IEEE Copyright notice

FAST SINGLE-PERSON 2D HUMAN POSE ESTIMATION USING MULTI-TASK CONVOLUTIONAL NEURAL NETWORKS

Christos Papaioannidis *Ioannis Mademlis* *Ioannis Pitas*

Department of Informatics, Aristotle University of Thessaloniki, Greece

ABSTRACT

This paper presents a novel neural module for enhancing existing fast and lightweight 2D human pose estimation CNNs, in order to increase their accuracy. A baseline stem CNN is augmented by a collateral module, which is tasked to encode global spatial and semantic information and provide it to the stem network during inference. The latter one outputs the final 2D human pose estimations. Since global information encoding is an inherent subtask of 2D human pose estimation, this particular setup allows the stem network to better focus on the local details of the input image and on precisely localizing each body joint, thus increasing overall 2D human pose estimation accuracy. Furthermore, the collateral module is designed to be lightweight, adding negligible runtime computational cost, so that the unified architecture retains the fast execution property of the stem network. Evaluation of the proposed method on public 2D human pose estimation datasets shows that it increases the accuracy of different baseline stem CNNs, while outperforming all competing fast 2D human pose estimation methods.

***Index Terms*—** 2D human pose estimation, skeleton estimation, Convolutional Neural Networks, Generative Adversarial Networks.

1. INTRODUCTION

2D human pose estimation (2D HPE) from RGB images consists in estimating the 2D pixel coordinates of a predefined set of human body joints on the corresponding 2D input image. Given the current prevalence of computer vision (e.g., [1]), 2D HPE has become an important algorithmic component in applications that involve visually captured human activities. Critical examples include traffic control gesture recognition [2] and pedestrian intention recognition [3]. There, 2D HPE is typically employed as a pre-processing step to extract 2D human skeletons from each video frame, before they are fed as input to the corresponding task classifier/recognizer [4, 5].

Estimating 2D human poses from RGB images can be challenging, as humans may be depicted under a huge range of body postures and/or in highly different scenes and scales. Furthermore, occlusion of specific body parts (e.g., arms, legs) is typical in most cases, rendering 2D HPE even more challenging.

Deep Convolutional Neural Networks (CNNs) have been effectively utilized to tackle these issues, typically by utilizing down-sampling and up-sampling sub-networks in order to predict high-resolution outputs, from which the 2D human body joint positions can be obtained [6, 7]. However, the large number of calculations required by such methods, due to their multiple down-sampling/up-sampling layers, renders them unsuitable for domains where fast execution is crucial to ensure safety (e.g., autonomous robots) [8]. As a result, fast and lightweight 2D HPE methods [9, 10] have emerged, which are specifically designed to achieve high inference speed in embedded systems. However, fast execution often comes at the expense of accuracy, with potentially catastrophic results (e.g., noisy estimations may cause a pedestrian intention recognition system deployed on a self-driving car to misclassify a "crossing" pedestrian as "no-crossing").

Motivated by the insufficient 2D HPE accuracy of the fast approaches that are suitable for embedded execution, this paper presents a method that aims to increase their accuracy without compromising execution speed. It augments typical fast and lightweight CNN architectures with a complementary *collateral* CNN module which encodes global spatial+semantic information and provides it during inference to the main CNN, which we call *stem network*. Thus, the latter can focus on precisely localizing each body joint on the 2D input image. Global information encoding is achieved by training the collateral module as a Generative Adversarial Network (GAN) [11] that reconstructs the colored image representation of the human body structure, while the final 2D human poses are obtained by the stem network through body joint heatmaps regression.

The GAN training framework ensures that the collateral module extracts *global* information from the input image. This is because the Generator only learns to output realistic-looking colored image representations of the human body structure (see Figure 1) that resemble the ground-truth ones, *but not necessarily match them exactly in their local details*. The encoded global information flows from the collateral module to the stem network through additional neural connections that are placed between them. Thus, the stem

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 871479 (AERIAL-CORE).

network exploits this information, along with the spatially localized semantic information encoded in its own features, to predict accurate 2D human poses.

To the best of the authors' knowledge, *global skeleton information extraction based on a collateral convolutional GAN* has not previously appeared as a *separate subtask* in 2D HPE literature. The proposed novel module is designed to achieve exactly this, in order to alleviate the cognitive load/burden of the stem network. Fusing the features of these two subnetworks before the final output prediction permits more accurate 2D HPE, as extensive experiments on two relevant, common public datasets confirm.

2. CNN-BASED 2D HPE

Early 2D HPE approaches aimed to directly regress the 2D body joints' pixel coordinates (e.g., [12]). For example, cascades of pose regressors were utilized to successively refine the body joint estimations before obtaining the final predictions. Under this paradigm, [12] utilized an Iterative Error Feedback process to progressively change the initial body joint location predictions until the error between the estimated and ground-truth 2D human poses is minimized.

In a more recent approach, the body joint locations in the 2D input image are indirectly obtained, by regressing body joint heatmaps [6, 7, 13, 14]. That is, the 2D pixel coordinates of the heatmap maximum value indicate the location of the corresponding body joint in the input image. Following this approach, 2D HPE methods (e.g. [14]) designed CNN architectures that consist of consecutive CNN modules that process/refine their corresponding inputs to predict intermediate feature maps until the final body joint heatmaps are obtained from the last CNN module. For instance, CPN [14] decomposed the 2D HPE problem into two steps. In the first step, a feature pyramid CNN is used to localize the "easy" body joints (e.g., hands), while the resulting multi-scale feature maps are subsequently fused and fed to a second network tasked to detect the "hard" body joints. In an alternative approach, a very simple CNN architecture based on convolutional and deconvolutional layers was proposed in [6] to effectively obtain high-resolution body joint heatmaps, from which the final 2D human poses can be accurately obtained. With the same goal in mind, [7] introduced a CNN architecture that was specifically designed to maintain high-resolution feature maps through the overall procedure.

Embedded execution of 2D HPE algorithms gave rise to methods that aimed to achieve fast inference along with increased 2D HPE accuracy. For example, deep and shallow sub-networks were employed in [9] to process low-resolution and high-resolution inputs, respectively, towards obtaining accurate body joint heatmaps in real-time. [15] designed a lightweight version of the network architecture proposed in [7] by replacing specific computationally intensive neural blocks with more lightweight ones, which achieved good 2D HPE performance with low complexity. Finally, [10] introduced a lightweight CNN architecture that utilized specifically designed convolutional and deconvolutional modules to reduce latency without hurting 2D HPE accuracy.

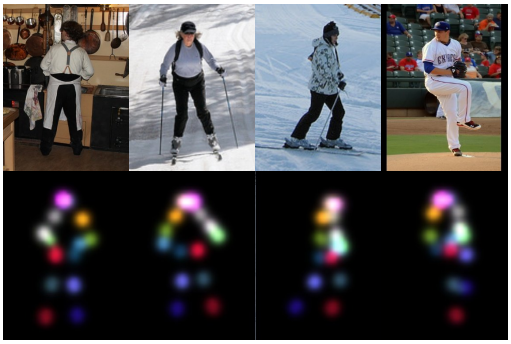

Fig. 1. Examples of input RGB images (1st row), along with the corresponding output human body structure images $\mathbf{S}$ (2nd row).

Differently from these approaches, the proposed method aims to enhance the 2D HPE performance of existing fast and lightweight architectures, while retaining fast execution. This is achieved by explicitly guiding the two neural pathways (the stem network and the proposed module) to separately address the two inherent problems of 2D HPE (i.e., precise body joint detection and global information encoding, respectively) and effectively combine the corresponding outputs to obtain accurate body joint heatmaps.

3. FAST AND ACCURATE 2D HPE

Let $\mathbf{X} \in \mathbb{R}^{M \times N \times 3}$ be an input RGB image and J be any fast and lightweight CNN architecture that can be used to predict 2D body joint heatmaps. The proposed method focuses on enhancing J towards increasing its 2D HPE accuracy, while retaining its fast execution property. In this direction, J is augmented by a collateral module A that is tasked to encode global spatial+semantic information from the input image and pass it to J, from which the final 2D body joint heatmaps are obtained. In this particular setup, J is able to exploit the information encoded by A to precisely detect each body joint on the input image.

3.1. Global Information Encoding

Successful 2D HPE requires encoding both global and local spatial+semantic information. A is used to encode global information and provide it to J, alleviating the latter from this task. Thus, J exploits the information encoded by A and focuses on precisely detecting each body joint.

To achieve this, A is tasked to reconstruct a colored image $\mathbf{S} \in \mathbb{R}^{M \times N \times 3}$ that represents the human body structure of the person that is depicted in the corresponding input image $\mathbf{X}$, via GAN-based Image-to-Image translation (I2I) [16]. $\mathbf{S}$ is carefully constructed to represent the human body structure and also contain identical semantic information to the target of J. This is achieved by centering a 2D Gaussian function at the ground-truth location of each body joint, while assigning

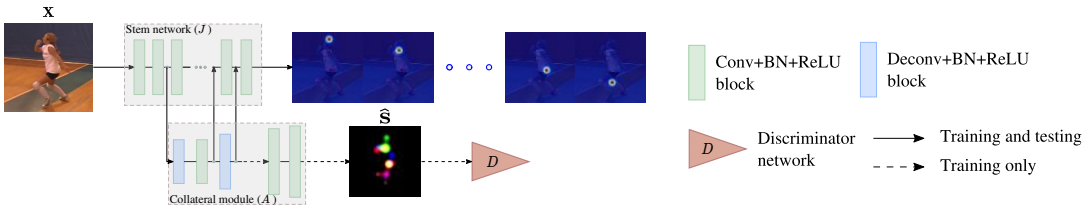

Fig. 2. Training and testing setup of the overall proposed network architecture.

a specific RGB value to it. Using different colors to identify each body joint ensures that ambiguous body joint representations (e.g., left and right wrists) are distinguishable in $\mathbf{S}$. Color assignments can be arbitrary, as long as different body joints are represented by different colors. Random input RGB images along with their corresponding human body structure images $\mathbf{S}$ can be seen in Fig. 1.

In order to encode global information via I2I, A is trained under the GAN learning framework, which relies on the interaction between a Generator and a Discriminator. Therefore, A is designed as a decoding CNN acting on intermediate layer activations of J, serving as a GAN Generator that produces human body structure images $\hat{\mathbf{S}}$ that fit the corresponding input images. Subsequently, a predicted $\hat{\mathbf{S}}$ and the corresponding input image $\mathbf{X}$ are jointly fed to the Discriminator D, which processes the pair and decides whether $\hat{\mathbf{S}}$ is a "fake" one produced by the Generator, or a ground-truth one. A is trained using the multi-objective loss function:

$$\mathcal{L}_A = \min_A \max_D \mathcal{L}_{GAN} + \beta_s \mathcal{L}_s + \beta_v \mathcal{L}_v, \quad (1)$$

where $\mathcal{L}_{GAN}$ is the typical loss function of conditional GANs [16], $\mathcal{L}_s$ is a similarity loss function based on the $\mathcal{L}1$ distance that is used to push the Generator to produce outputs that are close to the target images and $\mathcal{L}_v$ is a typical cross-entropy loss function used to train the Discriminator for predicting body joint visibility. Essentially, $\mathcal{L}_v$ forces D to predict which body joints are visible, in parallel to its main task, thus further strengthen it. As a result, A is forced to produce more accurate human body structure images to fool D. Finally, β_s, β_v are scaling hyper-parameters.

3.2. Unified Architecture

The unified proposed network architecture can be seen in Fig. 2. J acts on the input image, while the collateral module A acts on intermediate features maps of J. In this case, the early layers of J play the role of the encoding network, while A plays the role of the decoding one in a typical encoder-decoder Generator network architecture. Information flow between J and A is realized through connections placed between the neurons of two intermediate layers of A and J. Thus, the semantic features of J are enriched with global information encoded by A.

The unified network is jointly trained for both I2I and for 2D body joint regression, using the following multitask loss:

$$\mathcal{L} = \lambda \mathcal{L}_J + (1 - \lambda)\mathcal{L}_A, \quad (2)$$

where $\mathcal{L}_J$ is a typical body joint heatmaps regression loss that is used to train the corresponding stem network and λ is a hyperparameter for tuning the contribution of $\mathcal{L}_J$ and $\mathcal{L}_A$ to the total loss.

A is carefully designed to be lightweight, having in mind fast execution speed during inference. It consists of three convolutional and two deconvolutional layers, in order to increase the feature map resolution, while maintaining a relatively low number of parameters. Each convolutional and deconvolutional layer is followed by a Batch Normalization and a ReLu layer. Also, for simplicity, all convolutional and deconvolutional layers use 3×3 kernels. D is based on a standard PatchGAN [16] classifier, which was extended by an extra fully connected layer for the joint visibility classification task. J can be any fast dense image prediction CNN. Note that during the inference stage, the two last layers of A and the entire Discriminator D can be completely discarded to avoid computational overhead.

4. EXPERIMENTAL EVALUATION

The unified network architecture was trained using (2) for 200 epochs. J and A were trained using a learning rate of 0.01, which is reduced in each epoch using the "poly" learning rate strategy. D is trained with a constant learning rate of 0.00002. In all cases, the Adam optimizer was used. The batch size was 64, while λ in (2) was empirically set to $\lambda = 0.7$, in order to promote the 2D HPE task over I2I. The scaling hyperparameters β_s and β_v were set to 1 and 0.1, respectively, ensuring the smooth training of A. The online training data augmentation method of [7] was also adopted.

The proposed method was evaluated on two public 2D HPE datasets, COCO Keypoints 2017 [17] and MPII Human Pose [18], following the common two-stage top-down evaluation paradigm [6, 7, 14]. In order to ensure a fair comparison, the person detections provided by [6, 7] are used both for COCO *val2017* and *test-dev2017* sets. In MPII, each person location is provided with the dataset. The average precision (AP) and average recall (AR) metrics are reported for COCO, while for MPII, the head-normalized probability of correct keypoint (PCKh@0.5) metric is used. Execution speed is measured in Frames Per Second (FPS). Execution speed is also used as a fair model complexity measurement, since the numbers of model parameters and flops that are typically reported involve only specific layers of the model (convolutional, linear), ignoring any extra calculations required by other operations (e.g., resizing, addition, multiplication, etc.).

Table 1. Results on enhancing the baselines BiSeNet [19] and Lite-HRNet [15] on the COCO [17] *val2017* set and the MPII [18] validation set.

Method	Backbone	COCO *val2017* AP		MPII val PCKh@0.5
		Input Res.		Input Res.
		256×192	384×288	256×256
BiSeNet [19]	ResNet-18	68.4	71.4	87.3
BiSeNet + A	ResNet-18	**70.2**	**72.5**	**88.2**
BiSeNet [19]	ResNet-50	71.4	71.6	88.1
BiSeNet + A	ResNet-50	**73.7**	**74.0**	**89.7**
Lite-HRNet [15]	Lite-HRNet-18	64.8	67.6	86.1
Lite-HRNet + A	Lite-HRNet-18	**65.3**	**70.5**	**87.1**
Lite-HRNet [15]	Lite-HRNet-30	67.2	70.4	87.0
Lite-HRNet + A	Lite-HRNet-30	**68.0**	**71.7**	**88.3**

Table 2. Evaluation results on COCO [17] *val2017*. FPS-D and FPS-M denote Frames Per Second (inference speed) using a GeForce GTX 1080 Ti GPU and a Nvidia Jetson Xavier computing board, respectively. Input resolution is 384×288.

Method	FPS-D	FPS-M	AP	AP^{50}	AP^{75}	AR
CPN [14]	—	—	70.6	—	—	—
CPN_{OHKM} [14]	—	—	71.6	—	—	—
BiSeNet [19]	**42.3**	**17.1**	71.6	89.5	79.4	77.3
SB [6]	31.3	11.9	72.2	89.3	78.9	77.6
PPNet [9]	31.9	12.3	73.2	88.9	80.0	78.4
BiSeNet + A (*ours*)	37.7	13.4	**74.0**	**90.0**	**81.3**	**79.2**

In contrast, execution speed measured in the same machine involves all the calculations that are required to produce the final estimations.

BiSeNet [19] and Lite-HRNet [15] were separately adopted as the fast stem network J. A comparison between their baseline versions and their variants that have been architecturally augmented with the proposed module is presented in Table 1, for both COCO *val2017* and MPII validation sets. Results for alternative backbones of different complexity are also reported. Note that *BiSeNet* was adjusted for the 2D HPE task by simply tasking it to predict body joint heatmaps instead of segmentation maps.

The 2 CNNs that have been augmented with the proposed module (J+A) outperform the corresponding baselines (J-only) for all tested backbones in both datasets. This showcases that *more accurate 2D human poses can indeed be predicted by splitting up the 2D HPE task between A and J and facilitating information exchange between them through the added neural connections*.

The best performing variant of the proposed method (ResNet-50-based *BiSeNet* + A) is then compared against competing methods of similar complexity on the COCO *val2017* set in Table 2, for input image resolution of 384×288 pixels. Inference speed in FPS is measured for all available competing methods using both a high-end desktop PC equipped with a *GeForce GTX 1080 Ti* GPU and an *Nvidia Jetson Xavier* embedded AI computing board. As it can be seen, the proposed architecture outperforms all competing methods, while maintaining increased inference speed. It is faster and more accurate than the best performing competing methods *SB* and *PPNet*, while it is slower than the fastest competitor *BiSeNet* only by 3.7-4.6 FPS, which is however outperformed by the proposed method by a 2.4 AP score. Furthermore, the 2D HPE accuracy of the proposed method is evaluated on the COCO *test-dev2017* set. Comparisons are presented in Table 3 and indicate similar behaviour.

Table 3. Evaluation results on COCO [17] *test-dev2017*.

Method	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR
OpenPose [20]	61.8	84.9	67.5	57.1	68.2	66.5
Assoc. Emb. [21]	65.5	86.8	72.3	60.6	72.6	70.2
PersonLab [22]	68.7	89.0	75.4	64.1	75.5	75.4
MultiPoseNet [23]	69.6	86.3	76.6	65.0	76.3	73.5
Mask-RCNN [24]	63.1	87.3	68.7	57.8	71.4	—
SB [6]	71.5	91.1	78.7	67.8	78.0	76.9
RMPE [25]	72.3	89.2	79.1	68.0	78.6	—
BiSeNet + A (*ours*)	**73.3**	**92.1**	**81.3**	**70.0**	**79.0**	**78.6**

Table 4. Evaluation results on the MPII [18] test set. Input resolution is 256×256.

Method	Head	Should.	Elb.	Wrist	Hip	Knee	Ank.	Total
CPMs [26]	97.8	95.0	88.7	84.0	88.4	82.8	79.4	88.5
SB [6]	98.2	**96.4**	91.0	86.0	90.4	86.3	82.3	90.5
HG [13]	98.2	96.3	91.2	87.1	90.1	87.4	83.6	90.9
GLN [27]	98.1	96.2	91.2	87.2	89.8	87.4	84.1	91.0
SA-GCN [28]	97.1	96.1	91.5	86.8	**90.7**	87.6	84.3	91.1
BiSeNet + A (*ours*)	**98.3**	96.3	**91.7**	**87.6**	90.3	**88.0**	**84.5**	**91.3**

Comparisons against competitors in the MPII test set is reported in Table 4. An input resolution of 256×256 pixels is used in all cases for a fair comparison. As it can be seen, the proposed method yielded increased 2D HPE accuracy compared to all competitors.

5. CONCLUSIONS

This paper introduced a novel, collateral module A for augmenting any fast and lightweight 2D human pose estimation CNN architecture J. A relies on Image-to-Image Translation in order to encode global body information and pass it to J during inference through additional neural connections placed between them. As a result, J is exempt from this subtask and focuses only on precisely localizing each body joint on the 2D input image, thus ultimately achieving increased 2D human pose estimation accuracy. The overall architecture is trained in a unified end-to-end manner, using a multitask loss function. Importantly, A is specifically designed to only add negligible computational cost during inference, so that the fast execution properties of J are retained. Evaluation on two public datasets shows that the proposed method not only increases the accuracy of baseline CNNs, but also outperforms all competing fast 2D human pose estimation methods.

6. REFERENCES

[1] E. Kakaletsis, E. Symeonidis, M. Tzelepi, I. Mademlis, A. Tefas, N. Nikolaidis, and I. Pitas, "Computer vision for autonomous UAV flight safety: An overview and a vision-based safe landing pipeline example," *ACM Computing Surveys*, vol. 54, no. 9, pp. 1–37, 2021.

[2] J. Wiederer, A. Bouazizi, U. Kressel, and V. Belagiannis, "Traffic control gesture recognition for autonomous vehicles," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[3] K. Chen, X. Song, and X. Ren, "Pedestrian trajectory prediction in heterogeneous traffic using pose keypoints-based convolutional encoder-decoder network," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 5, pp. 1764–1775, 2020.

[4] C. Papaioannidis, D. Makrygiannis, I. Mademlis, and I. Pitas, "Learning fast and robust gesture recognition," in *Proceedings of the European Signal Processing Conference (EUSIPCO)*, 2021.

[5] D. Makrygiannis, C. Papaioannidis, I. Mademlis, and I. Pitas, "Optimal video handling in on-line hand gesture recognition using Deep Neural Networks," in *Proceedings of the IEEE Symposium Series on Computational Intelligence (SSCI)*, 2021.

[6] B. Xiao, H. Wu, and Y. Wei, "Simple baselines for human pose estimation and tracking," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[7] 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 (CVPR)*, 2019.

[8] P. Nousi, I. Mademlis, I. Karakostas, A. Tefas, and I. Pitas, "Embedded UAV real-time visual object detection and tracking," in *Proceedings of the IEEE International Conference on Real-time Computing and Robotics (RCAR)*, 2019.

[9] L. Zhao, N. Wang, C. Gong, J. Yang, and X. Gao, "Estimating human pose efficiently by parallel pyramid networks," *IEEE Transactions on Image Processing*, vol. 30, pp. 6785–6800, 2021.

[10] Y. Wang, M. Li, H. Cai, W.-M. Chen, and S. Han, "Lite pose: Efficient architecture design for 2d human pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[11] 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 (NIPS)*, 2014.

[12] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik, "Human pose estimation with iterative error feedback," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[13] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[14] 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 (CVPR)*, 2018.

[15] C. Yu, B. Xiao, C. Gao, L. Yuan, L. Zhang, N. Sang, and J. Wang, "Lite-hrnet: A lightweight high-resolution network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[16] P. Isola, J. Zhu, T. Zhou, and A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[17] 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 (ECCV)*, 2014.

[18] 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 (CVPR)*, 2014.

[19] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Bisenet: Bilateral segmentation network for real-time semantic segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[20] 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 (CVPR)*, 2017.

[21] A. Newell, Z. Huang, and J. Deng, "Associative embedding: End-to-end learning for joint detection and grouping," *arXiv preprint arXiv:1611.05424*, 2016.

[22] G. Papandreou, T. Zhu, L.-C. Chen, S. Gidaris, J. Tompson, and K. Murphy, "Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[23] M. Kocabas, S. Karagoz, and E. Akbas, "Multiposenet: Fast multi-person pose estimation using pose residual network," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[24] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[25] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu, "RMPE: Regional multi-person pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[26] 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 (CVPR)*, 2016.

[27] K. Sun, C. Lan, J. Xing, W. Zeng, D. Liu, and J. Wang, "Human pose estimation using global and local normalization," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[28] Y. Bin, Z.-M. Chen, X.-S. Wei, X. Chen, C. Gao, and N. Sang, "Structure-aware human pose estimation with graph convolutional networks," *Pattern Recognition*, vol. 106, pp. 107410, 2020.