This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI

10.1109/ACCESS.2020.2998121, IEEE Access

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

**PosePropagationNet: Towards Accurate and Efficient Pose Estimation in Videos**

**YU LIU1, JIANSHENG CHEN2, (Senior Member, IEEE).**

1 School of Aerospace Engineering, Tsinghua University, Beijing 100084, China

* Department of Electronic Engineering, Tsinghua University, Beijing 100084, China

Corresponding author: Jiansheng Chen (email: jschenthu@mail.tsinghua.edu.cn).

This work was supported by the National Natural Science Foundation of China under Grant 61673234.

 **ABSTRACT** We rethink on the contradiction between accuracy and efficiency in the field of video poseestimation. Large networks are typically exploited in previous methods to pursue superior pose estimation results. However, those methods can hardly meet the low-latency requirement for real-time applications because of their computationally expensive nature. We present a novel architecture, PosePropagationNet (PPN), to generate poses across video frames accurately and efficiently. Instead of extracting temporal cues or knowledge someways to enforce geometric consistency as most of the previous methods do, we explicitly propagate well-estimated pose from the preceding frame to the current frame by leveraging pose propagation mechanism, endowing lightweight networks with the capability of performing accurate pose estimation in videos. The experiments on two large-scale benchmarks for video pose estimation show that our method significantly outperforms previous state-of-the-art methods in both accuracy and efficiency. Compared with the previous best method, our two representative configurations, PPN-Stable and PPN-Swift, achieve 2.5 and 6 FLOPs reduction respectively, as well as significant accuracy improvement.

 **INDEX TERMS** Network Efficiency, Pose Propagation Mechanism, Video Pose Estimation

**I. INTRODUCTION**

IDEO pose estimation aims at localizing human body Vjoints across video frames. It can be applied in many areas, such as human-computer interaction, computer ani-mation and video surveillance. Most of the research works on pose estimation focus on the single-image level, while less attention has been paid to video-based pose estimation mainly because of the limited number of large-scale anno-tated datasets. Compared with image-based pose estimation, video-based pose estimation is more challenging due to sev-eral inevitable troublesome factors, including motion blur, perspective change and scale variation.

Previous methods for video pose estimation task mostly rely on large networks to produce high-quality image repre-sentations, facilitating body joint localization at pixel level. Temporal cues are additionally extracted and leveraged to ensure temporal dependency, improving preliminary pose estimation results. As shown in Fig. 1(a), LSTM units are employed to transfer temporal knowledge as hidden states. Besides, optical flow is also widely exploited [1]–[3] as a strong temporal cue. Although these methods demonstrate applaudable experimental performances, most of them are proven to be computationally expensive, preventing them

from meeting the low-latency requirement for real-time ap-plications such as real-time surveillance and autonomous driving.

Lightweight networks are weak in producing satisfying single-image pose estimation results because of their rela-tively low representational capacity when no supplementary information is provided. However, in the video domain, consecutive frames share great geometric consistency, which makes it possible for lightweight networks to perform accu-rate pose estimation if temporal knowledge can be somehow transferred across frames to provide guidance. As shown in Fig. 1(b), temporal knowledge is distilled and transferred in the form of pose kernels, providing guidance for lightweight networks in joint localization. Based on this understanding, we take efficiency problem into consideration and propose a novel architecture, PosePropagationNet (PPN), to enhance the capability of lightweight networks in the field of video pose estimation.

The pipeline of our proposed end-to-end trainable PPN is shown in Fig. 1(c). Instead of bothering to generate temporal cues or knowledge in a learnable form, we directly propagate the pose estimated from the previous frame to the subsequent frame as explicit temporal guidance. The subsequent pose

|  |  |
| --- | --- |
| VOLUME 4, 2016 | 1 |

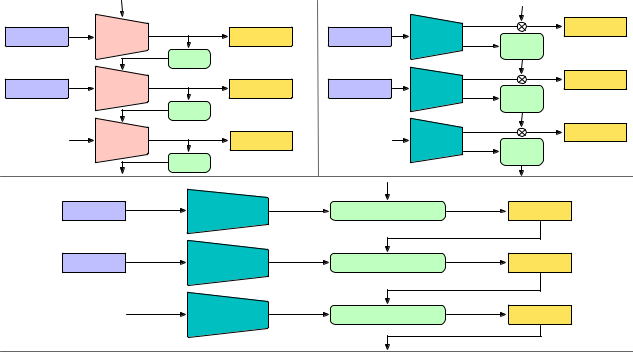
This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2998121, IEEE Access

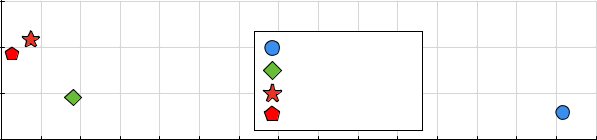


Y. Liu et al.: PosePropagationNet: Towards Accurate and Efficient Pose Estimation in Videos

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Frame *t* - 1** | **Large** | **Pose *t* - 1** |  | **Frame *t* - 1** | **Light** | **Pose *t* - 1** |  |
| **CNN** |  | **CNN** | **Pose** |  |
|  |  | **LSTM** |  |  |  | **Kernels** |  |
|  |  |  |  |  |  |  |
| **Frame *t*** | **Large** | **Pose *t*** |  | **Frame *t*** | **Light** | **Pose *t*** |  |
| **CNN** |  | **CNN** | **Pose** |  |
|  |  |  |  |  |
|  |  | **LSTM** |  |  |  | **Kernels** |  |
|  |  |  |  |  |  |  |
| **Frame *t* + 1** | **Large** | **Pose *t* + 1** |  | **Frame *t* + 1** | **Light** | **Pose *t* + 1** |  |
| **CNN** |  | **CNN** | **Pose** |  |
|  |  |  |  |  |
|  |  | **LSTM** |  |  |  | **Kernels** |  |
|  |  | **(a)** |  |  | **(b)** |  |
|  |  |  |  |  |  |
|  | **Frame *t* - 1** | **Light CNN** |  | **Pose Propagation Unit** | | **Pose *t* - 1** |  |
|  | **Frame *t*** | **Light CNN** |  | **Pose Propagation Unit** | | **Pose *t*** |  |
|  | **Frame *t* + 1** | **Light CNN** |  | **Pose Propagation Unit** | | **Pose *t* + 1** |  |



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **(c)** |  |
|  | 95 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **(%)** | 94 |  |  |  |  |  |  |  | LSTM PM (CVPR 2018) | | |  |  |  |  |  |  |
| ***torso*** |  |  |  |  |  |  |  |  | DKD (ICCV 2019) | |  |  |  |  |  |  |  |
| ***-*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| ***PCK*** | 93 |  |  |  |  |  |  |  | **PPN-Stable (Ours)** | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  | **PPN-Swift (Ours)** | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 92 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 | 55 | 60 | 65 | 70 | 75 |  |
|  | 0 |  |
|  |  |  |  |  |  |  |  | **FLOPs (G)** | |  |  |  |  |  |  | **(d)** |  |



**FIGURE 1.** Comparison of our method with the other two state-of-the-artmethods in video pose estimation. (a) Pipeline of the LSTM Pose Machines [4].

1. Pipeline of the Dynamic Kernel Distillation (DKD) network [5]. (c) The proposed pipeline which takes advantage of pose propagation mechanism, allowing lightweight networks to perform high-quality pose estimation in videos. We provide two representative configurations, PPN-Stable and PPN-Swift. Accuracy and computational efficiency of different methods are compared in (d). Evaluation is implemented on Penn Action Dataset with metric PCK-torso. The floating-point operations (FLOPs) is used to measure computational efficiency. Detailed numerical results are shown in Table. 5.

can be generated by transforming the previous pose accord-ing to joint motion offsets between the two frames. We imple-ment the pose propagation mechanism illustrated above with a specially designed module, namely the Pose Propagation Unit (PPU). As such, the process of localizing body joints is converted to pose propagation across frames, which is a less challenging task for lightweight networks. Compared with LSTM units [4] and pose kernels [5], our PPU carries explicit temporal guidance in a more computationally compact way, leading to dramatically FLOPs reduction while achieving significantly higher accuracy, as shown in Fig. 1(d). We evaluate our method on two widely used video pose estima-tion benchmarks, Penn Action Dataset [6] and Sub-JHMDB Dataset [7], obtaining state-of-the-art performances in both accuracy and efficiency.

Contributions of our work can be summarized as follows:

1. We propose a novel architecture, PosePropagationNet, for video pose estimation. Geometric consistency is guaranteed in the manner of pose propagation, facilitating the model to generate accurate and consistent pose estimation results in videos and achieve state-of-the-art accuracy on two ma-jor benchmarks. 2) Benefitting from the pose propagation mechanism we present, lightweight networks employed in PPN can perform pose estimation accurately and efficiently in videos. Significant FLOPs reduction over previous state-of-the-art methods allows our PPN to meet the low-latency requirement for real-time applications.

2

**II. RELATED WORK**

Human Pose Estimation in Images Early research works studying image-based single-person pose estimation are mostly based on pictorial structures [8]–[11], which model human body as a tree-structured graph. However, those methods naturally lack the ability to deal with complex occlusions. Most of the recent works take advantage of deep Convolutional Neural Network (CNN) and follow a regres-sion fashion: regressing joint coordinates [12] or regressing joint heatmaps [13]–[17]. These CNN-based methods either employ multi-stage architectures [13], [15] to recursively refine estimation results, or build strong backbones [14],

1. to efficiently extract high-level image representations, in order to achieve competitive performance on popular bench-marks [18], [19].

Human Pose Estimation in Videos Video pose estimation has attracted less attention compared with image-based pose estimation mainly because of the limited number of large-scale benchmarks in video domain. Existing research works focus on extracting temporal cues, such as optical flow [1]– [3], [20], to help refine framewise estimation results gen-erated by large networks. Song et al. [1] propose a deep spatio-temporal network, namely Thin-Slicing, which aligns joint heatmaps across frames based on dense optical flow computation. Recurrent architectures are exploited in [4],

1. to transfer temporal information in the form of hidden states. A large network is typically required to serve as im-age encoder, producing high-level image representations. 3D CNN is investigated in [22] to capture temporal dependency, facilitating multi-person pose estimation in videos. Nie et al. [5] propose a method that distills pose kernels and thus simplifies joint localization as a matching problem. We take the efficiency problem into consideration and explicitly prop-agate poses across frames as temporal guidance, allowing lightweight networks to perform accurate pose estimation in videos.

**III. METHODOLOGY**

As shown in Fig. 2(a), we build our PosePropagationNet (PPN) as a streamline architecture so that consecutive frames within a temporal range can be processed in a single-shot feed-forward manner. In the following, we first introduce the overall pipeline of our network and then go through the details of each component.

1. **OVERALL PIPELINE OF POSEPROPAGATIONNET**

Given a video sequence that contains T consecutive frames

* = fItgTt=1, where It 2 RH W 3 denotes the frame at time step t, we enable our proposed PPN to generate a set

of joint heatmaps H = fhtgTt=1, where ht 2 RHS WS K denotes the estimated joint heatmaps for frame It. We use

H and W to denote the height and width of frames, and use S and K to denote the total stride of the network and the number of joints, respectively. For frame It, the lightweight BodyNet takes charge of generating preliminary

^2 H W K

joint heatmaps ht R S S . Afterwards, together with

VOLUME 4, 2016

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

t=1

**B. FROM POSEWARPER TO POSE PROPAGATION UNIT** We get the inspiration of designing PPU from PoseWarper, which is proposed by Bertasius et al. [23] to solve the prob-lem of pose estimation in sparsely-annotated video datasets. Specifically, the relationship between two adjacent frames with opposite annotation status (one labeled, one unlabeled) is investigated. PoseWarper is designed to build that rela-tionship by estimating joint motion offsets between the two

VOLUME 4, 2016

X

^

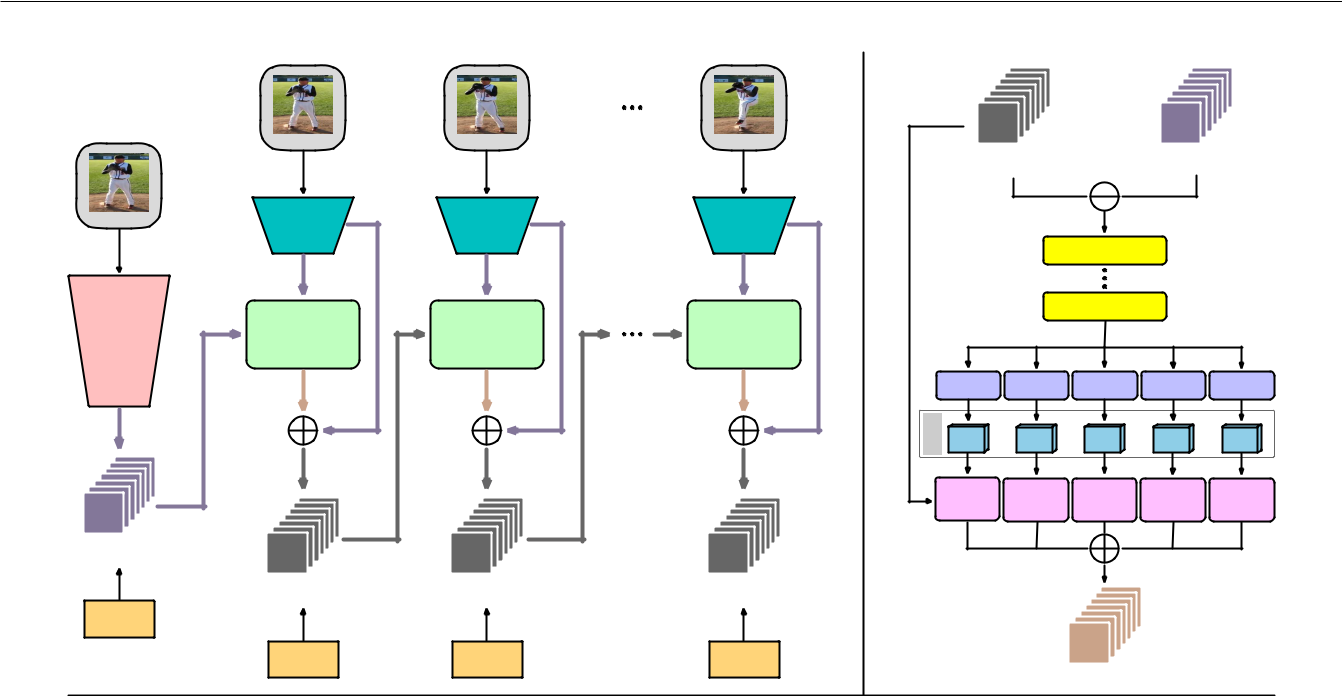
for the first frame, h1 and h1, are involved in loss computa-tion. Given the ground truth joint heatmaps gt for frame It, the loss is defined as the Mean Squared Error M SE( ) shown in Eq. 1.

T

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2998121, IEEE Access



Y. Liu et al.: PosePropagationNet: Towards Accurate and Efficient Pose Estimation in Videos



|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **Time *t*** |  | **Time *t +* 1** | |  |  |
|  | **Frame 1** | **Frame 2** | **Frame T** | **Final** |  | **Preliminary** | |  |
|  |  |  |  |  |  |
|  |  |  | **Joint Heatmaps** | |  | **Joint Heatmaps** | |  |
| **Frame 1** | **BodyNet** | **BodyNet** | **BodyNet** |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  | **Residual Bottleneck** | | |  |  |
|  | **Pose** | **Pose** | **Pose** | **Residual Bottleneck** | | |  |  |
|  |  |  |  |  |  |
| **HeadNet** | **Propagation** | **Propagation** | **Propagation** |  |  |  |  |  |
| **Unit** | **Unit** | **Unit** |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  | **conv2D** | **conv2D** | **conv2D** | **conv2D** | **conv2D** |  |
|  |  |  | **dilation=3** | **dilation=6** | **dilation=12** | **dilation=18** | **dilation=24** |  |
|  |  |  | **Offsets** |  |  |  |  |  |
|  |  |  | **Deform.** | **Deform.** | **Deform.** | **Deform.** | **Deform.** |  |
|  |  |  | **conv2D** | **conv2D** | **conv2D** | **conv2D** | **conv2D** |  |
|  |  |  | **dilation=3** | **dilation=6** | **dilation=12** | **dilation=18** | **dilation=24** |  |
| **Preliminary** |  |  |  |  |  |  |  |  |
| **Joint Heatmaps** |  |  |  |  |  |  |  |  |
|  | **Final** | **Final** | **Final** |  |  |  |  |  |
|  | **Joint Heatmaps** | **Joint Heatmaps** | **Joint Heatmaps** |  |  |  |  |  |
| **Loss** |  |  |  |  |  |  |  |  |
|  | **Loss** | **Loss** | **Loss** |  |  | **Time *t +* 1** |  |  |
|  |  | **Propagated** |  |  |  |
|  |  |  | **(a) (b)** |  |  |  |  |
|  |  |  |  | **Joint Heatmaps** | |  |  |

**FIGURE 2.** Network architecture of our proposed PosePropagationNet. (a) Overall pipeline of PPN. denotes elementwise addition. Network components with thesame color share weights throughout all time steps to help reduce parameter amounts. (b) The structure of Pose Propagation Unit, which is basically modified from PoseWarper [23]. denotes elementwise subtraction.

|  |  |  |
| --- | --- | --- |
| ^ | is fed |  |
| the joint heatmaps ht1 from the previous frame, ht |  |

into Pose Propagation Unit (PPU), which is able to prop-agate the previous pose to the current time step according to joint motion offsets between the two frames, outputting

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| H W | t | 2 R | H |  | W |  |  |
| the propagated joint heatmaps h |  |  |  | K . The final |  |
|  |  | S | S |  |

joint heatmaps ht 2 R S S K for frame It is computed by

^

combining ht and ht with elementwise addition. Since there is no predecessor for the first frame I1, we additionally design a HeadNet that is generally much larger than BodyNet, to

^

generate reliable initial joint heatmaps h1. In order to reduce parameter amounts in our network, BodyNets and PPUs throughout all time steps follow the weight-sharing principle.

Loss is computed on the produced final joint heatmaps ht across all frames. Note that the first frame appears twice in the feed-forward process, so both two sets of joint heatmaps

frames and performing pose estimation on the unlabeled frame by transforming the labeled pose according to the estimated offsets. We recognize the capability of PoseWarper to transfer labeled pose to adjacent unlabeled frames and build our PPU on the basis of PoseWarper architecture along with several significant modifications. In the following, we first mathematically formulate the pipeline of PoseWarper and then introduce the modifications we make.

Given labeled frame It and unlabeled frame It+1, Pose-Warper is trained to estimate poses for It+1 by transferring the labeled pose of It. Firstly, It+1 is fed into an image-based pose estimation network, outputting preliminary joint

^

heatmaps ht+1. On the other side, the joint heatmaps ht for

frame It can be obtained from ground truth. Afterwards, the

^

difference between ht and ht+1 is computed as

|  |  |  |  |
| --- | --- | --- | --- |
| ^ | ht | (2) |  |
| t;t+1 = ht+1 |  |

|  |  |  |
| --- | --- | --- |
| ^ | ; g1) + M SE(ht; gt) |  |
| L = M SE(h1 |  |

and fed into a stack of convolution blocks ( ). The output feature maps are then fed into a set of convolution layers with different dilation rates C(d)( ) to generate a set of offset

1. tensors, namely

|  |  |
| --- | --- |
| ot;t(d)+1 = C(d)( ( t;t+1)) d 2 D; | (3) |

where o(t;td)+1 denotes the estimated joint motion offset tensor between time step t and t + 1 with dilation rate d and D is an ensemble of different dilation rate values. Finally, those offset tensors are used to transform joint heatmaps ht via deformable convolution layers [24] D(d)( ) as

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| (d) | = D | (d) | (d) | (4) |  |
| ht+1 |  | (ht; ot;t+1) d 2 D; |  |
|  |  |  |  | 3 |  |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2998121, IEEE Access



Y. Liu et al.: PosePropagationNet: Towards Accurate and Efficient Pose Estimation in Videos

(d)

where ht+1 denotes the transformed joint heatmaps for It+1

with dilation rate d. Joint heatmaps produced with different dilation rate d are summed up as the final transformed joint heatmaps for It+1, namely,

X (d) 2

ht+1 = ht+1 d D: (5)

d

^

Conventionally, preliminary joint heatmaps ht+1 can be viewed as the final pose estimation result of It+1 following a single-image manner regardless of temporal dependency. In PoseWarper, geometric consistency is taken into considera-tion in the process of pose transferring and transformation, leading to a better pose estimation result for It+1. We design our PPU on the basis of PoseWarper. As shown in Fig. 2(b), PPU takes the estimated joint heatmaps from the previous frame and preliminary joint heatmaps of the current frame generated by BodyNet as inputs, producing the propagated

joint heatmaps ht+1 based on pose propagation mechanism illustrated above. The modifications we make are mainly in three folds:

1. We unify the pose propagation path during training and evaluation phases. As shown in Fig. 3(a), for PoseWarper, frames are sparsely annotated. During the training phase, poses are transferred from unlabeled frames to the labeled frame to meet supervision. On the contrary, during the eval-uation phase, the labeled pose is reversely transferred to unlabeled frames to perform dense pose estimation. In our architecture, the training path illustrated above fails to fit our HeadNet-BodyNet configuration, as it is somewhat counter-intuitive to refine a better pose by transforming a worse one, which intrinsically increases the difficulty of training. One possible pipeline for unify training path and evaluation path is shown in Fig. 3(b). Benefitting from densely-annotated datasets, poses can be propagated along that path to meet supervision at each time step during the training phase.
2. We modify the cascade scheme across frames. The pipeline shown in Fig. 3(b) propagates the high-quality pose from time step t to several subsequent time steps respectively. It perfectly corresponds to our HeadNet-BodyNet configura-tion, as the high-quality pose remains undamaged throughout the propagation process. However, it can be expected that the above pipeline would perform poorly when applied to long-range video sequences, since poses from temporally distant frames can hardly provide any useful guidance to the current frame in videos containing complicate human motions. We design our pose propagation path by building a connection between each neighboring frame pair, as shown in Fig. 3(c). In such a pipeline, poses are iteratively propagated from the previous frame to the current frame, ensuring the validity of the information flow. Therefore, our method is expected to be more scalable, and is capable of dealing with video sequences with different frame ranges, meeting various requirements in

real applications.

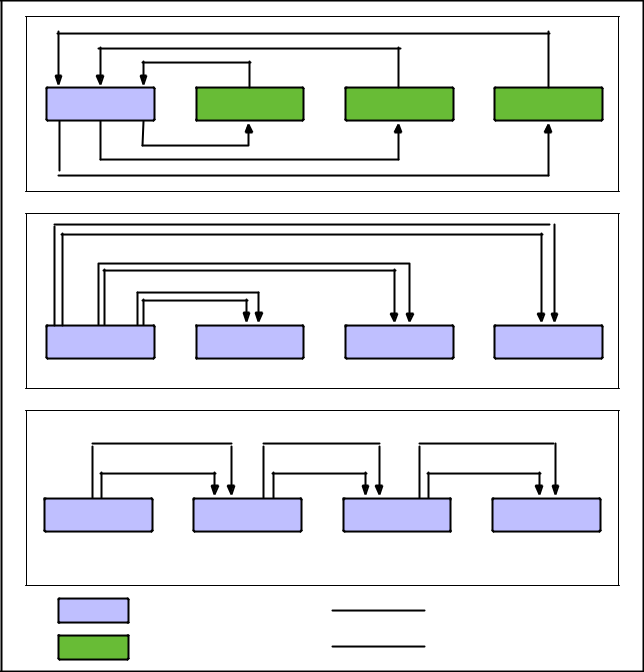
3) Instead of treating the propagated joint heatmaps ht

as the final joint heatmaps for It, we further fuse them

^

with the preliminary joint heatmaps ht via skip connection.

**Frame *t*** **Frame *t + 1*** **Frame *t + 2*** **Frame *t + 3***



**(a)**

**Frame *t*** **Frame *t + 1*** **Frame *t + 2*** **Frame *t + 3***

**(b)**

**Frame *t*** **Frame *t + 1*** **Frame *t + 2*** **Frame *t + 3***

**(c)**

**Labeled Frame** **Training Path**

**Unlabeled Frame** **Evaluation Path**

**FIGURE 3.** Comparison of different pose propagation paths. (a) Posepropagation path in PoseWarper. (b) One possible midway modification. (c) Pose propagation path in our PPN. Details can be viewed in text.

In our architecture, HeadNet, BodyNet and PPU can be

simultaneously trained. The propagated joint heatmaps ht are generated by transforming joint heatmaps from the pre-

vious frame ht1 via deformable convolution. The prelimi-

^

nary joint heatmaps ht generated by BodyNet are somehow vanished in that course and thus not directly involved in loss computation, which prevents BodyNet from receiving sufficient training. To solve the problem, we perform identity

^

mapping for ht and combine it with the propagated joint

heatmaps ht via elementwise addition. Following that fash-

^

ion, the preliminary joint heatmaps ht are explicitly involved in loss computation, facilitating the effective training of BodyNet.

**C. HEADNET AND BODYNET**

We employ two pose estimation networks for different time steps, namely HeadNet and BodyNet. HeadNet is responsible for performing pose estimation on the first frame. Generally speaking, the quality of initial pose decides the overall level of pose estimation results in that video sequence. Therefore, large networks are typically employed as HeadNet to guar-antee high performance. Afterwards, BodyNet takes charge of generating preliminary pose for each frame. Since pose propagation mechanism brings geometric knowledge from the previous frame to the current frame, BodyNet can be much more lightweight.

**IV. EXPERIMENTS**

4 VOLUME 4, 2016

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2998121, IEEE Access



Y. Liu et al.: PosePropagationNet: Towards Accurate and Efficient Pose Estimation in Videos

**A. DATASETS**

Penn Action Dataset. Penn Action Dataset [6] contains 2326 video sequences of 15 different actions, where 1258 clips are used for training and 1068 clips are used for testing. The number of frames varies among different video sequences. The 2D locations and visibility of totally 13 body joints are annotated for each frame, including head, shoulders, elbows, wrists, hips, knees and ankles. During testing, only visible joints are involved in evaluation.

Sub-JHMDB Dataset. JHMDB [7] is another dataset for video-based pose estimation. For fair comparison with pre-vious works, only a subset of JHMDB is used in our ex-periments, which is named as Sub-JHMDB. Sub-JHMDB consists of 316 video clips with 11200 frames in total. In Sub-JHMDB, only complete human bodies are involved and totally 15 body joints are annotated for each human instance. There are three split schemes for Sub-JHMDB and the split ratio of training and testing samples is roughly 3:1. Following previous works [1], [4], [5], we train and evaluate our method separately and report the average result over the three splits.

**B. IMPLEMENTATION DETAILS**

Data Augmentation. We perform data augmentation strate-gies following previous works [4], [5], including random scaling ([0.8, 1,4]), random rotation ([40 ; 40 ]) and ran-dom flipping. On account of sequential input, the trans-formation remains consistent across frames within a video sequence. All the frames are cropped based on the center and scale of the person instance and padded to a fixed size (256

256) as input.

Experiment Settings. Following previous works [4], [5], we pretrain all image-based pose estimation networks ex-ploited in our experiments on MPII dataset [18]. The frame range T of each sample is set as 5. Deconvolution lay-ers used in our experiments follow the settings illustrated in [16]. Adam optimizer [25] is adopted with 105 weight decay, and the learning rate is decreased linearly from 104 to 0. We set the batch size as 8 and train our network for 300k iterations. During evaluation phase, seven scales f0:8; 0:9; 1:0; 1:1; 1:2; 1:3; 1:4g are used for multi-scale in-ference.

Evaluation Metrics. We adopt the PCK metric proposed in [11] to evaluate our pose estimation results. In PCK, a joint is considered as being correctly localized if it falls within a predefined threshold L, where is a controlling coefficient and is conventionally set to 0.2. L is the reference distance, which is set as L = max(H; W ) in [1], [4], where H and W denote the height and width of bounding box of the person instance. However, since the scale of person is large, this metric has been considered to be too loose to differentiate different methods. Following [5], [26], we additionally adopt the definition of reference distance L as torso diameter, which is defined as the distance between left shoulder and right hip of ground-truth skeleton [26]. To avoid ambiguity, we term the above two metrics as PCK-body and PCK-torso respectively.

**C. ABLATION STUDIES**

We perform ablation studies to verify the effectiveness of our proposed PPN from two aspects. On the one hand, PPN can largely improve the performance of existing image-based pose estimation networks in video domain by introducing pose propagation mechanism. On the other hand, PPN en-dows lightweight networks with the capability of perform-ing accurate pose estimation by explicitly propagating high-quality pose generated from a large network forward across frames.

Firstly, we investigate one of the state-of-the-art image-based pose estimation networks, Simple Baseline [16], which follows a high-to-low-to-high pipeline that first extracts high-level low-resolution feature maps with ResNet family [27] and then raises the resolution back to a decent level via 2-strided deconvolution layers. Specifically, we vary the backbone of Simple Baseline models among ResNet-x; x 2 f18; 34; 50; 101g and evaluate each configuration on Penn Action Dataset following the single-image framewise manner as baselines, which are denoted as Framewise (ResNet-x) in Table. 1. Following original settings in [16], three 2-strided 4 4 deconvolution layers are appended to recover resolution. For comparison, we adopt Simple Baseline models as both HeadNet and BodyNet in our PPN. We use PPN (ResNet-x) in Table. 1 to denote our proposed network with ResNet-x as the backbone of HeadNet and BodyNet.

It can be observed from Table. 1 that PPN improves single-image framewise pose estimation results by a large margin. The improvement appears more obvious on evaluation metric PCK-torso, since results on PCK-body tend to be somewhat saturated. We can find that by introducing temporal pose

**TABLE 1.** Comparison of pose estimation results with and without exploitingpose propagation mechanism on Penn Action Dataset. Evaluation results on both PCK-body and PCK-torso metrics are reported. Better results are highlighted in Bold.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Backbone | Head Sho | | Elb | Wri | Hip | Knee Ank | | Mean |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | PCK-body | |
| Framewise (ResNet-18) | 98.7 | 97.8 | 95.3 | 94.6 | 97.1 | 96.1 | 95.7 | 96.3 |
| PPN (ResNet-18) | 99.1 | 98.5 | 96.5 | 95.7 | 98.1 | 97.1 | 97.5 | 97.6 |
| Framewise (ResNet-34) | 99.0 | 98.1 | 96.1 | 95.6 | 97.8 | 96.9 | 96.3 | 97.0 |
| PPN (ResNet-34) | 99.3 | 98.8 | 97.0 | 96.1 | 98.3 | 97.4 | 98.1 | 98.1 |
| Framewise (ResNet-50) | 99.1 | 98.4 | 97.3 | 96.8 | 98.1 | 97.8 | 97.3 | 97.7 |
| PPN (ResNet-50) | 99.2 | 98.9 | 97.8 | 97.4 | 98.7 | 98.5 | 98.3 | 98.7 |
| Framewise (ResNet-101) | 99.5 | 99.1 | 98.3 | 98.1 | 99.1 | 98.7 | 98.5 | 98.7 |
| PPN (ResNet-101) | 99.4 | 99.3 | 98.5 | 98.4 | 99.2 | 98.6 | 98.7 | 98.9 |
|  |  |  |  |  |  |  | PCK-torso | |
| Framewise (ResNet-18) | 94.7 | 89.8 | 89.8 | 88.4 | 83.3 | 88.8 | 89.0 | 88.7 |
| PPN (ResNet-18) | 95.4 | 92.8 | 93.0 | 91.7 | 87.1 | 92.5 | 92.3 | 92.1 |
| Framewise (ResNet-34) | 95.5 | 91.5 | 91.4 | 90.2 | 85.2 | 90.6 | 90.7 | 90.4 |
| PPN (ResNet-34) | 95.8 | 93.6 | 93.7 | 92.7 | 88.1 | 93.9 | 93.4 | 93.3 |
| Framewise (ResNet-50) | 95.6 | 93.1 | 93.4 | 92.3 | 86.9 | 93.0 | 92.5 | 92.2 |
| PPN (ResNet-50) | 95.9 | 95.3 | 94.4 | 93.5 | 89.5 | 95.4 | 94.8 | 94.1 |
| Framewise (ResNet-101) | 96.2 | 95.0 | 95.4 | 94.1 | 89.5 | 95.0 | 94.6 | 94.1 |
| PPN (ResNet-101) | 96.1 | 95.6 | 95.1 | 94.2 | 91.1 | 95.9 | 95.2 | 95.0 |
|  |  |  |  |  |  |  |  |  |

|  |  |
| --- | --- |
| VOLUME 4, 2016 | 5 |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2998121, IEEE Access



Y. Liu et al.: PosePropagationNet: Towards Accurate and Efficient Pose Estimation in Videos

**TABLE 2.** Ablation studies on Penn Action Dataset. Best results arehighlighted in Bold.

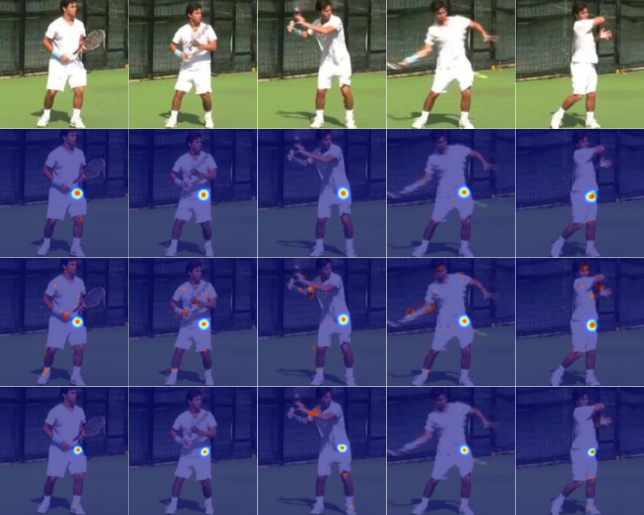
|  |  |  |  |
| --- | --- | --- | --- |
| Method | FLOPs (G) | PCK-body PCK-torso | |
|  |  |  |  |
| Framewise (ResNet-18-w-Deconv) | 7.74 | 96.3 | 88.7 |
| Framewise (ResNet-18-w-DUC) | 3.07 | 96.2 | 88.3 |
| PPN (ResNet-18-w-DUC)-w/o-SC | 3.87 | 97.9 | 93.2 |
| PPN (ResNet-18-w-DUC) | 3.87 | 98.8 | 94.2 |
|  |  |  |  |
| Framewise (MobileNet-V2-w-Deconv) | 5.64 | 95.8 | 87.9 |
| Framewise (MobileNet-V2-w-DUC) | 0.59 | 95.7 | 87.7 |
| PPN (MobileNet-V2-w-DUC)-w/o-SC | 1.39 | 97.5 | 92.9 |
| PPN (MobileNet-V2-w-DUC) | 1.39 | 98.5 | 93.8 |
|  |  |  |  |
|  |  |  |  |

propagation mechanism, PPN lifts the accuracy of pose es-timation by 0.90% on PCK-body metric and 2.28% on PCK-torso metric in average. The performance of PPN with a relatively smaller backbone, ResNet-18, even significantly surpasses the level of single-image framewise pose esti-mation results with a larger backbone, ResNet-34 (92.1% versus 90.4% on PCK-torso). The above results convincingly verify the effectiveness of our proposed Pose Propagation Unit for providing temporal guidance to refine single-image framewise pose estimation results.

Furthermore, we investigate the potential of lightweight networks for performing accurate pose estimation in videos by enforcing pose propagation mechanism. From Table. 1, we can find that lightweight networks alone are weak in producing satisfying pose estimation results. For example, Framewise (ResNet-18) achieves merely 88.7% accuracy on PCK-torso. We realize that deconvolution operation can be especially computationally intensive if applied on feature maps with large spatial size during the upsampling phase. Taking Framewise (ResNet-18-w-Deconv) shown in Table. 2 as baseline, we implement an ablation study to further re-duce network computational intensity and enhance network capability at the same time. On the one hand, instead of using expensive deconvolution layers, we investigate the usage of Dense Upsampling Convolution (DUC) layer that is proposed in [28] to implement 2 upsampling. As shown in the 1st and 2nd rows of Table. 2, by replacing deconvo-lution layers with DUC layers, we achieve over 2 FLOPs reduction with minor accuracy decrease. On the other hand, in order to introduce pose propagation mechanism, we adopt the state-of-the-art architecture on MPII benchmark [18], HRNet-W48 [14], as our HeadNet to generate high-quality initial pose for better performance, which is denoted as PPN (ResNet-18-w-DUC) in Table. 2. It can be observed from Table. 2 that despite of its weak performance on single-image level, the capability of lightweight network ResNet-18-w-DUC in video domain is dramatically boosted by propagat-ing high-quality pose generated by strong HeadNet across frames.

To further verify the capability of PPN to facilitate lightweight networks to perform accurate pose estimation in

**FIGURE 4.** Comparison of left hip heatmaps generated by BodyNet, PPU and



their combination respectively. The 2nd and 3rd rows are produced by PPN

(ResNet-18-w-DUC)-w/o-SC, while the 4th row is produced by PPN

(ResNet-18-w-DUC).

**TABLE 3.** Comparison of experimental results with metricPCK-torsounderdifferent frame range T on Penn Action Dataset.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| T | 1 | 2 | 5 | 10 | 15 |
|  |  |  |  |  |  |
| PCK-torso | 95.0 | 94.7 | 94.2 | 92.9 | 92.0 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

videos, we experiment with another smaller backbone for Bo-dyNet, MobileNet-V2 [29]. The effectiveness of MobileNet family is broadly evaluated in the field of image classifica-tion, object detection and semantic segmentation. As shown in Table. 2, we use Framewise (MobileNet-V2-w-Deconv) to denote single-image pose estimation with MobileNet-V2 as backbone and deconvolution layers as upsample unit. Like-wise, we replace deconvolution layers with DUC layers to perform single-image framewise pose estimation and denote it as Framewise (MobileNet-V2-w-DUC). Significant FLOPs

**TABLE 4.** Comparison with the method proposed by Nieet al.[5] on PennAction Dataset with evaluation metric PCK-torso. Experimental results related to DKD are borrowed from [5]. Better results are highlighted in Bold.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Method | FLOPs (G) | Head | Sho | Elb | Wri | Hip | Knee Ank | | Mean |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  | PCK-torso | |
| Framewise (ResNet-18) | 4.28 | 94.7 | 86.0 | 87.7 | 84.6 | 81.1 | 87.4 | 84.3 | 86.1 |
| DKD (ResNet-18) | 5.27 | 95.7 | 90.0 | 92.2 | 89.4 | 86.8 | 92.3 | 89.5 | 90.6 |
| PPN (ResNet-18) | 4.48 | 95.6 | 92.1 | 92.7 | 91.2 | 86.9 | 92.4 | 91.8 | 91.9 |
| Framewise (ResNet-34) | 6.69 | 95.8 | 88.7 | 88.5 | 86.7 | 83.6 | 89.6 | 85.3 | 87.3 |
| DKD (ResNet-34) | 7.68 | 96.4 | 91.9 | 93.0 | 90.8 | 88.6 | 93.5 | 91.9 | 92.1 |
| PPN (ResNet-34) | 6.89 | 95.8 | 93.3 | 93.5 | 92.4 | 88.1 | 93.8 | 93.2 | 93.1 |
| Framewise (ResNet-50) | 7.66 | 96.0 | 90.5 | 89.4 | 87.6 | 83.8 | 89.7 | 86.0 | 88.8 |
| DKD (ResNet-50) | 8.65 | 96.6 | 93.7 | 92.9 | 91.2 | 88.8 | 94.3 | 93.7 | 92.9 |
| PPN (ResNet-50) | 7.86 | 96.1 | 95.1 | 94.3 | 93.5 | 90.0 | 95.2 | 94.3 | 94.0 |
|  |  |  |  |  |  |  |  |  |  |

6 VOLUME 4, 2016

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2998121, IEEE Access



Y. Liu et al.: PosePropagationNet: Towards Accurate and Efficient Pose Estimation in Videos

**TABLE 5.** Comparison with state-of-the-art methods on Penn Action Dataset. Evaluation results on bothPCK-bodyandPCK-torsometrics are reported. Besides,the general network architecture of each method and FLOPs are reported as well. Best results are highlighted in Bold.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Method | Backbone (t = 1) | Backbone (t > 1) | Upsample Unit | FLOPs (G) | Head | Sho | Elb | Wri | Hip | Knee | Ank | Mean |
|  |  |  |  |  |  |  |  |  |  |  |  | |
|  |  |  |  |  |  |  |  |  |  |  | PCK-body | |
| Park et al. [30] | - | - | - | - | 62.8 | 52.0 | 32.3 | 23.3 | 53.3 | 50.2 | 43.0 | 45.3 |
| Nie et al. [31] | - | - | - | - | 64.2 | 55.4 | 33.8 | 24.4 | 56.4 | 54.1 | 48.0 | 48.0 |
| Gkioxari et al. [21] | 6 Conv | 6 Conv | 2 Deconv | - | 95.6 | 93.8 | 90.4 | 90.7 | 91.8 | 90.8 | 91.5 | 91.8 |
| Iqbal et al. [32] | VGG-16 [33] | VGG-16 | - | - | 89.1 | 86.4 | 73.9 | 73.0 | 85.3 | 79.9 | 80.3 | 81.1 |
| Song et al. [1] | CPM [15] | CPM | - | - | 98.0 | 97.3 | 95.1 | 94.7 | 97.1 | 97.1 | 96.9 | 96.5 |
| Luo et al. [4] | CPM | CPM | - | 70.98 | 98.9 | 98.6 | 96.6 | 96.6 | 98.2 | 98.2 | 97.5 | 97.7 |
| Nie et al. [5] | ResNet-101 | ResNet-50 | 2 Deconv | 8.65 | 98.8 | 98.7 | 96.8 | 97.0 | 98.2 | 98.1 | 97.2 | 97.8 |
| PPN-Stable | HRNet-W48 | ResNet-18 | 3 DUC | 3.87 | 99.0 | 99.3 | 98.5 | 98.3 | 98.8 | 98.8 | 98.7 | 98.8 |
| PPN-Swift | HRNet-W48 | MobileNet-V2 | 3 DUC | 1.39 | 98.9 | 99.2 | 98.4 | 97.9 | 98.4 | 98.6 | 98.5 | 98.5 |
|  |  |  |  |  |  |  |  |  |  |  | PCK-torso | |
| Luo et al. [4] | CPM | CPM | - | 70.98 | 96.0 | 93.6 | 92.4 | 91.1 | 88.3 | 94.2 | 93.5 | 92.6 |
| Nie et al. [5] | ResNet-101 | ResNet-50 | 2 Deconv | 8.65 | 96.6 | 93.7 | 92.9 | 91.2 | 88.8 | 94.3 | 93.7 | 92.9 |
| PPN-Stable | HRNet-W48 | ResNet-18 | 3 DUC | 3.87 | 96.1 | 95.2 | 94.8 | 93.9 | 89.1 | 95.4 | 95.2 | 94.2 |
| PPN-Swift | HRNet-W48 | MobileNet-V2 | 3 DUC | 1.39 | 96.1 | 95.2 | 94.4 | 93.3 | 88.8 | 94.9 | 94.9 | 93.8 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |



(a) (b)

**FIGURE 5.** Qualitative results on (a) Penn Action Dataset and (b) Sub-JHMDB Dataset. Best viewed in color.

reduction can be observed following that setting, which is down to no more than 0.6G. Then we treat that tiny network as BodyNet and employ HRNet-W48 as HeadNet to consti-tute our PPN, which is denoted as PPN (MobileNet-V2-w-DUC). Compared with PPN (ResNet-18-w-DUC), dramatic FLOPs reduction can be witnessed, while high performance is still maintained.

By comparing the 3rd and 4th rows, as well as 7th and 8th rows of Table. 2, we demonstrate the necessity of skip connection (SC) in PPN that fuses the propagated joint heatmaps with the preliminary joint heatmaps. Additionally, we visualize the joint heatmaps of left hip generated by BodyNet (2nd row), PPU (3rd row) and their combination via skip connection (4th row) in Fig. 4. It can be observed that the propagated joint heatmap of left hip is noisy with plenty of false positive points with relatively high response values. The preliminary joint heatmap generated by lightweight BodyNet

VOLUME 4, 2016

is relatively clean, while high responses fall in a large region around the precise location of left hip. With skip connection, the final joint heatmap is somewhat clean with high responses compactly aggregated.

Finally, we specially verify the scalability of our method to adaptively perform pose estimation for video sequences with different frame range T. To better simulate real application scenes, we directly apply our representative model, PPN (ResNet-18-w-DUC) that is trained with T = 5, to testing samples with different frame range T 2 f1; 2; 5; 10; 15g. As shown in Table. 3, without being specially trained, our network still maintains a competitive performance within long frame ranges.

Based on the experimental results shown above, we adopt PPN (ResNet-18-w-DUC) and PPN (MobileNet-V2-w-DUC) as two major configurations in our experiments that are capable of generating poses across video frames

7

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2998121, IEEE Access



Y. Liu et al.: PosePropagationNet: Towards Accurate and Efficient Pose Estimation in Videos

**TABLE 6.** Comparison with state-of-the-art results on Sub-JHMDB Dataset.

Best results are highlighted in Bold.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Method | Head | Sho | Elb | Wri | Hip | Knee | Ank | Mean |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | PCK-body | |
| Park et al. [30] | 79.0 | 60.3 | 28.7 | 16.0 | 74.8 | 59.2 | 49.3 | 52.5 |
| Nie et al. [31] | 80.3 | 63.5 | 32.5 | 21.6 | 76.3 | 62.7 | 53.1 | 55.7 |
| Iqbal et al. [32] | 90.3 | 76.9 | 59.3 | 55.0 | 85.9 | 76.4 | 73.0 | 73.8 |
| Song et al. [1] | 97.1 | 95.7 | 87.5 | 81.6 | 98.0 | 92.7 | 89.8 | 92.1 |
| Luo et al. [4] | 98.2 | 96.5 | 89.6 | 86.0 | 98.7 | 95.6 | 90.9 | 93.6 |
| Nie et al. [5] | 98.3 | 96.6 | 90.4 | 87.1 | 99.1 | 96.0 | 92.9 | 94.0 |
|  |  |  |  |  |  |  |  |  |
| PPN-Stable | 99.0 | 98.3 | 92.5 | 90.9 | 99.4 | 98.3 | 95.0 | 96.4 |
| PPN-Swift | 98.7 | 98.0 | 91.8 | 90.7 | 99.1 | 98.2 | 94.5 | 95.9 |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | PCK-torso | |
| Luo et al. [4] | 92.7 | 75.6 | 66.8 | 64.8 | 78.0 | 73.1 | 73.3 | 73.6 |
| Nie et al. [5] | 94.4 | 78.9 | 69.8 | 67.6 | 81.8 | 79.0 | 78.8 | 77.4 |
|  |  |  |  |  |  |  |  |  |
| PPN-Stable | 95.7 | 83.3 | 71.7 | 70.9 | 84.0 | 83.4 | 81.8 | 81.3 |
| PPN-Swift | 95.1 | 82.8 | 71.3 | 70.2 | 83.5 | 83.0 | 81.1 | 80.4 |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

accurately and efficiently, and term them as PPN-Stable and PPN-Swift respectively for simplicity.

**D. COMPARISON WITH STATE-OF-THE-ART METHODS**

To verify the superiority of our method, we compare our PPN with the previous state-of-the-art, which is the Dynamic Kernel Distillation (DKD) network proposed in [5], under the same settings. Specifically, in DKD, a large pose initializer is designed to generate initial pose and the following frame encoders for feature extraction are much smaller, which is fairly similar to our HeadNet-BodyNet configuration. Mod-ified from Simple Baseline [16] models, the pose initializer and frame encoder used in DKD both follow a high-to-low-to-high pipeline, where ResNet family is exploited to encode image representations and two 2-strided 4 4 deconvolution layers are appended to perform upsampling. Therefore, the total stride of pose initializer and frame encoder is 8. In DKD, the backbone of pose initializer is fixed as ResNet-101, and the backbone of frame encoder is chosen among ResNet-x; x 2 f18; 34; 50g. For fair comparison, we follow the settings of DKD, fixing the backbone of our HeadNet as ResNet101 and varying the backbone of our BodyNet among ResNet-x; x 2 f18; 34; 50g. Results are shown in Table. 4, where Framewise (ResNet-x) is used to denote single-image framewise pose estimation results with ResNet-x as backbone. DKD (ResNet-x) and PPN (ResNet-x) repre-sent DKD model with ResNet-x as the backbone of frame encoder and our PPN with ResNet-x as the backbone of BodyNet, respectively. Note that Framewise (ResNet-x) and PPN (ResNet-x) here denote different configurations from those in Table. 1. The FLOPs and evaluation result on PCK-torso of each configuration are reported.

It can be observed from Table. 4 that our PPN signifi-cantly outperforms DKD in the stricter metric PCK-torso, with 1.13% accuracy improvement in average. Especially for

localization of shoulder and wrist joints, PPN achieves 1.63% and 1.90% accuracy improvement in average, respectively. Moreover, compared with the pose kernels employed in DKD that transfer temporal knowledge, our designed PPU propa-gates well-estimated poses across frames to provide temporal guidance in a more compact manner (0.20G versus 0.99G additional FLOPs against baselines). The superiority of our method is thus verified from the perspective of both accuracy and efficiency.

In addition, we compare our two representative config-urations, PPN-Stable and PPN-Swift, with previous state-of-the-art methods in the field of video pose estimation on Penn Action Dataset, as shown in Table. 5. We can observe that our method significantly outperforms all of the previous state-of-the-art methods in both accuracy and efficiency. As for accuracy, PPN-Stable achieves 1.0% improvement on PCK-body and 1.3% improvement on PCK-torso over the previous best method. Our tiny configuration PPN-Swift also produces better results compared with the state-of-the-arts, achieving 0.7% improvement on PCK-body and 0.9% improvement on PCK-torso over the previous best method. Moreover, our method diminishes computational complexity by a large margin compared with the state-of-the-arts. Com-pared with LSTM Pose Machines proposed by Luo et al. [4], PPN reduces FLOPs by a magnitude over (3.87G/1.39G versus 70.98G). Compared with the previous best method [5], our two configurations, PPN-Stable and PPN-Swift, achieve 2.5 and 6 FLOPs reduction respectively. We visualize the comparison of accuracy and efficiency between our method and the above two state-of-the-art methods in Fig. 1(d), demonstrating the great superiority of our method.

Table. 6 shows the comparison results on Sub-JHMDB Dataset between our method and the previous state-of-the-arts. The scale of person instance in Sub-JHMDB Dataset is generally smaller than that in Penn Action Dataset, which makes it more challenging to generate accurate pose esti-mation results on Sub-JHMDB Dataset. Compared with the previous best method [5], our two configurations, PPN-Stable and PPN-Swift, achieve 2.4% and 1.9% accuracy improve-ment on metric PCK-body, and 3.9% and 3.0% accuracy improvement on metric PCK-torso.

**E. QUALITATIVE RESULTS**

We provide some qualitative results generated on randomly selected frames from Penn Action Dataset and Sub-JHMDB Dataset to demonstrate the capability of our PPN. As shown in Fig. 5, PPN can robustly produce accurate pose estimation results against several troublesome factors, such as motion blur (the 3rd row of Fig. 5(b)), scale change (the 4th row of Fig. 5(b)) and articulated occlusion (the 3rd and 4th rows of Fig. 5(a), the 1st and 2nd rows of Fig. 5(b)). Besides, frames with crowded background can be effectively dealt with, as shown in the 1st row of Fig. 5(a). Moreover, the person scale, viewpoint and illumination vary among frames, reflecting the great robustness of our proposed PPN.

8 VOLUME 4, 2016

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2998121, IEEE Access



Y. Liu et al.: PosePropagationNet: Towards Accurate and Efficient Pose Estimation in Videos

**V. CONCLUSIONS**

In this paper, we propose a novel architecture, PosePropa-gationNet, for video pose estimation. We implement pose propagation mechanism via the design of pose propagation unit in PPN, allowing well-estimated poses to be propagated across frames as the most explicit temporal guidance. Ben-efitting from the pose propagation mechanism, lightweight networks gain the capability of performing accurate pose estimation in videos. Our experiments on two large-scale benchmarks, Penn Action Dataset and Sub-JHMDB Dataset, show that our method significantly outperforms previous state-of-the-art methods both in accuracy and in efficiency. Our two representative configurations, PPN-Stable and PPN-Swift, achieve 2.5 and 6 FLOPs reduction respectively over the previous best method, as well as significant accuracy improvement.

**ACKNOWLEDGMENT**

This work was supported by the National Natural Science Foundation of China under Grant 61673234.

**REFERENCES**

1. J. Song and L. Wang and L. V. Gool and O. Hilliges, “Thin-Slicing Network: A Deep Structured Model for Pose Estimation in Videos", in Proc. CVPR, 2017, pp. 5563-5572.
2. T. Pfister and J. Charles and A. Zisserman, “Flowing Convnets for Human Pose Estimation in Videos", in Proc. CVPR, 2015, pp. 1913-1921.
3. D. Zhang and G. Guo and D. Huang and J. Han, “Poseflow: A Deep Motion Representation for Understanding Human Behaviors in Videos", in Proc. CVPR, 2018, pp. 6762-6770.
4. Y. Luo and J. Ren and Z. Wang and W. Sun and J. Pan and J. Liu and J. Pang and L. Lin, “LSTM Pose Machines", in Proc. CVPR, 2018, pp. 5207-5215.
5. X. Nie and Y. Li and L. Luo and N. Zhang and J. Feng, “Dynamic Kernel Distillation for Efficient Pose Estimation in Videos", in Proc. ICCV, 2019, pp. 6941-6949.
6. W. Zhang and M. Zhu and K. G. Derpanis, “From Actemes to Action: A Strongly-Supervised Representation for Detailed Action Understanding", in Proc. ICCV, 2013, pp. 2248-2255.
7. H. Jhuang and J. Gall and S. Zuffi and C. Schmid and M. J. Black, “Towards Understanding Action Recognition", in Proc. ICCV, 2013, pp. 3192-3199.
8. M. Andriluka and S. Roth and B. Schiele, “Pictorial Structures Revisited: People Detection and Articulated Pose Estimation", in Proc. CVPR, 2009, pp. 1014-1021.
9. S. Johnson and M. Everingham, “Clustered Pose and Nonlinear Appear-ance Models for Human Pose Estimation", in Proc. BMVC, 2010, pp. 12.1–12.11.
10. L. Pishchulin and M. Andriluka and P. V. Gehler and B. Schiele, “Strong Appearance and Expressive Spatial Models for Human Pose Estimation", in Proc. ICCV, 2013, pp. 3487-3494.
11. Y. Yang and D. Ramanan, “Articulated Pose Estimation with Flexible Mixtures-of-Parts", in Proc. CVPR, 2011, pp. 1385-1392.
12. A. Toshev and C. Szegedy, “Deeppose: Human Pose Estimation via Deep Neural Networks", in Proc. CVPR, 2014, pp. 1653-1660.
13. A. Newell and K. Yang and J. Deng, “Stacked Hourglass Networks for Human Pose Estimation", in Proc. ECCV, 2016, pp. 483-499.
14. K. Sun and B. Xiao and D. Liu and J. Wang, “Deep High-Resolution Representation Learning for Human Pose Estimation", in Proc. CVPR, 2019, pp. 5686-5696.
15. S-E. Wei and V. Ramakrishna and T. Kanade and Y. Sheikh, “Convolu-tional Pose Machines", in Proc. CVPR, 2016, pp. 4724-4732.
16. B. Xiao and H. Wu and Y. Wei, “Simple Baselines for Human Pose Estimation and Tracking", in Proc. ECCV, 2018, pp. 466-481.
17. W. Yang and S. Li and W. Ouyang and H. Li and X. Wang, “Learning Feature Pyramids for Human Pose Estimation", in Proc. ICCV, 2017, pp. 1290-1299.
18. M. Andriluka and L. Pishchulin and P. V. Gehler and B. Schiele, “2D Human Pose Estimation: New Benchmark and State of the Art Analysis", in Proc. CVPR, 2014, pp. 3686-3693.
19. B. Sapp and B. Modec, “Multimodal Decomposable Models for Human Pose Estimation", in Proc. CVPR, 2013, pp. 3674-3681.
20. P. Weinzaepfel and J. Revaud and Z. Harchaoui and C. Schmid, “Deep-flow: Large Displacement Optical Flow with Deep Matching", in Proc. ICCV, 2013, pp. 1385-1392.
21. G. Gkioxari and A. Toshev and N. Jaitly, “Chained Predictions Using Convolutional Neural Networks", in Proc. ECCV, 2016, pp. 728-743.
22. R. Girdhar and G. Gkioxari and L. Torresani and M. Paluri and D. Tran, “Detect-and-Track: Efficient Pose Estimation in Videos", in Proc. CVPR, 2018, pp. 350-359.
23. G. Bertasius and C. Feichtenhofer and D. Tran and J. Shi and L. Torresani, “Learning Temporal Pose Estimation from Sparsely-Labeled Videos", in Proc. NeurIPS, 2019, pp. 3021-3032.
24. J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, “Deformable Convolutional Networks", in Proc. ICCV, 2017, pp. 764-773.
25. D. P. Kingma and J. Ba. “Adam: A Method for Stochastic Optimization,"

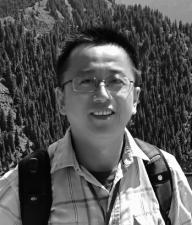
2014, [Online]. Available: https://arxiv.org/abs/1412.6980.

1. W. Yang and W. Ouyang and H. Li and X. Wang, “End-to-End Learning of Deformable Mixture of Parts and Deep Convolutional Neural Networks for Human Pose Estimation", in Proc. CVPR, 2016, pp. 3073-3082.
2. K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition," in Proc. CVPR, 2016, pp. 770-778.
3. P. Wang and P. Chen and Y. Yuan and D. Liu and Z. Huang and X. Hou and G. Cottrell, “Understanding Convolution for Semantic Segmentation", in Proc. WACV, 2018, pp. 1451-1460.
4. M. Sandler and A. Howard and M. Zhu and A. Zhmoginov and L. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks", in Proc. CVPR, 2018, pp. 4510-4520.
5. D. Park and D. Ramanan, “N-best Maximal Decoders for Part Models", in Proc. ICCV, 2011, pp. 2627-2634.
6. B. X. Nie and C. Xiong and S. Zhu, “Joint Action Recognition and Pose Estimation from Video", in Proc. CVPR, 2015, pp. 1293-1301.
7. U. Iqbal and M. Garbade and J. Gall, “Pose for Action - Action for Pose", in Proc. FG, 2017, pp. 438-445.
8. K. Simonyan and A. Zisserman, “Very Deep Convolutional Net-works for Large-scale Image Recognition", 2014, [Online]. Available: https://arxiv.org/abs/1409.1556.

YU LIU received the bachelor’s degree from Nan-jing University of Aeronautics and Astronautics, Nanjing, China, in 2017. He is currently pursuing the master degree with the Tsinghua University. His research interests involve machine learning and action recognition.



JIANSHENG CHEN is an associate professor in the Department of Electronic Engineering, Ts-inghua University, Beijing, China. He received the B.E. and M.E. degrees, both in computer science and technology, from Tsinghua University, Bei-jing, China, in 2000 and 2002, respectively. He received the Ph.D. degree in computer science and engineering from the Chinese University of Hong Kong in 2007. His research interests include im-age processing, pattern recognition and machine



learning.



|  |  |
| --- | --- |
| VOLUME 4, 2016 | 9 |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.