PAPER

FSPose: A Heterogeneous Framework with Fast and Slow Networks for Human Pose Estimation in Videos

Jianfeng XU[†a)], **Satoshi KOMORITA**[†], ***Members***, ***and*** **Kei KAWAMURA**[†], ***Senior Member***

SUMMARY We propose a framework for the integration of heterogeneous networks in human pose estimation (HPE) with the aim of balancing accuracy and computational complexity. Although many existing methods can improve the accuracy of HPE using multiple frames in videos, they also increase the computational complexity. The key difference here is that the proposed heterogeneous framework has various networks for different types of frames, while existing methods use the same networks for all frames. In particular, we propose to divide the video frames into two types, including key frames and non-key frames, and adopt three networks including slow networks, fast networks, and transfer networks in our heterogeneous framework. For key frames, a slow network is used that has high accuracy but high computational complexity. For non-key frames that follow a key frame, we propose to warp the heatmap of a slow network from a key frame via a transfer network and fuse it with a fast network that has low accuracy but low computational complexity. Furthermore, when extending to the usage of long-term frames where a large number of non-key frames follow a key frame, the temporal correlation decreases. Therefore, when necessary, we use an additional transfer network that warps the heatmap from a neighboring non-key frame. The experimental results on PoseTrack 2017 and PoseTrack 2018 datasets demonstrate that the proposed FSPose achieves a better balance between accuracy and computational complexity than the competitor method. Our source code is available at https://github.com/Fenax79/fspose.

key words: *human pose estimation, heterogeneous networks, temporal correlation, fast networks, slow networks*

1. Introduction

In the last decade, many technologies have been proposed for human pose estimation (HPE), which are able to localize human joints (also known as *keypoints*) in images [1]. Because it provides a compact representation of body shape and motion, HPE is widely used in human-computer interactions, gaming, virtual reality, video surveillance, sports analysis, and for understanding human activity [1], [2].

Most existing technologies, such as HRNet [3], are designed for the aforementioned applications with very high computational complexity. Furthermore, the accuracy can be improved by using temporal correlation, such as PoseWarper [4] with a cost of even higher computational complexity. In contrast, some applications require lightweight processing under a limited computational resource, e.g., a real-time application run on a mobile device. For this purpose, there are well-known backbones, such as the MobileNet family [5], [6], EfficientNet [7], and ShuffleNet [8]. However, their accuracy will likely decrease substantially compared to those technologies with high computational complexity, such as HRNet. To the best of our knowledge, few studies have focused on uniting the high accuracy and low computational complexity in the HPE task in videos. In this paper, our goal is to achieve a good balance between accuracy and computational complexity by combining heterogeneous networks for different types of frames in videos.

In terms of the balance between accuracy and computational complexity, the performance of existing methods using temporal correlation, such as PoseWarper or AI coach [9], has a very small gain from our preliminary experimental results. One of the reasons for this is the limitedly gained synergies from the homogeneous network architectures. For example, suppose we have a pose that is difficult to estimate in several frames including the current frame. In that case, if homogeneous networks are used, it will probably fail to estimate the pose in all frames when it fails in the current frame due to their similar performance capabilities. In contrast, our idea is that it may be better to propose a heterogeneous framework, referred to as *FSPose*, including a *slow network*, a *fast network* and a *transfer network* for key frames and non-key frames as shown in Fig. 1. A slow network is a neural network that probably has high accuracy but high computational complexity; a fast network is a neural network that probably has low accuracy but low computational complexity; and a transfer network is a neural network to warp the heatmap from one frame to another. Even though the fast network fails to estimate the pose in the current frame, it is possible to warp the heatmap via the transfer network from the slow network that has a higher accuracy and hopefully estimate a more correct pose. This example describes a special case for the synergy effect from slow networks. In other cases where the transfer network fails, a fast network helps estimate the pose in the current frame.

In a heterogeneous framework, it is possible to reduce the computational complexity by setting a large number of non-key frames, i.e., long-term frames. The issue here is that it is rather challenging to extend to the usage of long-term frames because temporal correlation decreases depending on the distance between two frames. To realize the long-term frames, we use an additional transfer network that warps the heatmap from the neighboring frame for those non-key frames far away from their key frames. As a result, the length of a frame set flexibly covers from short-term to long-term frames as shown in Fig. 1, which realizes the ben-

Manuscript received October 13, 2022.
Manuscript revised January 30, 2023.
Manuscript publicized March 20, 2023.

[†]The authors are with KDDI Research, Inc., Fujimino-shi, 356–8502 Japan.
a) E-mail: ji-xu@kddi.com

DOI: 10.1587/transinf.2022EDP7182

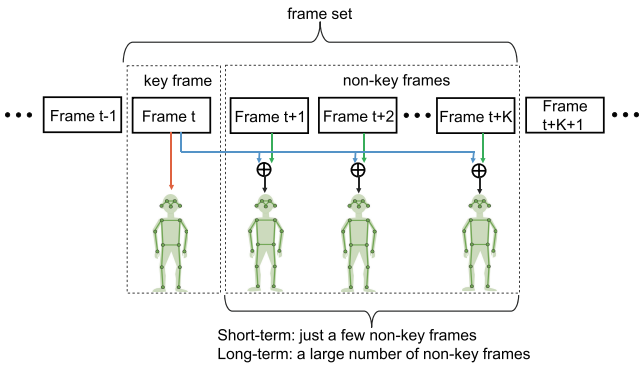


Fig. 1 A video is divided into two types of frames (key frames and non-key frames), where each key frame is followed by a fixed number of non-key frames. The set of a key frame and its following non-key frames is defined as a *frame set*. For the key frame, a slow network (orange arrow) is used. For the non-key frame, a fast network (green arrow) is fused with a transfer network (blue arrow) that warps the heatmap from the key frame.

efit of allowing a different choice on balancing accuracy and computational complexity.

In summary, the main contributions of this paper include the following:

- We propose a heterogeneous framework that uses slow networks, fast networks and transfer networks for different types of frames, i.e., key frames and non-key frames.
- We further improve our system's performance by using long-term frames. We introduce an additional transfer network for those non-key frames that are not adjacent to their key frames.
- We conduct experiments on PoseTrack 2017 [10] and PoseTrack 2018 [11] datasets to demonstrate that the proposed FSPose achieves a better balance between accuracy and computational complexity than the competitor method, i.e., PoseWarper [4].

2. Related Work

In this section, a brief overview of HPE methods is given in both still images and videos.

2.1 Human Pose Estimation in Still Images

In the past decade, many papers on human pose estimation have been published [1], [12], [13]. They are usually divided into two categories: bottom-up and top-down approaches. Because top-down approaches reported better performance [1], this paper also belongs to the top-down approaches. In this section, we briefly survey the typical networks with high complexity/high accuracy and lightweight ones.

As a typical bottom-up approach, OpenPose [14] detects all the body joints in the first stage and then associates them with person instances in the second stage. One of the advantages of bottom-up approaches is that the computational cost changes little even if the number of people in the images increases. One of the disadvantages is that the accuracy worsens when the association of joints has errors. More details can be found in the survey paper [15].

For top-down approaches [15], there are basically two steps. In the first step, the bounding boxes of people are detected in the input image, and in the second step, the joint locations are estimated in each bounding box. Here are three typical top-down approaches. The Cascaded Pyramid Network (CPN) [16] proposes a network structure that includes GlobalNet and RefineNet. The former extracts good feature representation, while the latter is employed to address the "hard" examples. HRNet proposes an architecture that preserves high-resolution feature maps, which consists of multiple branches with different resolutions. Furthermore, PoseNAS [17] proposes using neural architecture search (NAS) to automatically discover better network architectures for pose encoders and pose decoders.

The above technologies require the high computational complexity to conduct the inference. For HPEs in mobile applications, many lightweight network architectures have been proposed to replace the backbones in the above technologies [18]. The MobileNet family [5], [6] uses depthwise separable convolution to reduce the model size and computational complexity. EfficientNet [7] uses NAS covering network depth, width, and resolution to further optimize MobileNetV2 [6]. Alternatively, ShuffleNet [8] factorizes the weight matrix into a product of a permutation matrix and a block diagonal matrix.

2.2 Exploiting Temporal Information in Videos

For HPE in videos, it is essential to exploit their temporal information [10], [11]. Several earlier methods [10], [11], [19] approached the video pose estimation task as a two-stage problem, first detecting the body joints in individual frames and then applying temporal filtering techniques. Later, recurrent networks, especially LSTM [20] and GRU [21], were proposed for pose estimation [22], [23]. In addition, 3D convolution is also useful for temporal information [24], [25]. Girdhar et al. [24] extended Mask-RCNN with 3D convolution for human pose estimation. As a powerful tool for exploiting temporal information, optical flow was often used to temporally warp the heatmaps from the preceding frame to the current frame [26], [27]. In addition to the visual cues, a graph neural network (GNN) [28] is used to learn the pose dynamics directly, thereby enabling the recovery of missed poses and the refinement of estimated poses.

Recently, heatmap transfer/warping was realized by designing a particular subnet (referred to as the *transfer network* in this paper) [4], [9]. PoseWarper [4], which won the PoseTrack 2017 Challenge [10], [11], proposes convolutional layers with different dilation rates and deformable convolutions [29] to warp the heatmap from one frame to another. PoseWarper is regarded as the competitor method in this paper due to its outstanding performance in the PoseTrack 2017 challenge [10]. AI coach [9] proposes a spatial-

temporal relation network for HPE in sports videos, where the temporal relation of a specific keypoint is extracted by a distribution of the position offset. These methods [4], [9] use homogeneous networks for all frames, which is different from ours.

3. Proposed Method

In this section, we present our heterogeneous framework, where we combine slow and fast networks via transfer networks. The proposed method not only improves the balance between accuracy and computational complexity but also realizes the usage of long-term frames.

3.1 Overview

As shown in Fig. 1, we divide a video into two types of frames: key frames and non-key frames. We select one frame as the key frame every $(K + 1)$ frames in the video, where slow networks, such as HRNet, are used. The number of non-key frames is defined as K. In non-key frames, fast networks, such as MobileNetV2, are used. The heatmap from fast network and that from transfer network are fused, where the transfer network warps the heatmap from slow network in the key frame, as shown in Fig. 2. Therefore, key frames are independent, which uses no information from other frames, and their heatmaps are warped to non-key frames.

In this paper, we adopt a similar transfer network with PoseWarper. The complexity of the transfer network is lighter than that of MobileNetV2, as shown in Table 1. Our system uses heterogeneous networks that contain fast and slow networks for different types of frames, while PoseWarper uses homogeneous networks all frames. Furthermore, because the input heatmaps of the transfer network come from two different networks, their resolutions are different from each other. Accordingly, in contrast to PoseWarper, it is necessary to resize the heatmaps and train the transfer networks from scratch. However, it should be noted that other transfer networks, such as those used in AI coach [9], can be employed in our system because they share the similar functions of warping the heatmap from key frame to non-key frame. In principle, the proposed heterogeneous networks have the ability to provide a better balance between accuracy and computational complexity on an arbitrary network architecture used in the fast/slow networks.

In this paper, we adopt floating-point operations (FLOPs) as a metric of computational complexity [7], [8], [30] and object keypoint similarity (OKS)-based mean average precision (mAP) as a metric of accuracy [10], [11], [31], [32].

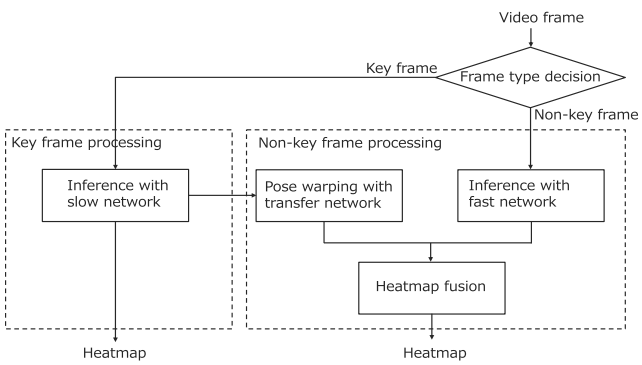


Fig. 2 Flow chart of the proposed heterogeneous framework.

3.2 Preliminary Experiment for the Homogeneous Framework

In the preliminary experiments, we investigate the limitation of existing methods that use homogeneous frameworks, such as the framewise method and PoseWarper, in terms of the balance between accuracy and computational complexity and also decide the parameters that we should use, such as the backbones and input resolutions. Note that the results in the top-left areas basically have a good balance of accuracy and computational complexity when they are plotted in figures, such as Fig. 3.

We compare HRNet_W32 [3] (a typical network with high computational complexity) and MobileNetV2 (a typical network for mobile devices) as the backbone of PoseWarper and use two neighboring frames to fuse the heatmaps via the transfer network. In addition, there are several choices for the input resolution in HRNet_W32. We adopt two typical resolutions as 384x288 and 256x192. The former is the highest resolution in HRNet_W32 and the latter is a smaller one but has just a little drop in accuracy as reported [3]. For MobileNetV2, we use a popular input resolution of 224x224. HRNet_W32, MobileNetV2, and the transfer networks are fine-tuned or trained on the training dataset of PoseTrack 2018.

The results of the preliminary experiments are shown in Fig. 3, where we have three observations. (1) As shown in Fig. 3, it is demonstrated that the resolution of 384x288 in HRNet_W32 only has a small gain in accuracy while largely increasing the computational complexity compared to the resolution of 256x192. Therefore, we use the lat-

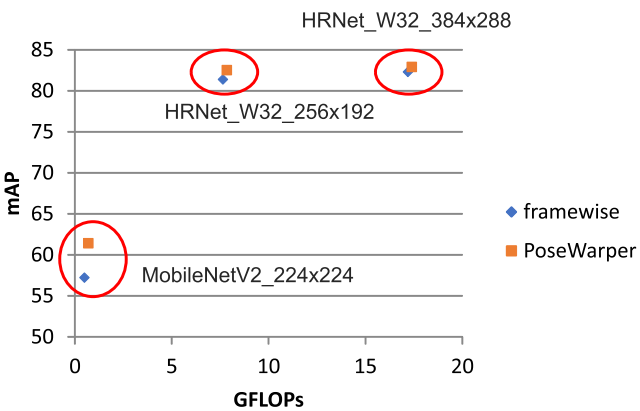


Fig. 3 Comparison of different choices in the PoseWarper and framewise methods on the PoseTrack 2018 dataset. Red circles denote that the same backbones are used.

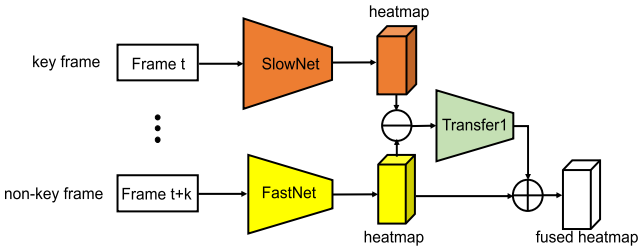


Fig. 4 The basic architecture of a heterogeneous framework with fast networks, slow networks, and transfer networks in a frame set.

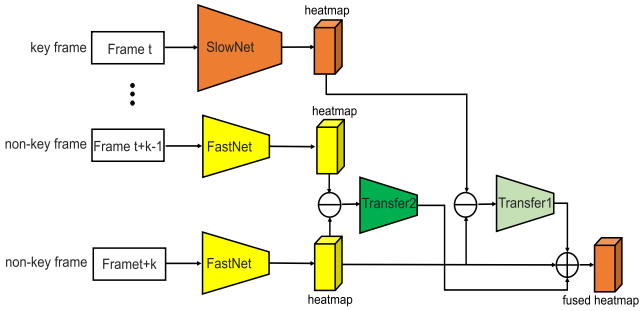


Fig. 5 An additional transfer network is used for the latter part of the non-key frames in a frame set, which warps the heatmap of the previous frame to the current frame. Two transfer networks, which do not share weights, are trained independently.

ter in the following investigation. (2) The results of MobileNetV2_224x224 show that the gain from framewise to PoseWarper is larger than HRNet_W32, but the absolute value of accuracy is much lower than HRNet_W32. (3) As shown in the circles of Fig. 3, if we compare the performance differences between the PoseWarper and the framewise methods, the results are pushed to the top-right areas instead of the top-left areas when using the backbones of HRNet_W32 or MobileNetV2. This is the issue of homogeneous network architectures from multiple frames, which increases both the accuracy and the computational complexity.

3.3 Proposed Heterogeneous Framework

In this section, we describe the basic architecture of the proposed heterogeneous framework which can solve the issue of homogeneous network architectures as mentioned in Sect. 3.2.

Figure 4 shows the basic architecture of a heterogeneous framework with fast and slow networks in a frame set. Suppose we have Frame (t) as a key frame, denoted as $\boldsymbol{F}(t)$, and Frame $(t+k) : 1 \le k \le K$ as non-key frames, denoted as $\boldsymbol{F}(t+k)$. The output heatmaps of slow networks (HRNet), fast networks (MobileNetV2), and transfer networks (referred to as Transfer1) can be computed as follows:

$$\boldsymbol{H}^s(t) = \boldsymbol{f}^s(\boldsymbol{F}(t); \boldsymbol{W}^s) \tag{1}$$

$$\boldsymbol{H}^f(t+k) = \boldsymbol{f}^f(\boldsymbol{F}(t+k); \boldsymbol{W}^f) \tag{2}$$

$$\boldsymbol{H}^w(t+k) = \boldsymbol{f}^w(\boldsymbol{H}^s(t), \boldsymbol{H}^f(t+k); \boldsymbol{W}^w) \tag{3}$$

where $\boldsymbol{f}^s$, $\boldsymbol{f}^f$, and $\boldsymbol{f}^w$ denote the slow networks, fast networks, and transfer networks with parameters of $\boldsymbol{W}^s$, $\boldsymbol{W}^f$, and $\boldsymbol{W}^w$, respectively. $\boldsymbol{H}^s(t)$ denotes the heatmap from the slow network of frame (t), $\boldsymbol{H}^f(t+k)$ denotes the heatmap from the fast network of frame $(t+k)$, and $\boldsymbol{H}^w(t+k)$ denotes the heatmap from the transfer network using frame (t) and frame $(t+k)$.

Then, for non-key frames, we fuse $\boldsymbol{H}^f(t+k)$ and $\boldsymbol{H}^w(t+k)$ together by simple averaging as follows:

$$\boldsymbol{H}^p(t+k) = \alpha \cdot \boldsymbol{H}^f(t+k) + (1-\alpha) \cdot \boldsymbol{H}^w(t+k) \tag{4}$$

where $\boldsymbol{H}^p(t+k)$ denotes the fused heatmap for non-key frame $(t+k)$, and α is a weight between 0 and 1, which was set as 0.5 in our experiment. Note that an adaptive weight can also be adopted, which may be proportional to k.

Thus, the joint locations are estimated from the locations of the maximum value on heatmaps, which can be the fused heatmap $\boldsymbol{H}^p(t+k)$ for non-key frames or the heatmap $\boldsymbol{H}^s(t)$ for key frames.

3.4 Extension to Long-Term Frames

To effectively reduce the computational complexity, there should be as many non-key frames that use fast networks in a heterogeneous framework. Obviously, it is important to know how many frames are possible to warp the heatmap effectively.

Since the temporal correlation weakens considerably for the latter part of the non-key frames in the case that the frame set length is large, the idea is that we should enhance the correlation with an additional frame that provides a strong temporal correlation with the current frame. Statistically, the neighboring frame has the strongest temporal correlation. Accordingly, suppose a frame $\boldsymbol{F}(t+k)$ in the latter part be from the λ-th frame to the K-th frame, where λ is set as $K/2$ in our experiments; we select the previous frame $\boldsymbol{F}(t+k-1)$ as an additional reference frame, as shown in Fig. 5, the heatmap of which is warped to the current frame $\boldsymbol{F}(t+k)$ via another transfer network (Transfer2 in Fig. 5) by the following:

$$\boldsymbol{H}^{wp}(t+k) = \boldsymbol{f}^{wp}(\boldsymbol{H}^f(t+k-1), \boldsymbol{H}^f(t+k); \boldsymbol{W}^{wp}) \tag{5}$$

where $\boldsymbol{f}^{wp}$ denotes the additional transfer network with parameters of $\boldsymbol{W}^{wp}$. $\boldsymbol{H}^f(t+k-1)$ and $\boldsymbol{H}^f(t+k)$ denote the heatmaps from the fast network of frames $\boldsymbol{F}(t+k-1)$ and $\boldsymbol{F}(t+k)$. Note that the network architecture of the new Transfer2 network is the same as that of the Transfer1 network except the input resolution because they share the same functions. Because the input heatmaps of Transfer2 are both from fast networks, we need to train it independently.

Then, as shown in Fig. 5, three heatmaps are fused by the following:

$$\begin{aligned}\boldsymbol{H}^p(t+k) \;=\; & \alpha \cdot \boldsymbol{H}^f(t+k) + \beta \cdot \boldsymbol{H}^{wp}(t+k) \\ & + (1-\alpha-\beta) \cdot \boldsymbol{H}^w(t+k)\end{aligned} \tag{6}$$

where α and β are two weights, which was set as 1/3 in our experiments. Note that adaptive weights can also be adopted.

4. Experiments

In this section, we present our results on the PoseTrack 2017 [10] and PoseTrack 2018 datasets [11] to demonstrate the effectiveness of the proposed FSPose.

4.1 Datasets

Based on the raw videos provided by the MPII Human Pose dataset [33], the PoseTrack dataset is a large-scale benchmark for HPEs in videos and has frequently been used in many papers, including PoseWarper and HRNet. PoseTrack 2017 contains 250 video sequences for training and 50 video sequences for validation, while PoseTrack 2018 increases the number of video sequences and contains 593 for training and 170 for validations. Because human detection is beyond the scope of this paper, which is the prerequisite of HPE tasks, we use exactly the same results as those of PoseWarper. In detail, the ground truth person bounding boxes are directly used to crop the person areas during training. Because the ground truth of the test videos is unavailable, we conduct the evaluation on the validation videos as PoseWarper did.

4.2 Evaluation Metrics

In the literature [1], [15], there are no evaluation metrics available, and the results are only plotted in figures, such as Fig. 8. It is preferable to design an objective metric to evaluate the balance between accuracy and computational complexity. The naive performance is the linear interpolation of fast network (MobileNetV2) and slow network (HRNet_W32) as shown by the blue dotted line in Fig. 8. Our metric measures how far the evaluated method is away from the linear interpolation. Accordingly, the metric is defined as the distance from a result of the evaluated method to the line connecting the fast network and the slow network as shown by the orange dotted line in Fig. 8. Suppose the results of the fast network and the slow network are $P1 = (x_1, y_1)$ and $P2 = (x_2, y_2)$, respectively. The distance of the result of the evaluated method $P3 = (x_0, y_0)$ from the line connecting $P1$ and $P2$ is calculated as follows:

$$score(P3) = \frac{|(x_2-x_1)(y_1-y_0)-(x_1-x_0)(y_2-y_1)|}{\sqrt{(x_2-x_1)^2+(y_2-y_1)^2}} \quad (7)$$

where *score* is the metric to show the distance to the linear interpolation of fast network and slow network, $P1$ and $P2$ denote the results of fast network and slow network, and $P3$ denotes the result of the evaluated method, as shown by the red fonts in Fig. 8.

4.3 Experimental Settings

Our loss function used in training the networks is defined as the mean square error between the predicted heatmaps and the ground truth heatmaps, commonly used in most HPE papers, including PoseWarper and HRNet. The ground truth heatmaps are generated by applying 2D Gaussian smoothing centered on the group truth location of each keypoint.

Our entire framework, as shown in Fig. 5, can be trained in an end-to-end learning approach. However, because the size of the dataset is insufficient to apply end-to-end learning, we independently train the four networks to avoid overfitting: fast networks for non-key frames, slow networks for key frames, transfer networks from key frame to non-key frame (Transfer1 in Fig. 5), and transfer networks between two non-key frames (Transfer2 in Fig. 5). We fine-tune the fast network (MobileNetV2) and slow network (HRNet_W32) using their pretrained models on the COCO dataset [30]. When training the transfer networks from scratch, we randomly select a key frame from the five preceding frames. All the training is terminated after 20 epochs. During testing, mAP and FLOPs are calculated by third-party libraries, such as fvcore [34].

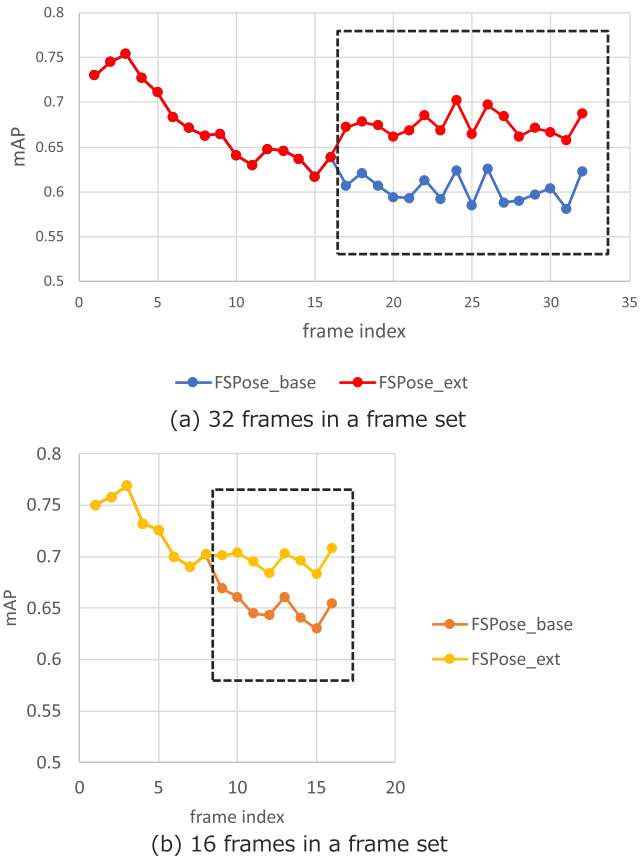


Fig. 6 The mAP accuracy for each non-key frame with the distance from the key frame. FSPose_base: just the heatmap from the key frame is warped and fused as described in Sect. 3.3. FSPose_ext: Both the heatmaps from the key frame and the previous frame are warped and fused in the latter half of the non-key frames as described in Sect. 3.4. The accuracy decreases in the latter half of the non-key frames for FSPose_base, while the accuracy can be improved considerably in the latter half of the non-key frames for FSPose_ext.

Table 1 Accuracy and computational complexity in non-key frames of PoseTrack 2017 for fast networks and transfer networks used in FSPose_ext16. Transfer1 denotes the transfer network from key frame to non-key frame, and Transfer2 denotes the transfer network between non-key frames. As a reference, we also list the data of slow network in the last row of the table.

network	FLOPs(G)	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Mean
Fast network (MobileNetV2 [6])	0.49	70.34	66.63	51.34	39.35	57.68	48.46	38.24	54.30
Transfer1	0.20	73.51	**76.16**	**60.71**	**46.27**	**67.19**	**57.33**	**47.95**	**62.12**
Transfer2	**0.09**	**74.72**	69.99	53.48	40.22	59.49	53.40	46.95	58.08
Slow network (HRNet_W32 [3])	7.65	83.07	89.85	83.93	75.59	82.15	80.83	73.74	81.43

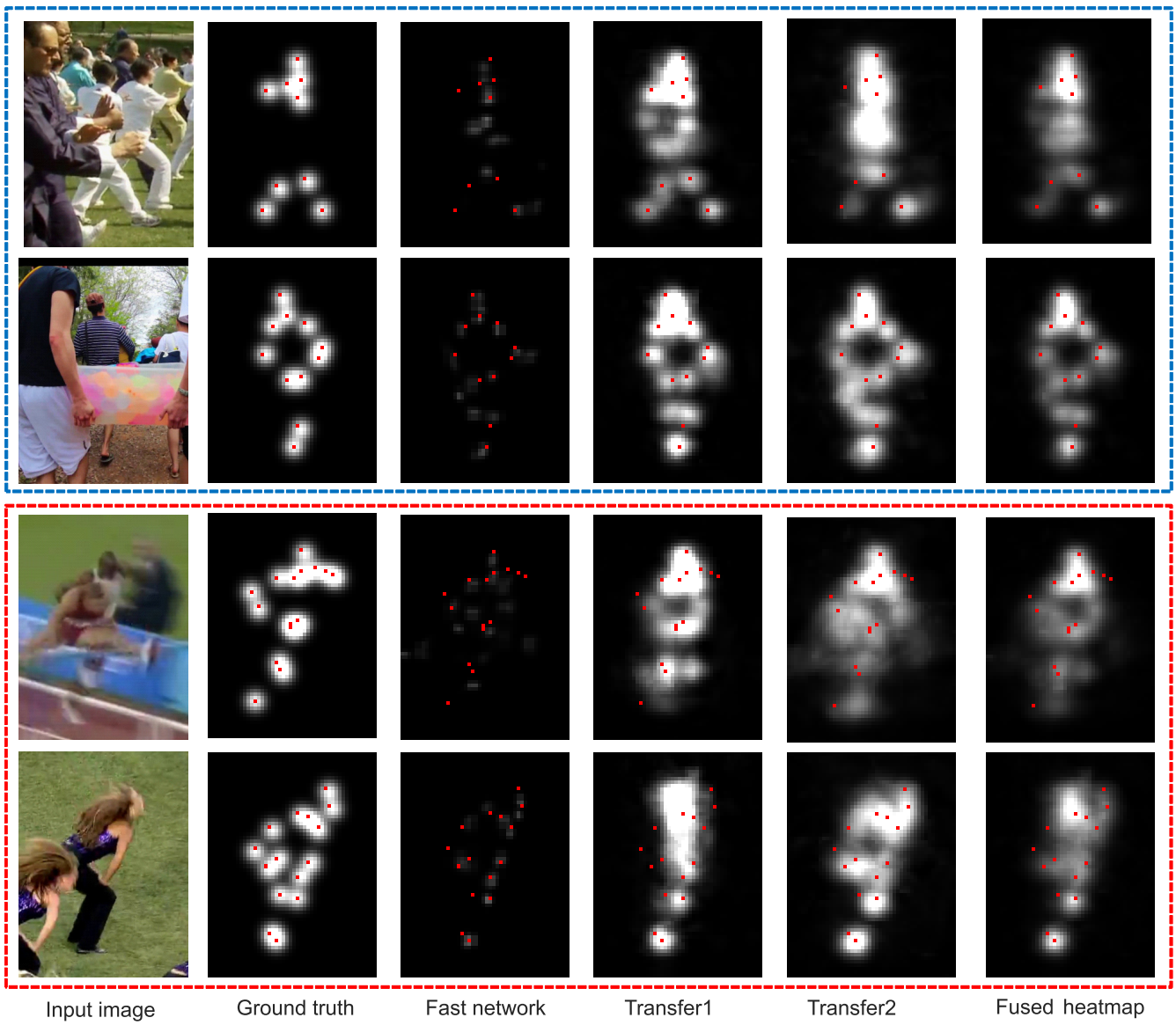


Fig. 7 Sample results from left to right: input images, heatmaps generated from ground truth, heatmaps from the fast network, heatmaps warped from the key frame via transfer1 network, heatmaps warped from the previous frame via transfer2 network, and fused heatmaps. Red points on the heatmaps are the locations of the ground truth. The top two rows inside the blue rectangle show successful examples, while the bottom two rows inside the red rectangle show some failures. Note that the heatmaps from the fast network are darker because they have lower peaks or flatter distributions.

4.4 Ablation Study

The first experiment is to check how much the accuracy drops when the frame set length is large and how effective our extension is to long-term frames. In Fig. 6, FSPose_base shows the mAP accuracy for each non-key frame with the distance from the key frame, where it is especially true that

Table 2 Comparison of the accuracy (in each keypoint and mean value) and computational complexity on the validation sets of PoseTrack 2017 and PoseTrack 2018. Heterogeneous denotes HRNet_W32 in key frames, MobileNetV2 in non-key frames, and transfer networks from key frame to non-key frame or between non-key frames. FSPose_base denotes the basic architecture of FSPose, FSPose_ext denotes the extension of FSPose_base, and the number (4, 8, 16, 32) denotes the frame number in a frame set. As a reference, the SOTA results from the original papers are listed with blue digits, the conditions of which may not be exactly the same as ours.

Method	Backbone	FLOPs(G)	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Mean
PoseTrack 2017 validation set										
Yang et al. [28]	HRNet + GNN	-	90.9	90.7	86.0	79.2	83.8	82.7	78.0	84.9
PoseNAS [17]	L18-C64	14.8	83.8	84.6	80.4	73.1	77.4	76.7	70.3	78.4
framewise [3]	HRNet_W32	7.65	83.07	89.85	83.93	75.59	82.15	80.83	73.74	81.43
framewise [6]	MobileNetV2	**0.49**	69.22	67.19	51.52	39.34	58.14	49.13	38.91	54.41
PoseWarper [4]	HRNet_W32	7.85	**84.41**	**90.56**	**84.71**	**77.13**	**83.09**	**82.33**	**77.27**	**82.89**
PoseWarper [4]	MobileNetV2	0.69	74.31	70.39	54.56	42.22	60.58	53.93	48.18	58.84
FSPose_base4	Heterogeneous	2.43	83.02	86.3	77.38	66.9	78.34	76.10	69.66	77.23
FSPose_base8	Heterogeneous	1.56	80.67	83.29	71.11	58.39	74.76	68.82	60.98	71.78
FSPose_base16	Heterogeneous	1.13	76.88	78.56	62.98	48.81	70.18	61.32	51.08	65.10
FSPose_base32	Heterogeneous	0.91	71.73	72.58	55.74	41.41	63.38	54.70	45.18	58.74
FSPose_ext8	Heterogeneous	1.61	81.47	82.98	71.29	59.71	74.72	69.76	62.83	72.47
FSPose_ext16	Heterogeneous	1.17	79.7	80.43	65.71	52.62	71.81	64.56	55.6	68.04
FSPose_ext32	Heterogeneous	0.96	76.45	76.22	60.57	46.74	66.99	58.8	50.78	63.30
PoseTrack 2018 validation set										
Yang et al. [28]	HRNet + GNN	-	85.1	87.7	85.3	80.0	81.1	81.6	77.2	82.7
framewise [3]	HRNet_W32	7.65	82.48	88.26	83.23	77.33	79.86	80.73	77.22	81.38
framewise [6]	MobileNetV2	**0.49**	70.60	68.81	53.98	43.68	59.56	52.90	44.22	57.21
PoseWarper [4]	HRNet_W32	7.85	**84.31**	**88.87**	**83.92**	**78.16**	**80.93**	**81.50**	**78.99**	**82.51**
PoseWarper [4]	MobileNetV2	0.69	75.06	71.94	56.72	46.49	62.11	57.41	53.23	61.40
FSPose_base4	Heterogeneous	2.43	82.72	86.14	79.21	72.44	77.66	77.39	74.80	78.90
FSPose_base8	Heterogeneous	1.56	80.80	83.41	74.03	65.41	74.58	72.20	67.95	74.50
FSPose_base16	Heterogeneous	1.13	77.37	78.98	66.91	56.95	69.91	65.75	59.38	68.52
FSPose_base32	Heterogeneous	0.91	73.12	73.81	60.06	49.60	64.49	59.97	53.35	62.79
FSPose_ext8	Heterogeneous	1.61	81.21	83.03	73.63	65.85	74.37	72.66	68.92	74.70
FSPose_ext16	Heterogeneous	1.17	79.58	80.80	69.32	60.00	71.69	68.71	63.42	71.11
FSPose_ext32	Heterogeneous	0.96	77.05	77.69	64.41	54.45	68.39	64.08	58.35	67.06

the mAP accuracy of the latter part of the non-key frames becomes lower. However, FSPose_ext, which is our extension to long-term frames, can effectively improve the accuracy of the latter part of the non-key frames. This experiment shows that an additional heatmap from the neighboring frame is effective to solve the temporal correlation decreasing problem in the latter part of the non-key frames.

The second experiment is to study the effect of each component in the proposed heterogeneous framework. We report the accuracy and computational complexity in non-key frames of PoseTrack 2017 dataset for fast networks and two transfer networks used in FSPose_ext16 depicted in Table 1, which shows that the two transfer networks have higher accuracy than the fast network. Therefore, the information from other frames (key frame and the neighboring non-key frame) is of high quality, which is helpful in pose estimation for the current frame. As shown in Table 1, this is especially true for Transfer1, which warps the heatmap from the key frame. In other words, given a non-key frame, the mean accuracy will be 54.30% if only using fast network with the computational cost of 0.49 GFLOPs. With less than half of computational cost, the Transfer1 network provides an even higher mean accuracy of 62.12%, which results in a better balance between accuracy and computational complexity.

The third experiment is to provide heatmap samples generated from the fast network, transfer networks, and fused results, as shown in Fig. 7. Note that the heatmap resolutions, which are defined by the network architectures, are so small that they look blurred in Fig. 7. This implies that slow motions (in the success cases) are easier to warp than fast motions (in the failure cases). In the case of failures, the warped heatmaps from both Transfer1 and Transfer2 are incorrect, which implies that the fast motions make transfer networks less effective. However, in the successful cases, the warped heatmaps from both Transfer1 and Transfer2 are correct. In addition, if we compare the fused heatmaps and heatmaps from each network, e.g., those heatmaps of feet, we can see that the peaks of fused heatmaps become closer to the ground truth. This means that the fused heatmaps improve the pose estimation accuracy.

4.5 Comparison with Existing Methods

We report the results of our method and other state-of-the-art methods on the PoseTrack 2017 and PoseTrack 2018 datasets in Table 2. In the heterogeneous framework of FSPose, the computational complexity is calculated as follows. For key frames, the computational complexity comes from only slow networks. For non-key frames, the computational complexity comes from fast networks and transfer networks. Then, we average the computational complexity

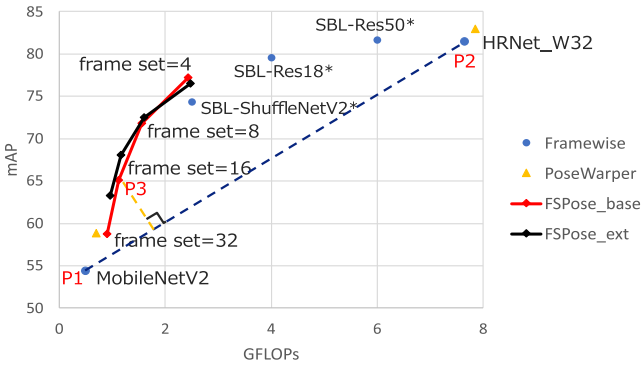


Fig. 8 Performance comparison between our methods (FSPose_base, FSPose_ext) and PoseWarper [4] (competitor method) and framewise methods (baseline) on PoseTrack 2017 validation set. Beside our results, we additionally plot three results directly from the existing paper [35], which are marked with asterisks.

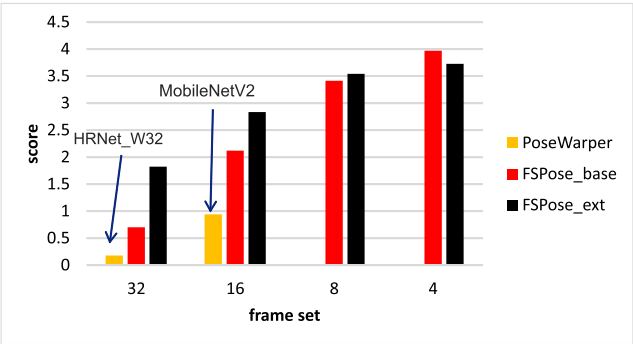


Fig. 9 Score comparison between our methods (FSPose_base, FSPose_ext) and the PoseWarper (competitor method) with the same backbones (MobileNetV2 and HRNet_W32) used in our methods on the PoseTrack 2017 validation set.

in all the frames. According to Table 2, compared to the framewise methods, PoseWarper achieves an accuracy gain of 1.5 points using an HRNet backbone or an accuracy gain of 4.4 points using a MobileNetV2 backbone for a computational cost of 0.2 GFLOPs using the transfer network. In contrast, FSPose achieves much greater gains compared to the framewise methods. For example, compared to the framewise method of MobileNetV2, FSPose_base16, which has 16 frames in a frame set using the basic architecture described in Sect. 3.3, achieves an accuracy gain of 10.7 points for a computational cost of 0.64 GFLOPs. Furthermore, FSPose_ext16, which has 16 frames in a frame set using the extension architecture described in Sect. 3.4 achieves an accuracy gain of 13.6 points for a computational cost of 0.68 GFLOPs.

For an intuitive comparison with the methods that use HRNet_W32 and MobileNetV2 as the backbones, Fig. 8 shows the results of framewise methods as a baseline, PoseWarper as a competitor method, and FSPose as a proposed method, where both FSPose_base and FSPose_ext are much farther away from the baseline than PoseWarper, i.e., they are locating the top-left area to PoseWarper. For FSPose_ext, which provides an additional transfer network warping the heatmap from the previous frame for the latter half of the non-key frames, the results are actually rather close to FSPose_base, although FSPose_ext effectively improves the accuracy, e.g., 4.6 points when there are 32 frames in a frame set. As shown in Table 2, the accuracy gain decreases as a frame set becomes shorter, i.e., the accuracy gains are 4.6 points to 2.9, 0.7, and −0.8 points when there are 32, 16, 8 and 4 frames in a frame set, respectively. Thus, FSPose_ext should not be applied to the case that there are less than 8 frames in a frame set, where the temporal correlation is strong enough. In the case of 4 frames in a frame set, the negative gain infers that warping the heatmaps of the previous frames may add noise in the fused heatmaps due to their relatively low quality compared to those in key frames.

As an objective metric, we also calculate the scores by Eq. (7). Note that the larger *score* is, the better the performance. As shown in Fig. 9, the scores of the proposed methods (FSPose_base and FSPose_ext) are more than twice as much as those of the competitor method (PoseWarper) with different backbones, where the highest score comes from FSPose_base4. In addition, compared with FSPose_base, FSPose_ext achieves higher scores if using the same length of a frame set. Moreover, the score difference is larger when the frame number is larger in a frame set.

5. Conclusion

We present a heterogeneous framework with transfer networks, fast and slow networks for different types of frames, referred to as *FSPose*, which is simple yet effectively achieves a good balance between accuracy and computational complexity. FSPose is based on the observation that the high-quality warped heatmap from slow networks helps improve the accuracy. In addition, there is only a minor increase in computational complexity in the transfer network. As a result, the heterogeneous framework is better in terms of the balance between accuracy and computational complexity. Furthermore, we extend the usage of long-term frames by solving the temporal correlation decreasing problem, where an additional heatmap from the neighboring frame is warped for the latter part of the non-key frames. The experimental results on PoseTrack 2017 and PoseTrack 2018 demonstrate that FSPose achieves more than twice as much as a competitor method in terms of scores.

In our future work, we will improve the balance between accuracy and computational complexity by introducing an adaptive frame set according to human motions. For example, the videos with fast motions should have a shorter frame set than those with slow motions. Another challenge is to import a third type of frame, where the heatmaps from both preceding and subsequent frames are warped.

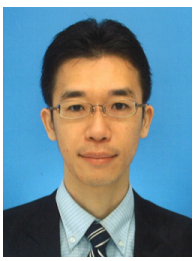
References

[1] Q. Dang, J. Yin, B. Wang, and W. Zheng, "Deep Learning based 2d Human Pose Estimation: A Survey," Tsinghua Science and Technology, vol.24, no.6, pp.663–676, 2019.

[2] D. Yang, R. Dai, Y. Wang, R. Mallick, L. Minciullo, G. Francesca, and F. Bremond, "Selective Spatio-temporal Aggregation Based Pose Refinement System: Towards Understanding Human Activities in Real-world Videos," Proc. IEEE/CVF Winter Conference on Applications of Computer Vision, pp.2363–2372, 2021.
[3] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep High-resolution Representation Learning for Human Pose Estimation," Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.5693–5703, 2019.
[4] G. Bertasius, C. Feichtenhofer, D. Tran, J. Shi, and L. Torresani, "Learning Temporal Pose Estimation from Sparsely-labeled Videos," Advances in Neural Information Processing Systems 32, pp.3027–3038, 2019.
[5] A. Howard, M. Sandler, B. Chen, W. Wang, L.-C. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, H. Adam, and Q. Le, "Searching for MobileNetV3," Proc. IEEE/CVF International Conference on Computer Vision, pp.1314–1324, 2019.
[6] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.4510–4520, 2018.
[7] M. Tan and Q. Le, "Efficientnet: Rethinking Model Scaling for Convolutional Neural Networks," International Conference on Machine Learning, pp.6105–6114, PMLR, 2019.
[8] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An Extremely Efficient Convolutional Neural Network for Mobile Devices," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.6848–6856, 2018.
[9] J. Wang, K. Qiu, H. Peng, J. Fu, and J. Zhu, "AI Coach: Deep Human Pose Estimation and Analysis for Personalized Athletic Training Assistance," Proc. 27th ACM International Conference on Multimedia, pp.2228–2230, 2019.
[10] U. Iqbal, A. Milan, and J. Gall, "PoseTrack: Joint Multi-person Pose Estimation and Tracking," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.4654–4663, 2017.
[11] M. Andriluka, U. Iqbal, E. Insafutdinov, L. Pishchulin, A. Milan, J. Gall, and B. Schiele, "PoseTrack: A Benchmark for Human Pose Estimation and Tracking," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.5167–5176, 2018.
[12] Y. Kawana and N. Ukita, "Occluded Appearance Modeling with Sample Weighting for Human Pose Estimation," IEICE Trans. Inf. & Syst., vol.E100-D, no.10, pp.2627–2634, 2017.
[13] N. Ukita, "Pose Estimation with Action Classification Using Global-and-pose Features and Fine-grained Action-specific Pose Models," IEICE Trans. Inf. & Syst., vol.E101-D, no.3, pp.758–766, 2018.
[14] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: Realtime Multi-person 2d Pose Estimation Using Part Affinity Fields," IEEE Trans. Pattern Anal. Mach. Intell., vol.43, no.1, pp.172–186, 2021.
[15] C. Wang, F. Zhang, and S.S. Ge, "A Comprehensive Survey on 2d Multi-person Pose Estimation Methods," Engineering Applications of Artificial Intelligence, vol.102, p.104260, 2021.
[16] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, and J. Sun, "Cascaded Pyramid Network for Multi-person Pose Estimation," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.7103–7112, 2018.
[17] Q. Bao, W. Liu, J. Hong, L. Duan, and T. Mei, "Pose-native Network Architecture Search for Multi-person Human Pose Estimation," Proc. 28th ACM International Conference on Multimedia, pp.592–600, 2020.
[18] X. Dai, I. Spasić, S. Chapman, and B. Meyer, "The State of the Art in Implementing Machine Learning for Mobile Apps: A Survey," 2020 SoutheastCon, pp.1–8, IEEE, 2020.
[19] E. Insafutdinov, M. Andriluka, L. Pishchulin, S. Tang, E. Levinkov, B. Andres, and B. Schiele, "Arttrack: Articulated Multi-person Tracking in the Wild," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.1293–1301, 2017.
[20] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol.9, no.8, pp.1735–1780, 1997.
[21] K. Cho, B. van Merriënboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations Using RNN Encoder-decoder for Statistical Machine Translation," EMNLP, pp.1724–1734, 2014.
[22] B. Artacho and A. Savakis, "Unipose: Unified Human Pose Estimation in Single Images and Videos," Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.7033–7042, 2020.
[23] Y. Luo, J. Ren, Z. Wang, W. Sun, J. Pan, J. Liu, J. Pang, and L. Lin, "LSTM Pose Machines," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.5207–5215, 2018.
[24] R. Girdhar, G. Gkioxari, L. Torresani, M. Paluri, and D. Tran, "Detect-and-track: Efficient Pose Estimation in Videos," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.350–359, 2018.
[25] L. Zhou, Y. Chen, J. Wang, and H. Lu, "Progressive Bi-c3d Pose Grammar for Human Pose Estimation," Proc. AAAI Conference on Artificial Intelligence, vol.34, no.7, pp.13033–13040, 2020.
[26] T. Pfister, J. Charles, and A. Zisserman, "Flowing ConvNets for Human Pose Estimation in Videos," Proc. IEEE International Conference on Computer Vision, pp.1913–1921, 2015.
[27] J. Song, L. Wang, L. Van Gool, and O. Hilliges, "Thin-slicing Network: A Deep Structured Model for Pose Estimation in Videos," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.5563–5572, 2017.
[28] Y. Yang, Z. Ren, H. Li, C. Zhou, X. Wang, and G. Hua, "Learning Dynamics via Graph Neural Networks for Human Pose Estimation and Tracking," Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.8070–8080, 2021.
[29] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable Convolutional Networks," Proc. IEEE International Conference on Computer Vision, pp.764–773, 2017.
[30] 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," European Conference on Computer Vision, vol.8693, pp.740–755, Springer, 2014.
[31] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. Gehler, and B. Schiele, "Deepcut: Joint Subset Partition and Labeling for Multi Person Pose Estimation," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.4929–4937, 2016.
[32] B. Xiao, H. Wu, and Y. Wei, "Simple Baselines for Human Pose Estimation and Tracking," Proc. European Conference on Computer Vision, vol.11210, pp.472–487, 2018.
[33] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2d Human Pose Estimation: New Benchmark and State of the Art Analysis," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.3686–3693, 2014.
[34] "Github: fvcore," https://github.com/facebookresearch/fvcore, 2021.
[35] L. Xu, Y. Guan, S. Jin, W. Liu, C. Qian, P. Luo, W. Ouyang, and X. Wang, "Vipnas: Efficient Video Pose Estimation via Neural Architecture Search," Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.16067–16076, 2021.

Jianfeng Xu received the B.S. (with honor) and the M.S. degrees from Tsinghua University, China, in 2001 and 2004, respectively, and the Ph.D. degree from the University of Tokyo, Japan, in 2007. He has been working at KDDI Research, Inc. since 2007 and now is a Core Researcher in Advanced Visual Communication Laboratory. His research interests include human motion analysis, sports analysis, deep learning technologies, and dynamic mesh coding technologies.

Satoshi Komorita received the B.E. and M.E. degrees from the University of Tokyo in 2004 and 2006, respectively. He joined KDDI Corporation in 2006 and engaged in mobile network research, IEEE Standardization, and smartphone development. He is currently the Group Leader in charge of 3D Space Transmission Laboratory in KDDI Research, Inc. His current research interests are human pose recognition and position estimation from images.



Kei Kawamura received his B.E., M.Sc., and Ph.D. degrees in Global Information and Telecommunication Studies from Waseda University, Japan, in 2004, 2005, and 2013, respectively. He joined KDDI in 2010. He has been involved with the development of HEVC and VVC standards under JCT-VC and JVET. He is currently engaged in the research and development of a video coding system at KDDI Research, Inc. His research interests include image and video processing, video coding, and multimedia distribution. He is a member of a steering committee of PCSJ/IMPS. He is a member of IEEE and a senior member of IEICE.