2021 14th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI) | 978-1-6654-0004-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/CISP-BMEI53629.2021.9624376

SLBNet: Shallow and Lightweight Bilateral Network for Pose Estimation

Maolin Zhou[1,2], Wanhu Sun[1,2], Fan Yang[1,2], and Sheng Zhang[1,2]
[1]Shenzhen International Graduate School, Tsinghua University
[2]Department of Electronic Engineering, Tsinghua University
Shenzhen 518000, China

***Abstract*—Human pose estimation from images is an important task in many real-life applications. However, most existing methods focus on improving the effectiveness without considering efficiency, making the networks computationally expensive with a huge size. As depthwise separable convolution can help compress the model size and floating point operations (FLOPs), some methods combined it to make human pose estimation affordable on resource-constrained devices. However, depthwise separable convolution also slows down the inference speed, especially on GPU devices. In this paper, we introduce a shallow and lightweight bilateral network (SLBNet). Our network inferences much faster than the existing methods while achieves competitive performance. We evaluate our networks on the MPII and COCO datasets. Specially, our SLBNet yields 67.8 Average Precision (AP) on COCO test set with only 3.6M parameters and 4.5G FLOPs at 253 FPS on a single 2080Ti GPU, and 25 FPS on an Intel i7-8700K CPU machine.**

I. Introduction

Aiming to localize human body skeletal keypoints in given RGB image, human pose estimation (HPE) has been a fundamental but important task in computer vision. Benefiting from the rapid development of deep learning technologies, convolutional neural networks (CNN) based methods [24], [17], [7], [13], [5], [21] greatly outperform the traditional methods [6], [28] in HPE. Among these prior works, HRNet [21] achieved state-of-the-art results by maintaining high-resolution representations with a multi-branch framework. However, the above methods constructed heavy and complex architectures to pursue high accuracy, hence were hard to be deployed on the resource-limited computing devices commonly used in real-life scenarios. SimpleBaseline [27] utilized a simple network to achieve high accuracy and show a fast inference speed on GPU devices, while the amount of the parameters of the model was too high to afford.

To design a lightweight network for the resource-limited computing devices, Bulat *et al.* [3] binarized the network architecture to compress the model size and improve the efficiency, while the performance dropped significantly. Some methods such as LPN [32], EfficientPose [31], DANet [15] used depthwise separable convolution to reduce the parameters and FLOPs of the model, which achieved competitive results on CPU edge-devices with a low inference latency. However, depthwise separable convolutions require much more input/output reads than ordinary convolutions under the same FLOPs, which cannot fully utilize the GPU capacity, leading to the depthwise separable convolutions based methods inferencing slow on GPU devices. And with the development of the edge-devices, GPU resources become able to be deployed on it. Current applications also begin to require more utilization of GPU resources to increase the efficiency of HPE.

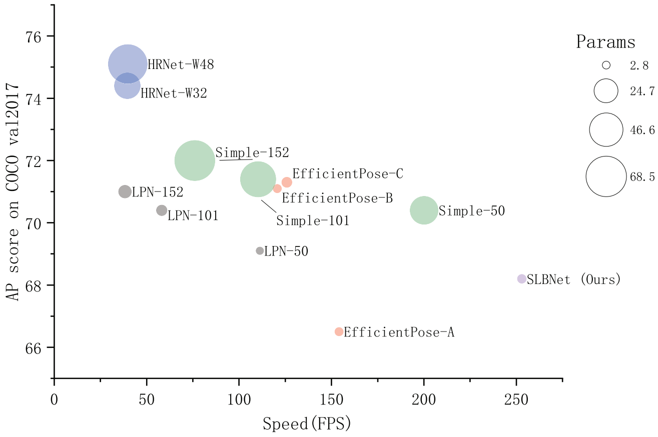


Fig. 1. Measurement of AP score, speed and params of the network architecture referred in Table I on a GPU platform. The area of a circle represents the scale of the params of the corresponding method.

HRNet [21] also achieved excellent results in semantic segmentation tasks but was too heavy to realize real-time because of its deep and complex network architecture. Motivated by the multi-resolution design principle in HRNet and the bilateral network such as Fast-SCNN [19] and BiSeNet [29], DDRNet [11] achieved new state-of-the-art trade-off between accuracy and speed on semantic segmentation tasks.

Inspired by the above methods performing well in computer vision, we propose a shallow and lightweight bilateral network (SLBNet) to realize effective pose estimation on resources-limited devices. SLBNet complete HPE with straightforward and concise structure instead of stacking deep layers to repeatedly downsampling and upsampling feature maps.

SLBNet starts with one trunk to downsample the feature maps to 1/4 resolution of the input image, then splits the networks into two shallow stems with different resolutions. We keep the resolution invariant after the split and enable the information flow between stems. Then the two stems will be concatenated as input to our decoder module, which consists of a pyramid pooling module (PPM) [33] combined with an

978-1-6654-0004-6/21/$31.00 ©2021 IEEE

attention mechanism [12].

We empirically show a good trade-off between the effectiveness and efficiency of our methods over two benchmark datasets: the COCO keypoint detection dataset [14] and the MPII human pose dataset [1]. We achieve competitive performance compared to the state-of-the-art methods with much smaller complexity and faster inference speed on GPU. In summary, our contributions are as follow:

- We propose a efficient bilateral pose estimation network framework SLBNet, consisting of two shallow stems of different resolutions without repeatedly downsampling. Our network attains new state-of-the-art inference speed without any extra bells or whistles.
- Experiments show that our network achieves a better trade-off between effectiveness and efficiency than the existing methods. Our SLBNet can achieve 67.8 in AP score on COCO test-dev set with only 3.65M parameters and 4.45G FLOPs.
- Our SLBNet is GPU-friendly. We achieve a 25% speedup compared to the state-of-the-art simplebaseline-R50 with competitive accuracy and much small model size (10.6%), while the existing lightweight methods inference slow on GPU devices.

II. Related works

A. Human pose estimation

Early human pose estimation works depended on hand-crafted features to detect joints and consequently output pose information after complicated post-processing. In recent years, the development of CNNs greatly promoted the performance of works in this area. Instead of directly outputting the pose keypoints, most of works recently predicted spatial intermediate representation, followed with an argmax operation to locate the keypoints [17], [25]. Newell *et al.* proposed a Stacked Hourglass [17] architecture consisting of continuous encoder-decoder Fully Convolutional Networks, which is still followed by part of current state-of-the-art single person pose estimation works. Chen *et al.* proposed a Cascade Pyramid Network (CPN) [7] to tackle pose estimation with a coarse-to-fine way. [27] achieved good performance on COCO with a simple and fast network SimpleBaseLine. [21] proposed a multi-branch network fusing features with different resolutions, achieve state-of-the-art results on COCO. RSN [4] combines attention mechanism and aggregates features with the same spatial size to win COCO Keypoint Challenge 2019.

Prior works mainly focused on improving the accuracy of HPE, with less consideration of the complexity and latency of the networks, leading to the limits of the applications on the weak-computation devices. For applications in real-life scenarios, some current works focused on proposing a light pose estimation method to tackle the problem.

B. Light pose estimation

To alleviate the heavy computation cost and high latency of methods like HRNet, FPD [30] used a teacher-student network architecture to compress the model. SimpleBaseLine [27] with ResNet [10] made a fast pose estimation on GPU platform, however, bringing with lots of parameters and computation. To improve the efficiency of pose estimation, LPN [32] introduced a light block by replacing standard convolution with depthwise separable convolution, which can significantly compress the amount of the model parameters. EfficientPose [31] proposed a encoder-decoder networks with neural architecture search (NAS) adjusted and get a better trade-off between model size and efficiency. Depthwise separable convolution was widely used in LPN and EfficientPose, as it provided significant performance benefits owing to the reduction in both parameters and multi-adds. However, depthwise separable convolution layers on GPU were slow currently due to their computation way mismatched with the GPU architecture and cannot fully utilize the GPU resources, leading to slow training and inference speed.

C. Multi-resolution fusion

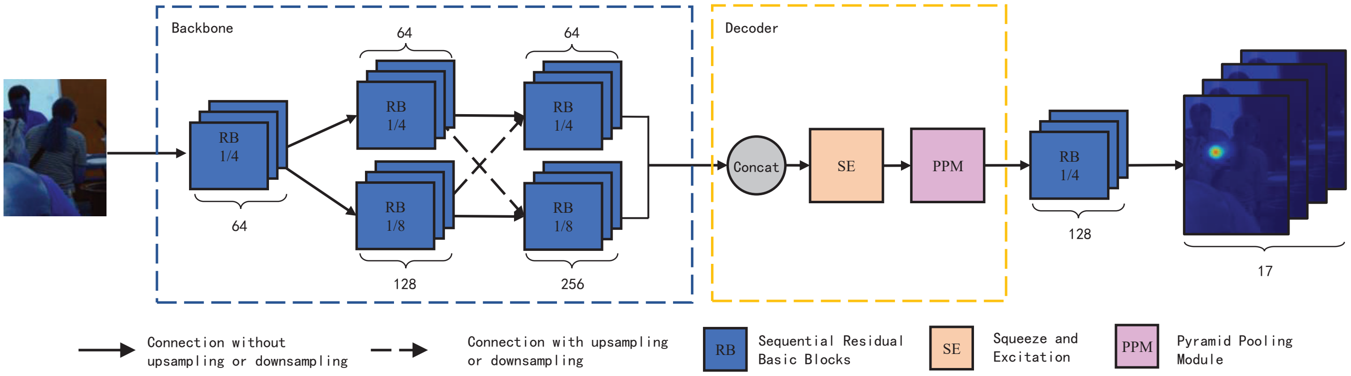
Multi-resolution fusion has been a hot topic in vision-based tasks. Methods like [27], [17], [16], [20] consisted of a high-to-low encoder and a low-to-high decoder. The high-to-low encoder was used to extract context information by repeatedly reducing resolution of the feature maps. And the low-to-high decoder was used to recover the high-resolution representations from the low-resolution representations by upsampling and skip connection. As HRNet pointed out, high-resolution representations were very important for dense vision predicting tasks. HRNet gradually added high-resolution sub-networks to the main structure and connected the multi-resolution sub-networks in parallel, ensuring information flow fluently across different resolutions. Unipose [2] connected the backbone with a multi-resolution context information aggregation module WASP. Zhao *et al.* [33] proposed PPM to efficiently capture abundant context information through pyramid pooing.

To tackle real-time semantic segmentation tasks, BiSeNet [29] proposed a two-stem architecture. It extracted semantic information by a relatively deep sub-network in a stem and maintained rich high-resolution information in another stem. Based on BiSeNet, DDRNet [11] merged first several stages and added a bilateral path to make information flow between the two stems. Inspired by the multi-resolution works, we try to design a lightweight network fully utilizing relatively-high resolution representations in our work.

III. Method

Human pose estimation tasks require localization of the specific K person keypoints for an RGB image input. We adopt a top-down paradigm: First, we use a detector to locate the persons in the given image, then crop them out and resize to a fixed size respectively and estimate the single-person pose.

Our proposed SLBNet is presented in Figure 2. We adopt the widely-used keypoints heatmap pipeline in our method. The value of the pixel on the heatmap indicates the relative possibility. After the network outputs the $K \times H' \times W'$ heatmaps, we use an argmax post-processing to get the final



Backbone
RB 1/4
64
64
RB 1/4
RB 1/8
128
64
RB 1/4
RB 1/8
256
Decoder
Concat
SE
PPM
RB 1/4
128
17
Connection without upsampling or downsampling
Connection with upsampling or downsampling
RB Sequential Residual Basic Blocks
SE Squeeze and Excitation
PPM Pyramid Pooling Module

Fig. 2. Illustrating the architecture of the presented SLBNet. Backbone of our SLBNet consists of two shallow stems split from one trunk. Feature maps are concatenated to decoder to output the human keypoint heatmaps. See Section II for details.

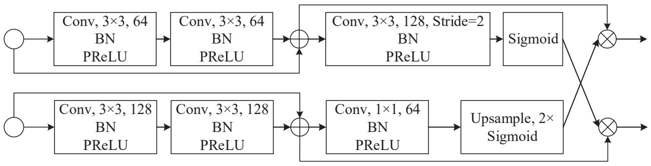


Fig. 3. In the backbone, we extract dual resolution feature information in parallel. The resolution of every single stem is kept invariant. To match the number of channels in Figure 2, consecutive convolutional layers are added, which are not drawn in this figure.

pose estimation results, where W' and H' correspond to the heatmap width and height, respectively. To supervise the network training, $K \times H' \times W'$ target heatmaps are generated in a 2D Gaussian form centered on the ground-truth location.

A. Motivation

SLBNet mainly benefits from the relatively high-resolution representations and two-stem backbone. When we locate human pose by our eyes in our daily life, we tend to rely on high-resolution information in perceptual. Following HRNet [21], we consider the high-resolution representations to be necessary for the position-sensitive computer vision tasks, especially for the top-down paradigm human pose estimation.

A popular kind of method tends to construct a high-to-low bilateral network to realize real-time semantic segmentation, such as BiSeNet [29] and DDRNet [11]. Among these methods, high-resolution feature maps are split into two stems: One maintains a high resolution (usually 1/8 of the input size), and the other one repeatedly downsamples to extract context information. Finally an upsampling is performed to recover the segmentation map with origin resolution. The two-stem structure can utilize the multi-resolution feature information effectively, is also adopted in our design.

A lightweight HPE network architecture is further considered. We notice that deep layers extracting rich context information from the low-resolution representations with a large receptive field are less critical, since all human body occupies the most area of the cropped image in the top-down paradigm. Furthermore, too deep network also brings high computation burden and numerous parameters. Hence we pay less attention to the low-resolution representations in our network design. Specifically, our SLBNet consists of two stems with relatively high resolutions and makes information flow across them. By this means, HPE can be realized with a shallower structure and fewer parameters.

As illustrated in Figure 2, the input image is firstly fed into consecutive 3×3 convolutional layers to get the initial feature maps of $1/4$ resolution. Then the feature maps is input to our dual-resolution module, which maintains two stems with their resolution respectively, to extract multi-resolution information. Finally, the feature maps are concatenated and fed into a decoder module to get final pose heatmaps with size of $K \times 1/4H \times 1/4W$ [1], where H and W correspond to the height and width of the input image, respectively.

We use Figure 4 to describe our difference between our network and the networks that inspire us.

B. Dual-resolution backbone

We use a bilateral structure to extract dual-resolution feature maps as illustrated in Figure 3, which contains two basic residual blocks in parallel. Our structure keeps the resolution invariant rather than repeatedly downsampling to extract low-resolution context information, which is conducted with a sequential deep layers. Feature maps of two different resolutions are fed into the dual-resolution module, and then feature maps with the same resolution are output.

As we predict human keypoints heatmap with $1/4$ size of the input, we maintain the exact resolution in one stem until the final output. We choose $1/4$ and $1/8$ to be the dual resolutions individually in our bilateral structure to extract feature maps in parallel. Finally the $1/8$ stem is upsampled to concatenate with the $1/4$ stem at the final output stage.

To ensure information can flow fluently across the two stems, we use a 3×3 convolution with stride two and batch normalization (BN) to downsample the feature maps of relatively high resolution, and then feed them into a sigmoid function to multiply with the feature maps of relatively low

[1] The scaling ratio is set to 1/4 according to [17].

resolution. In another stem, an 1×1 convolution and BN are used to adjust the number of channels, then the feature maps of relatively low resolution are upsampled followed by a sigmoid function to multiply with the feature maps of relatively high resolution.

C. Decoder module

A decoder module as shown in Figure 2 is added to get the final output with the size of $K \times 1/4H \times 1/4W$. Feature maps from the two stems of different resolutions are concatenated as input to a Squeeze and Excitation (SE) [12] block first to adjust the weight of different channels within the concatenated feature maps. As mentioned above, we pay less attention to low-resolution representations in our network design. Compared with other multi-resolution networks, we delete the low-resolution feature maps which contain rich global context information in the backbone. As compensation, a PPM [33] module is followed to extract global context information, which uses pyramid pooling of four different rates to get a large perceptional field to collect information of different levels. Followed by the PPM module, three consecutive convolutional layers are added to get the final keypoint heatmaps.

IV. Experiments

In this section, we describe the implementation details of our work on COCO [14] and MPII [1] dataset and compare the results with the state-of-the-art methods. Ablation experiments are performed to demonstrate the components' benefits in our network design.

A. COCO keypoint detection

1) Dataset: The COCO dataset contains over 200K images and 250K person instances labeled with 17 keypoints. We train our SLBNet on the train2017 set, including 57K images and 150K person instances. We evaluate our method on the val2017 set and test-dev2017 set, containing 5K images and 20K images, respectively. Object Keypoint Similarity (OKS) based AP metric is used to evaluate the accuracy of the keypoint prediction.

2) Training: Following SimpleBaseline [27], we first extend the human detect box to 4:3 in terms of aspect ratio and then resize to a fixed size 256×192. To augment the training data, we use rotation ($\pm 40°$), flipping and random scale ($\pm 30°$) randomly. We use Adam as our optimizer with a batch size of 32 on four 2080Ti GPUs. The initial learning rate is set to 1e-3 and reduced by a factor of 10 at the 90th and 120th epoch. The training process of each stage is terminated at the 150th epoch.

3) Testing: During testing, the two-stage top-down paradigm is used. We use the same detecting results provided by SimpleBaseline [27] for both validation set and test-dev set, which achieve 56.4 in AP score on COCO val2017 set. Following the strategy in Hourglass [17], images and their flipped versions are both evaluated, then the two predicted heatmaps are averaged and blurred by a Gaussian filter. Finally, a quarter offset in the direction of the second-highest response is adjusting rather than directly extracting the location of the highest response in heatmaps.

4) Measurement of inference speed: We measure the inference speed of the network without considering the post-processing of the keypoint heatmaps. To measure the performance on a GPU device, we experiment on a single GTX 2080Ti GPU by setting the batch size to 1 and with CUDA 10.2, CUDNN 7.6.5 and PyTorch 1.7, Intel (R) Core (TM) i7-8700K CPU. We run the same network 10000 times under input resolution 256×192 for COCO, and report the average time to avoid the occasional error. We also measure the CPU-only environment with the same setting to contrast with the lightweight method. Results are shown in Table I.

5) Results on the validation set: We report the results of our method and compare them with the state-of-the-art methods in Table I on the COCO validation dataset. Input sizes are all 256×192. Our SLBNet achieves 68.2 in AP score, only with 3.6M parameters and 4.5G FLOPs. Moreover, SLBNet reaches the fastest network inference speed on GPU (253.0 FPS) as far as we know, and a high inference speed on CPU (24.6 FPS). In detail: (i) Our network outperforms Hourglass [17] with a much smaller model size (14.1%), and we also have a similar AP performance compared to the heavy CPN [7] model. (ii) Compared with the well-performed and fast simpleBaseline-50 [27], our network achieves 26.4% speedup on GPU and 18.3% speed up on CPU, while the number of parameters of our method is only 10.6%. Compared to the state-of-the-art HRNet [21], our network achieves 540.5% speedup on GPU and 121.6% speedup on CPU, while the number of parameters is only 15.8%. (iii) Compared with the lightweight methods [32], [31], our method is significantly faster on GPU, and the inference speed on CPU surpasses most other methods except LPN-50.

6) Results on the COCO test set: We report the pose estimation performance of the existing state-of-the-art methods and our method in Table II. Our network achieves a better trade-off between efficiency and effectiveness than other state-of-the-art methods.

B. MPII human pose estimation

1) Dataset: The MPII dataset contains approximately 25K images with about 40K annotated persons, including 12K annotated persons for testing. We follow the same standard training settings as MSCOCO, except that all input images are cropped to 256×256 for fair comparisons. The results are evaluated by Percentage of Corrected Keypoints (PCK) of the head diameter, which considers a keypoint prediction correct when it lies within the head diameter length of ratio τ, we report the PCKh@0.5 (τ=0.5) score.

2) Testing: Following the procedure in COCO, we also use a top-down paradigm to test on MPII, however we use the provided boxes rather than a human detector to locate persons in image. We use a single-scale testing strategy rather than the multi-scale testing used in HRNet [21]. We also compare with

TABLE I
COMPARISON OF RESULTS ON COCO VALIDATION SET. PARAMS, FLOPS AND FPS ARE ONLY CALCULATED FOR THE POSE ESTIMATION NETWORK. INPUT SIZE ARE ALL 256×192.

Method	Backbone	Params	FLOPs	FPS (GPU)	FPS (CPU)	AP	AR
CPN [7]	ResNet-50	27.0M	6.2G	-	-	68.6	-
8-stage Hourglass [17]	Hourglass	25.6M	26.2G	23.4	3.7	66.9	-
SimpleBaseline [27]	ResNet-50	34.0M	8.9G	200.1	20.8	70.4	76.3
SimpleBaseline [27]	ResNet-101	53.0M	12.4G	110.3	14.0	71.4	77.1
SimpleBaseline [27]	ResNet-152	68.6M	15.7G	76.0	10.8	72.0	77.8
HRNet-W32 [21]	HRNet-W32	28.5M	7.1G	39.5	11.2	74.4	79.8
HRNet-W48 [21]	HRNet-W48	63.6M	14.6G	39.8	7.3	75.1	80.4
EfficientPose-A [31]	NAS searched	3.3M	1.1G	154.1	55.9	66.5	-
EfficientPose-B [31]	NAS searched	3.3M	1.1G	127.7	36.8	71.1	-
EfficientPose-C [31]	NAS searched	5.0M	1.6G	125.9	35.8	71.3	-
LPN [32]	ResNet-50	2.9M	1.0G	111.3	25.5	69.1	74.9
LPN [32]	ResNet-101	5.3M	1.4G	58.1	18.7	70.4	76.2
LPN [32]	ResNet-152	7.4M	1.8G	38.3	14.7	71.0	76.8
Ours	DHRNet	3.6M	4.5G	**253.0**	24.6	68.2	74.3

TABLE II
COMPARISON OF RESULTS ON COCO TEST SET. PARAMS, FLOPS AND FPS ARE ONLY CALCULATED FOR THE POSE ESTIMATION NETWORK.

Method	Backbone	Input size	Params	FLOPs	FPS (GPU)	FPS (CPU)	AP	AR
Mask-RCNN [9]	ResNet-50-FPN	-	-	-	-	-	63.1	-
CPN [7]	ResNet-Inception	384 × 288	-	-	-	-	72.1	78.5
G-RMI [18]	ResNet-101	353 × 257	42.6M	57.0G	-	-	64.9	69.7
Integral Regression [22]	ResNet-101	256 × 256	45.0M	11.0G	-	-	67.8	-
RMPE [8]	PyraNet	320 × 256	28.1M	26.7G	-	-	72.6	-
SimpleBaseline [27]	ResNet-50	256 × 192	34.0M	8.9G	200.1	20.8	70.0	75.6
SimpleBaseline [27]	ResNet-152	256 × 192	68.6M	15.7G	76.0	10.8	71.6	77.3
HRNet-W32 [21]	HRNet-W32	384 × 288	28.5M	16.0G	39.5	11.2	74.9	80.1
HRNet-W48 [21]	HRNet-W48	384 × 288	63.6M	32.9G	39.8	7.3	75.5	80.5
EfficientPose-B [31]	NAS searched	256 × 192	3.3M	1.1G	127.7	36.8	70.5	76.1
EfficientPose-C [31]	NAS searched	256 × 192	5.0M	1.6G	125.9	35.8	70.9	76.5
LPN [32]	ResNet-50	256 × 192	2.9M	1.0G	111.3	25.5	68.7	74.5
LPN [32]	ResNet-101	256 × 192	5.3M	1.4G	58.1	18.7	70.0	75.7
LPN [32]	ResNet-152	256 × 192	7.4M	1.8G	38.3	14.7	70.4	76.2
Ours	SLBNet	256 × 192	3.6M	4.5G	**253.0**	24.6	67.8	73.3

the single-scale testing results of the state-of-the-art methods for fair comparisons.

3) Results on the test set: We show our results in Table V on MPII validation dataset and compare with the state-of-the-art methods. On the one hand, our method achieve competitive results with far fewer parameters and FLOPs, for example we use 10.6% parameters and 49.1% FLOPs of the SimpleBaseline-R50[27] with only 0.4 drop on PCKH@0.5 metric. On the other hand, the lightweight methods[31] with fewer parameters and FLOPs behave a high latency during GPU inference with near accuracy.

C. Ablation experiments

In this subsection, we perform ablation experiments to demonstrate the effectiveness of the components in our network, respectively, as shown in Table III. In our dual-resolution backbone, experiment 0 downsamples the feature maps to 1/16 in the stem with relatively low resolution, while experiment 1 does not. We show that we do not need to repeatedly downsample the resolution of the context stem to extract detailed context information as described in BiSeNet [29].

The comparison between the experiments 1 and 2 shows that sigmoid connection brings a slight increase in AP score and inference speed. The comparison between the experiments 2 and 3 shows PReLU can improve the accuracy but slow down the inference speed.

1) Decoder module: We test the decoder with 3 types: A: SE [12] + DAPPM [11]. B: Direct output. C: SE + PPM [33]. To compare the performance of the different decoder types, we continue the experiments 4 and 5. C decoder achieves a better trade-off between the effectiveness and the efficiency by using a simple PPM module. We finally adopt the C decoder.

2) Compare with different backbones: The structures of the backbones are shown in Figure 4. At first, we use the DDRNet-23-Slim [11] to experiment on the COCO and upsample 2× the output of the network. DDRNet-23-Slim shows competitive inference speed but performs poorly on the COCO validation set. We also test with the BiSeNet [29] but

TABLE III
ABLATION STUDY OF COMPONENTS OF THE NETWORK ON COCO VALIDATION SET.

Num	Bilateral connection	Activation	Decoder	Downsample	Params	FLOPs	FPS (GPU)	AP
0	Plus	ReLU	A	Yes	4.2M	5.0G	242.2	66.5
1	Plus	ReLU	A	No	4.2M	6.1G	241.9	67.9
2	Sigmoid	ReLU	A	No	4.2M	6.1G	245.7	68.0
3	Sigmoid	PReLU	A	No	4.2M	6.1G	223.9	68.5
4	Sigmoid	PReLU	B	No	3.4M	4.3G	286.4	66.5
5	Sigmoid	PReLU	C	No	3.6M	4.4G	253.0	68.2

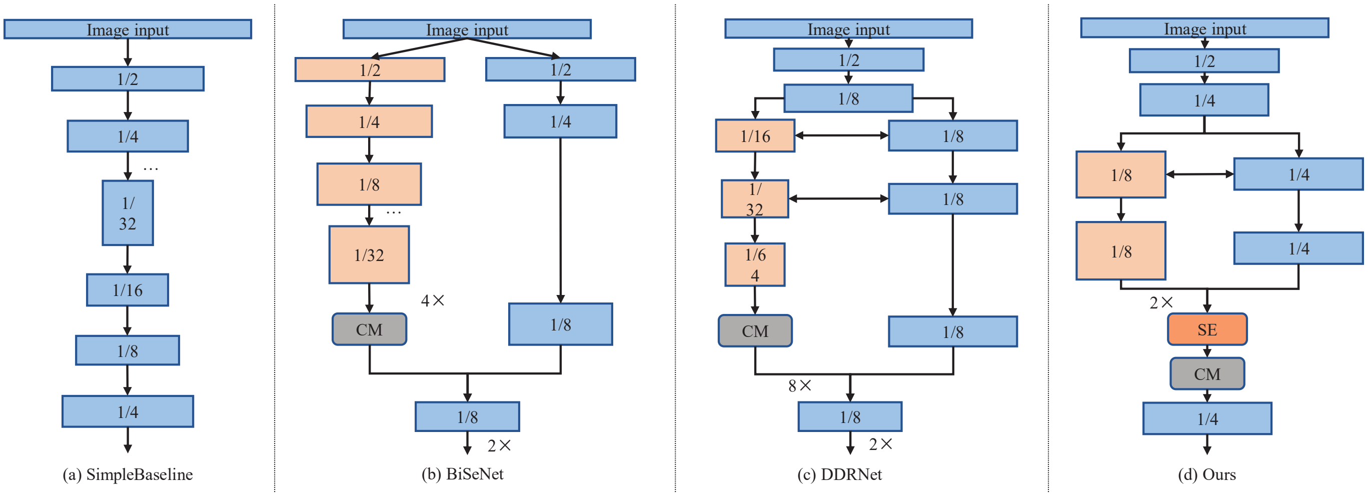


Fig. 4. Comparison with other methods for vision tasks

TABLE IV
COMPARE WITH DIFFERENT BACKBONE ON COCO VALIDATION SET.

Backbone	Params	FLOPs	FPS	AP
DDRNet [11]	5.7M	0.8G	205.9	61.5
SLBNet	3.6M	4.4G	253.0	68.2

TABLE V
RESULTS AND COMPARISON WITH OTHER METHODS ON MPII VALIDATION DATASET.

Method	Params	FPS (GPU)	PCKh@0.5
CPMs [26]	31.0M	-	88.0
DLCM [23]	15.5M	-	87.5
Hourglass [17]	25.1M	23.7	89.2
SimpleBaseline-R50 [27]	34.0M	196.1	88.5
SimpleBaseline-R101 [27]	52.0M	105.7	89.1
HRNet-W32 [21]	28.5M	38.2	90.3
EfficientPose-A [31]	1.3M	157.9	88.1
EfficientPose-B [31]	3.3M	125.6	89.3
Ours	3.6M	**250.7**	88.1

do not get a convergent output. SLBNet shows significantly better results than the semantic segmentation backbones. We compare different backbones in Table IV.

V. CONCLUSIONS

This paper presents a simple and shallow bilateral network architecture (SLBNet) for human pose estimation. Benefiting from the dual resolution feature representations, SLBNet can achieve pretty competitive results on the COCO dataset compared with those state-of-the-art methods, while the size and FLOPs of the model are pretty small. As our model comprises regular convolution layers instead of depthwise separable convolutions, SLBNet achieves state-of-the-art speed in GPU and CPU platforms with a small model. It can be generally applied to resource-limited computing devices.

ACKNOWLEDGMENT

This work is supported by Shenzhen Science and Technology Program (JCYJ20180508152046428)

REFERENCES

[1] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3686–3693, 2014.

[2] Bruno Artacho and Andreas Savakis. Unipose: Unified human pose estimation in single images and videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7035–7044, 2020.

[3] Adrian Bulat and Georgios Tzimiropoulos. Binarized convolutional landmark localizers for human pose estimation and face alignment with limited resources. In *IEEE International Conference on Computer Vision (ICCV)*, pages 3726–3734, 2017.

[4] Yuanhao Cai, Zhicheng Wang, Zhengxiong Luo, Binyi Yin, Angang Du, Haoqian Wang, Xiangyu Zhang, Xinyu Zhou, Erjin Zhou, and Jian Sun. Learning delicate local representations for multi-person pose estimation. In *European Conference on Computer Vision*, pages 455–472. Springer, 2020.

[5] Zhongzheng Cao, Rui Wang, Xiangyang Wang, Zhi Liu, and Xiaoqiang Zhu. Improving human pose estimation with self-attention generative adversarial networks. In *IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 567–572. IEEE, 2019.

[6] Xianjie Chen and Alan Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. *arXiv preprint arXiv:1407.3399*, 2014.

[7] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7103–7112, 2018.

[8] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. Rmpe: Regional multi-person pose estimation. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2353–2362, 2017.

[9] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[11] Yuanduo Hong, Huihui Pan, Weichao Sun, Yisong Jia, et al. Deep dual-resolution networks for real-time and accurate semantic segmentation of road scenes. *arXiv preprint arXiv:2101.06085*, 2021.

[12] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7132–7141, 2018.

[13] Wenbo Li, Zhicheng Wang, Binyi Yin, Qixiang Peng, Yuming Du, Tianzi Xiao, Gang Yu, Hongtao Lu, Yichen Wei, and Jian Sun. Rethinking on multi-stage networks for human pose estimation. *arXiv preprint arXiv:1901.00148*, 2019.

[14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proceedings of the European conference on computer vision (ECCV)*, pages 740–755, Cham, 2014. Springer International Publishing.

[15] Zhengxiong Luo, Zhicheng Wang, Yuanhao Cai, Guanan Wang, Yan Huang, Liang Wang, Erjin Zhou, and Jian Sun. Efficient human pose estimation by learning deeply aggregated representations. *arXiv preprint arXiv:2012.07033*, 2020.

[16] Sachin Mehta, Mohammad Rastegari, Anat Caspi, Linda Shapiro, and Hannaneh Hajishirzi. Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation. In *Proceedings of the european conference on computer vision (ECCV)*, pages 552–568, 2018.

[17] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 483–499, Cham, 2016. Springer International Publishing.

[18] George Papandreou, Tyler Zhu, Nori Kanazawa, Alexander Toshev, Jonathan Tompson, Chris Bregler, and Kevin Murphy. Towards accurate multi-person pose estimation in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3711–3719, 2017.

[19] Rudra PK Poudel, Stephan Liwicki, and Roberto Cipolla. Fast-scnn: fast semantic segmentation network. *arXiv preprint arXiv:1902.04502*, 2019.

[20] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[21] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5686–5696, 2019.

[22] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 529–545, 2018.

[23] Wei Tang, Pei Yu, and Ying Wu. Deeply learned compositional models for human pose estimation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 190–206, 2018.

[24] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1653–1660, 2014.

[25] Fangyun Wei, Xiao Sun, Hongyang Li, Jingdong Wang, and Stephen Lin. Point-set anchors for object detection, instance segmentation and pose estimation. In *European Conference on Computer Vision*, pages 527–544. Springer, 2020.

[26] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 4724–4732, 2016.

[27] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *Proceedings of the European conference on computer vision (ECCV)*, pages 466–481, 2018.

[28] Yi Yang and Deva Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1385–1392, 2011.

[29] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 325–341, 2018.

[30] Feng Zhang, Xiatian Zhu, and Mao Ye. Fast human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3517–3526, 2019.

[31] Wenqiang Zhang, Jiemin Fang, Xinggang Wang, and Wenyu Liu. Efficientpose: Efficient human pose estimation with neural architecture search. *arXiv preprint arXiv:2012.07086*, 2020.

[32] Zhe Zhang, Jie Tang, and Gangshan Wu. Simple and lightweight human pose estimation. *arXiv preprint arXiv:1911.10346*, 2019.

[33] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6230–6239, 2017.