stage) is just $37\,\mathrm{samples/s}$. HRNet (W48) is the state-of-the-art model under the input size 384×288. PPNet (M4-D2-W48) already makes an improvment by 0.5 AP with less than half of the parameters and two thirds of the GFLOPs. Making the backbone a litter bigger, PPNet (M4-D3-W48) further achieves 0.3 points gain, while keeping its advantage in complexity and efficiency over HRNet (W48).

Conducting a comprehensive analysis of the performance by the four compared networks, we can find that SimpleBaseline [15] consistently gives the worst results. Looking at the score of AP^{50}, there is no obvious gap between the result of SimpleBaseline and the other three methods, however, the score of AP^{75} reported by SimpleBaseline can be significantly lower. This indicates the inability of SimpleBaseline to preserve precise spatial locations of keypoints, since only dilated convolutions are utilized to recover high-resolution heatmaps. All other three methods take advantage of multi-scale fusion to maintain the spatial information from low-level features, which demonstrates its necessity and effectiveness. But from the perspective of efficiency, though the number of parameters and GFLOPs of SimpleBaseline may be much larger than HRNet [14] or Hourglass [13], the training speed is surprisingly faster. The reason may lie in the repeated multi-scale fusion in the process, and it can be quite time-consuming. Thus it would be misleading if only the numbers of parameters and GFLOPs are presented for comparison. Our PPNet finds a fine balance between accuracy and efficiency, which reaches faster training speeds than SimpleBaseline [15] and delivers even stronger performance than Hourglass [13] and HRNet [14].

Results on the test-dev set. Here we only compare the results of large models of each network. Table V reports the results for comparison. The performance of several other methods that are listed on the leader board[4] is also exhibited for reference. The results are consistent with the conclusion reached on the validation set. Our most efficient model PPNet (M4-D2-W48) achieves a score of 74.8 AP, outperforming the state-of-the-art model HRNet (W48) 0.3 AP. Taking model size (#Params) and computation complexity (GFlOPs) into account, PPNet undoubtedly demonstrates its outstanding network design and can be a better backbone for human pose estimation.

B. Evaluation on MPII Human Pose Estimation

The MPII Human Pose Dataset [9] is constructed from video frames depicting a wide-range of real-world activities. Each person instance is annotated with a ground-truth bounding box and 16 keypoints. There are about $25K$ images and $40K$ person instances, where $28K$ subjects are used for training. Because the scales of person instances in MPII are mostly large and the crowding is not as heavy as in the COCO dataset, it is relatively easy to deliver good performance on this dataset. We follow the standard train/val split as in other works [32], and the evaluation is conducted on the validation set.

Evaluation Metric. We use the standard metric PCKh (head-normalized Percentage of Correct Keypoints). The head size l is used to normalize the distance, which corresponds

[3]The result reported in other papers [16], [14], [62] is just 66.9 AP. This can not be the right score of 8-stage Hourglass. We guess the difference may be caused by unfair comparisons such as using different data processing and person detectors.

[4]http://cocodataset.org/#keypoints-leaderboard

TABLE I

COMPARING THE MODELS OF MINI-SIZE ON THE COCO VALIDATION SET. OUR SMALLEST MODEL PPNET (M2-D2-W32) SURPASSES SIMPLEBASELINE (RESNET-50) BY OVER 1.0 POINTS IN AP, ONLY USING 4.8X FEWER PARAMETERS AND 2.7X FEWER FLOPS. AND THE TRAINING SPEED CAN BE MUCH FASTER. THE A LITTLE BIGGER MODEL PPNET (M2-D3-W32) FURTHER IMPROVES THE PERFORMANCE, WHILE THE EFFICIENCY IS STILL GREATER. THE UNIT OF TRAIN SPEED IS IN SAMPLES PER SECOND (SAMPLES/S).

Method	Backbone	Input size	Params	FLOPs	Train Speed	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR
SimpleBaseline [15]	ResNet-50	256×192	33.9M	8.9G	190 sam/s	69.9	88.4	77.5	66.3	76.9	75.8
PPNet	M2-D2-W32		**7.0**M	**3.3**G	**235** sam/s	70.9	88.6	78.4	67.7	77.4	76.7
PPNet	M2-D3-W32		9.7M	4.1G	205 sam/s	**71.5**	88.8	78.4	68.2	78.0	77.1
SimpleBaseline [15]	ResNet-50	384×288	33.9M	20.2G	90 sam/s	70.8	88.7	77.4	66.6	78.6	76.4
PPNet	M2-D2-W32		**7.0**M	**7.5**G	**110** sam/s	72.3	88.8	79.1	68.6	79.2	77.6
PPNet	M2-D3-W32		9.7M	9.2G	95 sam/s	**73.2**	88.9	80.0	69.5	80.1	78.4

TABLE II

COMPARING THE MODELS OF SMALL SIZE ON THE COCO VALIDATION SET. PPNET (M4-D2-W32) IS ABLE TO PRODUCE THE BEST RESULT WITH THE FASTEST TRAINING SPEED. WHETHER UNDER THE INPUT SIZE 256×192 OR 384×288, THE PROMOTION IS MORE THAN 1.0 AP. NONETHELESS, THE NUMBER OF PARAMETERS AND FLOPS IS SIGNIFICANTLY LESS THAN SIMPLEBASELINE (RESNET-101). WHILE THE PARAMETERS OF PPNET (M4-D2-W32) IS A LITTLE LARGER THAN HOURGLASS (2-STAGE), THE FLOPS IS SMALLER.

Method	Backbone	Input size	Params	FLOPs	Train Speed	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR
SimpleBaseline [15]	ResNet-101	256×192	53.0M	12.4G	135 sam/s	71.2	89.1	78.9	67.9	78.1	77.2
Hourglass [13]	2-stage HG		**6.7**M	6.3G	135 sam/s	71.7	88.8	78.9	68.4	78.3	77.2
PPNet	M4-D2-W32		12.7M	**5.0**G	**150** sam/s	**72.7**	89.4	79.8	69.4	79.3	78.2
SimpleBaseline [15]	ResNet-101	384×288	53.0M	27.9G	60 sam/s	73.0	89.3	79.7	69.3	80.5	78.7
PPNet	M4-D2-W32		**12.7**M	**11.3**G	**75** sam/s	**74.3**	89.6	81.0	70.7	81.1	79.4

TABLE III

COMPARING THE MODELS OF MEDIUM SIZE ON THE COCO VALIDATION SET. ALL THE FOUR COMPARED NETWORKS COMPETE VIGOROUSLY IN THIS CATEGORY. OUR PPNET (M4-D3-W32) STILL DELIVERS THE BEST PERFORMANCE WITH THE LOWEST COMPLEXITY AND HIGHEST EFFICIENCY. COMPARED TO HRNET (W32), PPNET (M4-D3-W32) HAS 1.6X FEWER PARAMETERS, AND THE TRAINING SPEED IS 1.3X FASTER.

Method	Backbone	Input size	Params	FLOPs	Train Speed	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR
SimpleBaseline [15]	ResNet-152	256×192	68.6M	15.7G	100 sam/s	71.3	88.9	78.9	68.0	78.1	77.2
Hourglass [13]	4-stage HG		**13.0**M	10.7G	75 sam/s	73.5	89.5	80.9	70.2	80.0	78.9
HRNet [14]	HRNet-W32		28.5M	7.1G	100 sam/s	73.4	89.5	80.7	70.2	80.1	78.9
PPNet	M4-D3-W32		18.1M	**6.5**G	**130** sam/s	**73.6**	89.4	80.8	70.2	80.1	78.9
SimpleBaseline [15]	ResNet-152	384×288	68.6M	35.4G	45 sam/s	73.0	88.9	79.8	69.3	80.5	78.7
HRNet [14]	HRNet-W32		28.5M	16.0G	50 sam/s	73.9	89.6	80.7	70.2	81.0	79.4
PPNet	M4-D3-W32		**18.1**M	**14.7**G	**65** sam/s	**74.7**	89.7	81.2	70.9	81.7	79.7

TABLE IV

COMPARING THE MODELS OF LARGE SIZE ON THE COCO VALIDATION SET. OUR PPNET (M4-D3-W48) USES 1.6X FEWER GFLOPS AND TRAINS 2.6X FASTER THAN HOURGLASS (8-STAGE), PRODUCING CLOSE RESULTS. COMPARED TO HRNET (W48), PPNET (M4-D2-W48) IS ABLE TO INCREASE THE SCORE BY 0.5 AP, WHILE USING LESS THAN HALF OF THE PARAMETERS AND TWO THIRDS OF THE GFLOPS.

Method	Backbone	Input size	Params	FLOPs	Train Speed	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR
Hourglass [13]	8-stage HG	256×192	**25.5**M	19.6G	37 sam/s	**74.8**	89.8	82.0	71.3	81.5	80.0
HRNet [14]	HRNet-W48		63.6M	14.6G	70 sam/s	74.7	89.9	82.1	71.2	81.5	80.1
PPNet	M4-D3-W48		39.2M	**12.5**G	**95** sam/s	74.4	89.9	81.2	71.0	81.1	79.7
HRNet [14]	HRNet-W48	384×288	63.6M	32.9G	32 sam/s	75.0	90.0	81.9	72.0	81.4	80.9
PPNet	M4-D2-W48		**27.1**M	**20.6**G	**53** sam/s	75.5	89.8	82.1	71.7	82.5	80.5
PPNet	M4-D3-W48		39.2M	28.1G	42 sam/s	**75.8**	90.0	82.4	72.1	82.6	80.8

TABLE V
COMPARISONS ON THE COCO TEST-DEV SET. COMPARED TO THE BEST COMPETITOR (HRNET), PPNET MANIFESTS CONSISTENT SUPERIORITY BY PRODUCING BETTER RESULTS WITH MUCH FEWER PARAMETERS AND GFLOPS.

Method	Backbone	Pretrain	Input size	Params	FLOPs	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR
G-RMI [67]	ResNet-101	-	353×257	42.6M	57.0G	64.9	85.5	71.3	62.3	70.0	69.7
CPN [16]	ResNet-Inception	-	384×288	-	-	72.1	91.4	80.0	68.7	77.2	78.5
RMPE [58]	PyraNet [48]	-	320×256	28.1M	26.7G	72.3	89.2	79.1	68.0	78.6	-
SimpleBaseline [15]	ResNet-152	N	384×288	68.6M	35.4G	72.6	91.1	80.4	69.4	78.7	78.1
Hourglass [13]	8-stage HG	N	256×192	**25.5**M	19.6G	73.9	91.9	81.9	70.8	79.5	79.2
HRNet [14]	HRNet-W48	N	384×288	63.6M	32.9G	74.5	92.0	82.2	71.0	80.5	79.9
PPNet	M4-D2-W48	N	384×288	27.1M	**20.6**G	74.8	92.0	82.2	71.5	80.6	79.9
PPNet	M4-D3-W48	N	384×288	39.2M	28.1G	**74.9**	92.0	82.3	71.5	80.7	80.0

to 60% of the diagonal length of the ground-truth head bounding box. If the distance between a predicted joint and the corresponding ground-truth position is less than αl pixels, the prediction is regarded as correct. α is a parameter to control the strictness of the metric, we report the PCKh@0.5 ($\alpha = 0.5$) score.

Training. The procedure of data processing for training is almost the same as that in COCO, except that the input size is adjusted to 256×256 as a common setting on MPII. The training strategy is also similar to that in COCO. Because the number of training samples is much less, the total number of epochs is changed to 160 and the learning rate is decayed at the 120th and 150th epochs, respectively.

Testing. The testing procedure is identical to that in COCO. Because ground-truth person boxes are provided in MPII, person detectors are not needed. This can remove the influence of person detection on the final result of human pose estimation, and the performance of human pose estimation models can be purely reflected.

Results on the validation set. Table VI shows the results of several typical models of each compared network. Hourglass [13] is able to obtain the highest accuracy. But taking the efficiency into consideration, PPNet can still be the better method. Comparing our PPNet (M4-D2-W32) with Hourglass (2-stage), we use 1.7G fewer FLOPs (6.7G versus 8.4G) but increase the PCKh@0.5 score by 0.5 (89.2 to 89.7). The performance of PPNet (M4-D3-W32) and Hourglass (4-stage) is at the same top level, but PPNet is much more efficient in terms of computation complexity (8.7G versus 14.3G FLOPs). Compared to HRNet (W32), PPNet enjoys the advantage of both better performance and higher efficiency. Taking the relatively small model (PPNet M4-D2-W32) as an example, the improvement of the PCKh@0.5 score is 0.2 and the number of GFLOPs is 1.4x fewer. For all the methods, using larger backbone networks, such as HRNet W48 and PPNet M4-D3-W48, does not bring considerably higher accuracy. The reason may be that the performance is relatively easy to get saturated on this dataset, since its size is much smaller and the conditions are not as challenging as that in COCO.

C. Discussion

The main design choices of the proposed parallel pyramid network are explored. We investigate the difference made by using different building blocks (residual module or residual block). And the suitable number of parallel branches and pyramid modules to build a network is discussed. The configurations of PPNet mainly contain the depth, the width, and the feature resolution of each parallel branch. Thus we will compare different depth and width increase ratios between two adjacent branches. In addition, using the input size 384×288, different down-sample factors of the feature resolution will be analysed. The experiments are conducted on COCO validation set [10] using the input size of 256×192 unless otherwise specified.

TABLE VII
COMPARISON OF PERFORMANCE AND COMPLEXITY BETWEEN MODELS CONSTRUCTED BY RESIDUAL MODULES AND RESIDUAL BLOCKS. THE EXPERIMENTS ARE CONDUCTED ON THE COCO VALIDATION SET [10]. THE INPUT IMAGE SIZE IS 256×192. RES. B = RESIDUAL BLOCK, RES. M = RESIDUAL MODULE.

Model	Res. B	Res. M	Parmas	GFLOPs	AP
M2-D2-W32	✓		7.0M	3.3	70.9
		✓	10.8M	5.4	71.6
M4-D2-W32	✓		12.7M	5.0	72.7
		✓	20.9M	8.6	73.6

Residual block or residual module. In the above experiments, the PPNet models are all constructed by residual blocks (Fig. 5(b)). However, residual modules (Fig. 5(a)) are also can be used to build the models. Taking PPNet M2-D2-W32 and M4-D2-W32 as examples, Table VII presents a comparison of performance and complexity between the models constructed by residual blocks and residual modules. Using residual modules gives a little better performance, but the number of parameters and GFLOPs are also larger. Considering the balance between accuracy and efficiency, we choose residual blocks to build PPNet.

The number of parallel branches. Our PPNet contains 3 parallel branches, that is to say two times of down-sampling are conducted in the main body of the network. As we know, the existing networks such as ResNet [8] and HRNet [14] typically do down-sampling three times in the process. Thus whether adding one branch to PPNet can be a better choice. To do the experiment, a miniature network PPNet M2-D2-W32 and a small network PPNet M4-D2-W32 are selected as the baselines. Fig. 8 presents the results of adding one branch. Compared to both baselines, using four parallel branches significantly increases the number of parameters and GFLOPs, but the promotion of performance is very limited. The reason

TABLE VI
COMPARISONS ON THE MPII VALIDATION SET. THE PERFORMANCE OF PPNET IS ON A PAR WITH HOURGLASS AND HRNET, BUT IT USES MUCH LESS PARAMETERS AND GFLOPS TO REACH HIGH LEVEL.

Method	Backbone	Params	FLOPs	Hea	Sho	Elb	Wri	Hip	Kne	Ank	Mean
SimpleBaseline [15]	ResNet-152	68.6M	21.0G	96.3	95.1	88.8	82.3	88.3	83.6	79.6	88.3
Hourglass [13]	2-stage HG	**6.7**M	8.4G	96.3	95.4	89.5	84.7	88.7	84.8	81.1	89.2
Hourglass [13]	4-stage HG	13.0M	14.3G	**97.0**	**96.0**	**90.5**	**86.3**	**89.4**	**86.5**	**82.9**	**90.2**
HRNet [14]	HRNet-W32	28.5M	9.5G	96.8	95.8	89.8	84.8	88.9	85.6	81.6	89.5
PPNet	M4-D2-W32	12.7M	**6.7**G	96.9	95.8	90.1	85.2	88.8	85.6	81.8	89.7
PPNet	M4-D3-W32	18.1M	8.7G	96.9	**96.0**	90.4	85.5	**89.4**	86.3	82.5	90.0

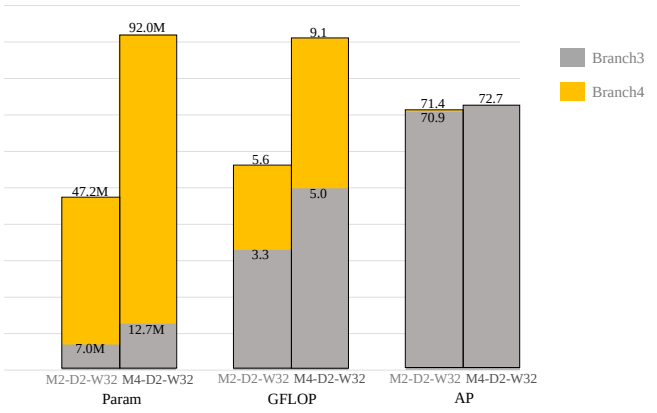


Fig. 8. Comparison of model complexity and validation accuracy while PPNet uses three branches and four branches. Two networks of different sizes M2-D2-W32 and M4-D2-W32 are selected to do the evaluation.

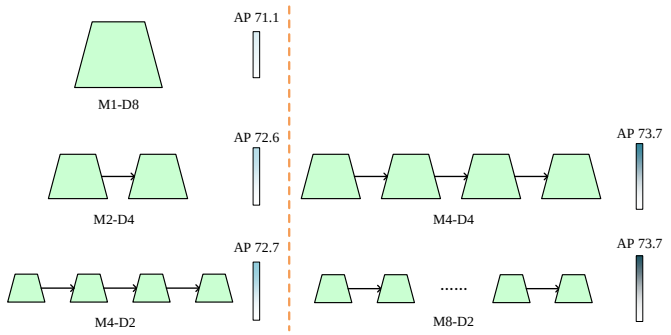


Fig. 9. Comparison of validation accuracy produced by the networks of the same size but with different number of modules. The three networks in the left part are the same size, the size of the two networks in the right part is two times of those in the left part.

may be features in a very low resolution like 8×6 can not provide enough information, and it is not worth to do deep processing.

The suitable number of modules. To explore the effect of using different number of modules, we must ensure the change in performance is caused by changing the architecture shape but not attributed to an increase in network size. For this purpose, PPNet with the backbone M4-D2-W32 is utilized as the baseline network, which consists of 4 parallel pyramid modules and the depth of each module is 2. To make the comparison, we change the architecture of the baseline network to have two variations M1-D8-W32 and M2-D4-W32, respectively. Hence it is able to see, with the same network size, whether increasing the number of modules but decreasing the module size can produce better performance. The comparison of the three networks is illustrated in the left part of Fig. 9. As can be seen, there is a wide gap (1.2 AP) between the performance of PPNet M1-D8-W32 and M2-D4-W32. This demonstrates the necessity of stacking multiple modules, because the multi-scale fusion and intermediate supervision after each module can make subsequent modules take in more information and do a deeper reconsideration of the features. Increasing the number of modules from 2 to 4 further brings 0.1 AP improvement, which shows more modules can be better. This raises a question if increasing the number of modules can always get better results. The right part of Fig. 9 shows the results of PPNet M4-D4-W32 and M8-D2-W32. The performance of these two networks is almost the same. Considering using more modules will slightly increase the number of parameters and GFLOPs, 4 modules can be a better choice. This is why we use 2 modules to construct PPNet in mini size and 4 modules to build bigger networks.

The depth increase ratio. With the goal of improving efficiency, we make the first parallel branch (processing high-resolution features) of PPNet shallow and the last parallel branch (processing low-resolution features) deep. Thus it needs to set a suitable depth increase ratio between two adjacent branches, taking both performance and efficiency into consideration. Based on PPNet M2-D2-W32, we compare four depth increase ratios, 2-2, 2-3, 3-2, and 3-3. Taking the increase ratio 2-3 as an example, because in PPNet M2-D2-W32 the first branch has 2 residual blocks, there are 4 and 12 residual blocks in the second and third branches respectively. The width increase ratio between adjacent branches is kept as 2. Fig. 10 compares the model complexity and performance between the four depth increase ratios. Larger depth increase ratios improve the model accuracy, however, the number of parameters and GFLOPs also becomes larger. We select the depth increase ratio 2 between adjacent branches, because efficiency is the main priority of building a PPNet network.

The width increase ratio. Like the depth increase ratio, we also need to set a suitable width increase ratio between adjacent parallel branches of PPNet. The experiments are still conducted based on PPNet M2-D2-W32, and the same four width increase ratios, 2-2, 2-3, 3-2, 3-3, are compared, keeping the depth increase ratio between adjacent branches as 2. The model complexity and performance using different width increase ratios is shown in Fig. 11. The APs, the number

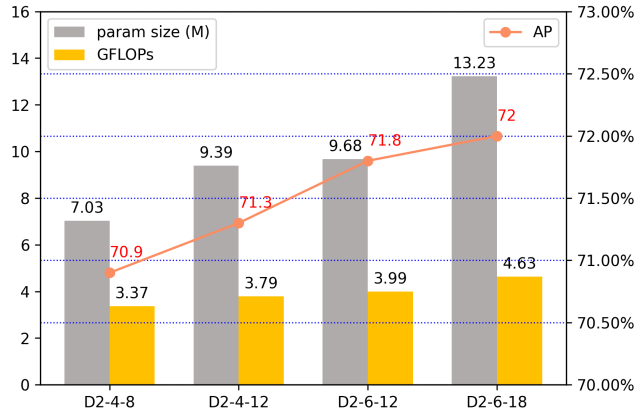


Fig. 10. Comparison of the model complexity and performance using different depth increase ratios. D2-4-8 corresponds to the depth increase ratio 2-2, and means the depth of the three parallel branches are 2, 4, and 8 respectively.

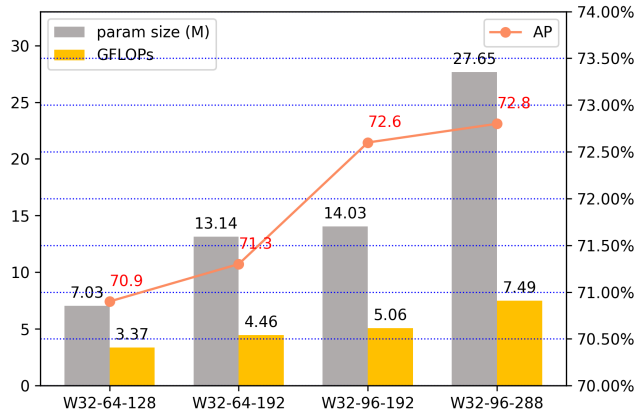


Fig. 11. Comparison of the model complexity and accuracy using different width increase ratios. W32-64-128 corresponds to the width increase ratio 2-2, and means the width of the three parallel branches are 32, 64, and 128 respectively.

of parameters, and GFLOPs all consistently increase with larger width increase ratios. Making the width increase ratio between the first and second branch as 3 other than 2, the improvement of model performance is obvious. Thus, it may be better to set a larger number for the channels of the second branch. However, we still make 2 as the width increase ratio between all adjacent branches because of efficiency, and this choice also follows the common practice used in well-known networks such as VGG [7] and ResNet [18].

The feature resolution decrease ratio. The accuracy and the number of GFLOPs of a model are highly correlated to the resolution of features. Using PPNet M2-D2-W32, we conduct experiments to compare three feature resolution decrease ratios between adjacent branches, 2-2, 2-3, and 3-2, with the input size of 384×288. A larger decrease ratio means lower feature resolution in the next parallel branch of PPNet. For example, if the decrease ratio 3-2 is used, the feature resolutions in the three branches will be 96×72, 32×24, and 16×12 respectively. The model performance and GFLOPs under each feature resolution decrease ratio are given in Table VIII. Using

TABLE VIII
COMPARISON OF MODEL PERFORMANCE AND GFLOPS USING DIFFERENT FEATURE RESOLUTION DECREASE RATIOS BETWEEN ADJACENT PARALLEL BRANCHES OF PPNET. THE RESULTS ARE OBTAINED BASED ON PPNET M2-D2-W32 WITH THE INPUT SIZE 384×288.

Decrease ratio	GFLOPs	AP	AP^{50}	AP^{75}
2-2	7.57	72.3	88.8	79.1
2-3	5.88	71.9	88.6	78.7
3-2	5.20	71.4	88.7	78.5

a lager decrease ratio will slightly lower the mean AP, but the reducing of GFLOPs is considerable. We choose the decrease ratio 2 between all adjacent branches of PPNet, as this is the most used down-sampling factor.

As discussed above, the networks of PPNet can be built flexibly by various choices. For the depth and width increase ratio between two adjacent branches, it is hard to determine the best value, and we just use 2 to follow common practices. There is a high probability other values (not limited to integer) can deliver stronger performance with greater efficiency. We need to consider the depth, the width, and the feature resolution of each parallel branch together, and determine their suitable values simultaneously. Nowadays neural architecture search (NAS) [68] proves high effectiveness in the design of task-specific convolutional networks. Based on the architecture of SimpleBaseline [15] EvoPose2D [69] carries out the search and gets very efficient networks. Because our PPNet demonstrates to be a more elegant architecture and there is much more freedom of search, we believe better configurations of PPNet can be obtained by NAS, outputting very efficient networks.

V. FURTHER ANALYSIS

A. Support for The Motivation

With the purpose of increasing efficiency, the design of PPNet tries to separate semantic and spatial information into low-resolution and high-resolution feature maps respectively. Here, we conduct investigation on the feature pyramids of PPNet, aiming to support the motivation by providing more empirical evidence.

First, we investigate the feature maps of PPNet to see whether the spatial and semantic information is processed separately in different resolutions. To do the verification, the feature maps produced by the last parallel pyramid module of PPNet are visualized. We present both the feature maps of the highest resolution and the lowest resolution in a feature pyramid. The corresponding feature maps generated by HRNet [14] and ResNet (SimpleBaseline [15]) are also visualized for comparison. For the visualization, we use images of size 256×192 as the input, then the sizes of the highest and lowest resolution feature maps are 64×48 and 16×12 respectively. We compare the feature maps from three networks, PPNet M4-D2-W32, HRNet-W32 and ResNet-101, as these three models are of similar size. The visualization of the feature maps is shown in Fig. 12. Comparing the lowest resolution feature maps of the three networks, the features of PPNet look similarly to the other two, all of them give responses mainly

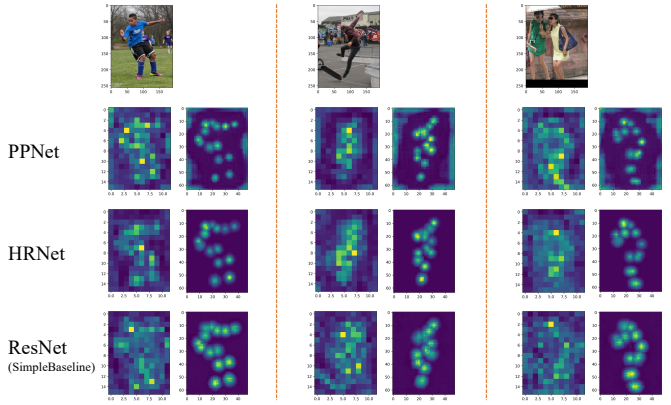


Fig. 12. Visualization of different resolution feature maps in PPNet. The corresponding feature maps in HRNet [14] and ResNet (SimpleBaseline [15]) are also presented for comparison. The feature maps of PPNet are from the last parallel pyramid module of PPNet M4-D2-W32, the feature maps of HRNet are from the last stage of HRNet-W32, and the feature maps of ResNet are from the C4 layer of ResNet-101 and the last deconvolution layer. We show the feature maps after using 1×1 max pooling on the channels.

to the area of the target human body. Because both HRNet and ResNet utilize a high-to-low process to generate low-resolution features, and these high-level representations are considered mainly contain semantic information. We argue that the lowest resolution features of PPNet also mainly include semantic information, though there is not a high-to-low process in PPNet.

From the highest resolution feature maps, the keypoints of interesting on human body can be easily noticed. PPNet and HRNet [14] both get the final high-resolution representation by fusing multi-scale features. SimpleBaseline [15] only uses several deconvolutions to generate the high-resolution representation from the low-resolution features. In Fig. 12, we can see that the highest resolution features of PPNet and HRNet can more precisely indicate the locations of keypoints than those of ResNet. HRNet [14] has explained that its ability to maintain high-resolution makes the representation more spatially precise. PPNet only utilizes shallow sub-networks to process high-resolution features, and the final high-resolution representation is also spatially precise. Hence the spatial information is well preserved in the high-resolution features of PPNet.

B. ImageNet Pre-training

Using ImageNet [18] to pre-train models then fine-tuning on specific down-stream tasks such as object detection [45] and human pose estimation [14] is usually able to get better performance. We conduct experiments to test how the performance of PPNet can be improved by ImageNet pre-training. Two models of PPNet, M2-D2-W32 and M4-D2-W32, are pre-trained on ImageNet, then the COCO keypoint dataset [10] is used for fine-tuning. All other training and testing details are the same as described in Section IV-A. To better demonstrate the capability of PPNet, the results of SimpleBaseline [15] using ImageNet pretraining are given for comparison. We select two models of SimpleBaseline, ResNet-50 and ResNet-101, to do the comparison, because they have similar size to

TABLE IX
THE PERFORMANCE OF PPNET MODELS PRE-TRAINED ON IMAGENET. EXPERIMENTS ARE CONDUCTED ON THE COCO KEYPOINT DATASET, AND THE COCO VALIDATION SET IS USED FOR TESTING. THE RESULTS OF SIMPLEBASELINE MODELS PRE-TRAINED ON IMAGENET ARE ALSO GIVEN FOR COMPARISON. SB = SIMPLEBASELINE.

Method	Backbone	Input size	AP	AP^{50}	AP^{75}
PPNet	M2-D2-W32	256×192	72.1	89.0	79.5
		384×288	73.8	89.7	80.4
	M4-D2-W32	256×192	73.8	89.8	81.1
		384×288	75.7	90.1	82.2
SB [15]	ResNet-50	256×192	70.4	88.6	78.3
		384×288	72.2	89.3	78.9
	ResNet-101	256×192	71.4	89.3	79.3
		384×288	73.6	89.5	80.5

TABLE X
COMPARISON OF INFERENCE SPEED. THE INFERENCE TIME OF PPNET AND SEVERAL OTHER NETWORKS ARE GIVEN. WE REPORT THE TIME A MODEL TAKES TO DO THE INFERENCE OF A BATCH. THE EXPERIMENTS ARE DONE ON PYTORCH 1.0 USING A SINGLE 2080TI GPU, THE BATCH SIZE IS 32. SEC. = SECONDS, SB = SIMPLEBASELINE [15], HG = HOURGLASS [13], HRNET = HRNET [14].

Model	GFLOPs	Input size	Infer time (sec./batch)	AP
PPNet M2-D2-W32	3.3	256×192	0.073	70.9
SB ResNet-50	8.9		0.074	69.9
PPNet M4-D2-W32	5.0		0.080	72.7
HG 2-stage	6.3		0.102	71.7
PPNet M4-D2-W48	20.6	384×288	0.193	75.5
SB ResNet-152	35.4		0.250	73.0
HRNet W48	32.9		0.297	75.0

PPNet M2-D2-W32 and M4-D2-W32 respectively. Table IX shows the results of all compared models. Compared to the results in Table I and Table II, pre-training PPNet models on ImageNet can improve AP by more than 1.0 points, whether the input size is 256×192 or 384×288. Making a comparison between ImageNet pre-trained models, PPNet still outperforms SimpleBaseline by a large margin.

C. Inference Speed

Other than the training speed, inference speed is also important to reflect the efficiency of models. Table X compares the inference time of PPNet with other networks. The experiments are conducted on the COCO validation dataset [10], and we use a single 2080Ti GPU. The batch size is 32 for both the input size 256×192 and 384×288. PPNet takes lower inference runtime cost compared to other networks of similar size. This is consistent with the efficiency reflected by the training speed given in Tables of Section IV.

D. Impact of data processing

In above experiments, the results of PPNet are obtained with the standard data processing [14], [15] for fair comparison with other methods. Recent works Dark [61] and UDP [62] show that unbiased data processing can increase the accuracy of human pose estimation, and these technologies are model agnostic. We conduct experiments to see how the performance

TABLE XI
THE IMPROVEMENT PPNET CAN BE OBTAINED BY USING UNBIASED DATA PROCESSING DARK [61] AND UDP [62]. THE EVALUATION IS CONDUCTED ON THE COCO VALIDATION SET. COMPARING TO THE CORRESPONDING RESULTS IN TABLE I AND II, THE APS ARE IMPROVED BY ABOUT 0.5 TO 1.0 POINTS USING UNBIASED DATA PROCESSING, THE IMPROVEMENT ON THE INPUT SIZE 256×192 CAN BE A LITTLE BIGGER.

Backbone	Input size	Dark	UDP	AP	AP^{50}	AP^{75}
M2-D2-W32	256×192	✓		71.8	88.6	78.9
			✓	72.0	88.8	78.9
	384×288	✓		72.7	88.8	79.2
			✓	72.6	88.7	79.1
M4-D2-W32	256×192	✓		73.4	89.2	80.4
			✓	73.7	89.2	80.4
	384×288	✓		74.3	89.4	80.6
			✓	74.7	89.6	81.1

of PPNet can be improved with these technologies. The models of PPNet M2-D2-W32 and M4-D2-W32 are trained using both Dark [61] and UDP [62] on the COCO keypoint Dataset. Table XI gives the results of using unbiased data processing with the input size of 256×192 and 384×288. For both PPNet models, the accuracy can be improved by about 0.5 to 1.0 AP whether using Dark or UDP. The improvement on the input size 256×192 is a little bigger than 384×288. These results are consistent with the improvement obtained on other backbone networks such as SimpleBaseline [15] and HRNet [14].

VI. CONCLUSION

In this paper, we demonstrate the efficiency of a parallel pyramid network for human pose estimation. The network features separating spatial location preserving from semantic information acquisition in architecture design. With this unique design, superior performance is able to be delivered by using less parameters and GFLOPs comparing to the existing network architectures in the literature. In addition, our proposed PPNet has great freedom to build models of different sizes, meeting various requirements in practical applications. In future work, PPNet will be applied to other dense prediction tasks, such as semantic segmentation, facial landmark detection, and super-resolution. And based on the architecture of PPNet, we will try neural architecture search to get more efficient networks.

ACKNOWLEDGMENT

This work was supported by NSF of China (Grant Nos. 61802189, U1713208, 62036007, 61922066, 61876142, 61973162), NSF of Jiangsu Province (Grant Nos. BK20180464), the Fundamental Research Funds for the Central Universities (Grant Nos. 30919011280, 30920032202, 30921013114), China Postdoctoral Science Foundation (Grant Nos. 2020M681609), and the "Young Elite Scientists Sponsorship Program" by CAST (Grant Nos. 2018QNRC001).

REFERENCES

[1] X. Zheng, X. Chen, and X. Lu, "A joint relationship aware neural network for single-image 3d human pose estimation," *IEEE Transactions on Image Processing*, vol. 29, no. 2, pp. 4747–4758, 2020.

[2] W. Yang, W. Ouyang, X. Wang, J. Ren, H. Li, and X. Wang, "3d human pose estimation in the wild by adversarial learning," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018, pp. 5255–5264.

[3] Q. Ke, M. Bennamoun, H. Rahmani, S. An, F. Sohel, and F. Boussaid, "Learning latent global network for skeleton-based action prediction," *IEEE Transactions on Image Processing*, vol. 29, no. 9, pp. 959–970, 2020.

[4] W. Mao, M. Liu, M. Salzmann, and H. Li, "Learning trajectory dependencies for human motion prediction," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9489–9497.

[5] H. Wang and L. Wang, "Beyond joints: Learning representations from primitive geometries for skeleton-based action recognition and detection," *IEEE Transactions on Image Processing*, vol. 27, no. 9, pp. 4382–4394, 2018.

[6] L. Wang, D. Q. Huynh, and P. Koniusz, "A comparative review of recent kinect-based action recognition algorithms," *IEEE Transactions on Image Processing*, vol. 29, no. 7, pp. 15–28, 2020.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[9] M. Andriluka, L. Pishchulin, P. Gehler, and 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.

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

[11] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

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

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

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

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

[16] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, and 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.

[17] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[18] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE conference on computer vision and pattern recognition*. IEEE, 2009, pp. 248–255.

[19] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*. IEEE, 2005, pp. 886–893.

[20] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1627–1645, 2009.

[21] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele, "Poselet conditioned pictorial structures," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 588–595.

[22] B. Sapp and B. Taskar, "Modec: Multimodal decomposable models for human pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3674–3681.

[23] L. Zhao, X. Gao, D. Tao, and X. Li, "Tracking human pose using max-margin markov models," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5274–5287, 2015.

[24] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial structures for object recognition," *International journal of computer vision*, vol. 61, no. 1, pp. 55–79, 2005.

[25] D. Ramanan, "Learning to parse images of articulated bodies," *Advances in neural information processing systems*, vol. 19, pp. 1129–1136, 2006.

[26] L. Zhao, X. Gao, D. Tao, and X. Li, "Learning a tracking and estimation integrated graphical model for human pose tracking," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 12, pp. 3176–3186, 2015.

[27] Y. Yang and D. Ramanan, "Articulated human detection with flexible mixtures of parts," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 12, pp. 2878–2890, 2012.

[28] L. Pishchulin, A. Jain, M. Andriluka, T. Thormählen, and B. Schiele, "Articulated people detection and pose estimation: Reshaping the future," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 3178–3185.

[29] L. Zhao, X. Gao, D. Tao, and X. Li, "A deep structure for human pose estimation," *Signal Processing*, vol. 108, pp. 36–45, 2015.

[30] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*. IEEE, 2015, pp. 3431–3440.

[31] A. Toshev and 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.

[32] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler, "Joint training of a convolutional network and a graphical model for human pose estimation," in *Proceedings of the Advances in neural information processing systems*, 2014, pp. 1799–1807.

[33] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, "Efficient object localization using convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 648–656.

[34] X. Chen and A. L. Yuille, "Articulated pose estimation by a graphical model with image dependent pairwise relations," in *Proceedings of the Advances in neural information processing systems*, 2014, pp. 1736–1744.

[35] X. Fan, K. Zheng, Y. Lin, and S. Wang, "Combining local appearance and holistic view: Dual-source deep neural networks for human pose estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1347–1355.

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

[37] G. Moon, J. Y. Chang, and K. M. Lee, "Posefix: Model-agnostic general human pose refinement network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7773–7781.

[38] L. Zhao, J. Xu, S. Zhang, C. Gong, J. Yang, and X. Gao, "Perceiving heavily occluded human poses by assigning unbiased score," *Information Sciences*, vol. 537, pp. 284–301, 2020.

[39] L. Zhao, J. Xu, C. Gong, J. Yang, W. Zuo, and X. Gao, "Learning to acquire the quality of human pose estimation," *IEEE Transactions on Circuits and Systems for Video Technology TCSVT.2020.3005522*, 2020.

[40] J. Wang, X. Long, Y. Gao, E. Ding, and S. Wen, "Graph-pcnn: Two stage human pose estimation with graph pose refinement," *arXiv preprint arXiv:2007.10599*, 2020.

[41] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. V. Gehler, and B. Schiele, "Deepcut: Joint subset partition and labeling for multi person pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4929–4937.

[42] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7291–7299.

[43] A. Newell, Z. Huang, and J. Deng, "Associative embedding: End-to-end learning for joint detection and grouping," in *Advances in Neural Information Processing Systems*, 2017, pp. 2277–2287.

[44] B. Cheng, B. Xiao, J. Wang, H. Shi, T. S. Huang, and L. Zhang, "Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5386–5395.

[45] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[46] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[47] X. Chu, W. Yang, W. Ouyang, C. Ma, A. L. Yuille, and X. Wang, "Multi-context attention for human pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1831–1840.

[48] W. Yang, S. Li, W. Ouyang, H. Li, and X. Wang, "Learning feature pyramids for human pose estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1281–1290.

[49] Y. Chen, C. Shen, X.-S. Wei, L. Liu, and J. Yang, "Adversarial posenet: A structure-aware convolutional network for human pose estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1212–1221.

[50] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[51] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang *et al.*, "Deep high-resolution representation learning for visual recognition," *IEEE transactions on pattern analysis and machine intelligence 10.1109/TPAMI.2020.2983686*, 2020.

[52] C. Jiang, K. Huang, S. Zhang, X. Wang, and J. Xiao, "Pay attention selectively and comprehensively: Pyramid gating network for human pose estimation without pre-training," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 2364–2371.

[53] J. Wang, Z. Wei, T. Zhang, and W. Zeng, "Deeply-fused nets," *arXiv preprint arXiv:1605.07716*, 2016.

[54] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2881–2890.

[55] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[56] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[57] X. Peng, R. S. Feris, X. Wang, and D. N. Metaxas, "A recurrent encoder-decoder network for sequential face alignment," in *European conference on computer vision*. Springer, 2016, pp. 38–56.

[58] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu, "Rmpe: Regional multi-person pose estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2334–2343.

[59] P. Zhang, C. Lan, W. Zeng, J. Xing, J. Xue, and N. Zheng, "Semantics-guided neural networks for efficient skeleton-based human action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1112–1121.

[60] J. Wu, H. Zheng, B. Zhao, Y. Li, B. Yan, R. Liang, W. Wang, S. Zhou, G. Lin, Y. Fu *et al.*, "Ai challenger: A large-scale dataset for going deeper in image understanding," *arXiv preprint arXiv:1711.06475*, 2017.

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

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

[63] K. Su, D. Yu, Z. Xu, X. Geng, and C. Wang, "Multi-person pose estimation with enhanced channel-wise and spatial information," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5674–5682.

[64] Y. Bin, X. Cao, X. Chen, Y. Ge, Y. Tai, C. Wang, J. Li, F. Huang, C. Gao, and N. Sang, "Adversarial semantic data augmentation for human pose estimation," in *European Conference on Computer Vision*. Springer, 2020, pp. 606–622.

[65] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[66] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural networks for machine learning*, vol. 4, no. 2, pp. 26–31, 2012.

[67] G. Papandreou, T. Zhu, N. Kanazawa, A. Toshev, J. Tompson, C. Bregler, and K. Murphy, "Towards accurate multi-person pose estimation in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4903–4911.

[68] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning*, 2019, pp. 6105–6114.
[69] W. McNally, K. Vats, A. Wong, and J. McPhee, "Evopose2d: Pushing the boundaries of 2d human pose estimation using neuroevolution," *arXiv preprint arXiv:2011.08446*, 2020.

Lin Zhao received his B.E. degree from Xidian University in 2010, and the PhD degree in pattern recognition and intelligent systems form Xidian University in 2017. He is currently a lecture with the School of Computer Science and Engineering, Nanjing University of Science and Technology. His research interests include human related computer vision tasks, such as human pose estimation and tracking, 3D human shape reconstruction, and abnormal human behaviour detection.

Nannan Wang (M'16) received the B.Sc. degree in information and computation science from the Xi'an University of Posts and Telecommunications in 2009 and the Ph.D. degree in information and telecommunications engineering from Xidian University in 2015. From September 2011 to September 2013, he was a Visiting Ph.D. Student with the University of Technology, Sydney, NSW, Australia. He is currently a Professor with the State Key Laboratory of Integrated Services Networks, Xidian University. He has published over 100 articles in refereed journals and proceedings, including IEEE T-PAMI, IJCV, NeurIPS etc. His current research interests include computer vision, pattern recognition, and machine learning.

Chen Gong received his B.E. degree from East China University of Science and Technology (ECUST) in 2010, and dual doctoral degree from Shanghai Jiao Tong University (SJTU) and University of Technology Sydney (UTS) in 2016 and 2017, respectively. Currently, he is a full professor in the School of Computer Science and Engineering, Nanjing University of Science and Technology. His research interests mainly include machine learning, data mining, and learning-based vision problems. He has published more than 90 technical papers at prominent journals and conferences such as IEEE T-PAMI, IEEE T-NNLS, IEEE T-IP, ICML, NeurIPS, CVPR, AAAI, IJCAI, etc. He also serves as the reviewer for more than 20 international journals such as AIJ, IEEE T-PAMI, IEEE T-NNLS, IEEE T-IP, and also the SPC/PC member of several top-tier conferences such as ICML, NeurIPS, CVPR, AAAI, IJCAI, ICDM, AISTATS, etc. He received the "Excellent Doctorial Dissertation" awarded by Shanghai Jiao Tong University (SJTU) and Chinese Association for Artificial Intelligence (CAAI). He was enrolled by the "Young Elite Scientists Sponsorship Program" of Jiangsu Province and China Association for Science and Technology. He was also the recipient of "Wu Wen-Jun AI Excellent Youth Scholar Award".

Jian Yang received the PhD degree from Nanjing University of Science and Technology (NUST), on the subject of pattern recognition and intelligence systems in 2002. In 2003, he was a Postdoctoral researcher at the University of Zaragoza. From 2004 to 2006, he was a Postdoctoral Fellow at Biometrics Centre of Hong Kong Polytechnic University. From 2006 to 2007, he was a Postdoctoral Fellow at Department of Computer Science of New Jersey Institute of Technology. Now, he is a Chang-Jiang professor in the School of Computer Science and Technology of NUST. He is the author of more than 200 scientific papers in pattern recognition and computer vision. His papers have been cited more than 6000 times in the Web of Science, and 15000 times in the Scholar Google. His research interests include pattern recognition, computer vision and machine learning. Currently, he is/was an associate editor of Pattern Recognition, Pattern Recognition Letters, IEEE Trans. Neural Networks and Learning Systems, and Neurocomputing. He is a Fellow of IAPR.

Xinbo Gao (M'02-SM'07) received the B.Eng., M.Sc., and Ph.D. degrees in signal and information processing from Xidian University, Xi'an, China, in 1994, 1997, and 1999, respectively. From 1997 to 1998, he was a research fellow at the Department of Computer Science, Shizuoka University, Shizuoka, Japan. From 2000 to 2001, he was a post-doctoral research fellow at the Department of Information Engineering, the Chinese University of Hong Kong, Hong Kong. Since 2001, he has been at the School of Electronic Engineering, Xidian University. He is currently a Cheung Kong Professor of Ministry of Education of P. R. China, a Professor of Pattern Recognition and Intelligent System, and the President of Chongqing University of Posts and Telecommunications. His current research interests include Image processing, computer vision, multimedia analysis, machine learning and pattern recognition. He has published six books and around 300 technical articles in refereed journals and proceedings. Prof. Gao is on the Editorial Boards of several journals, including Signal Processing (Elsevier) and Neurocomputing (Elsevier). He served as the General Chair/Co-Chair, Program Committee Chair/Co-Chair, or PC Member for around 30 major international conferences. He is a Fellow of the Institute of Engineering and Technology and a Fellow of the Chinese Institute of Electronics.