Simple Baseline for Human Pose Estimation and Tracking (2018)

Seho Kim

https://arxiv.org/pdf/1804.06208.pdf

Introduction

- ‘How good could a simple method be?’

  ; provide baseline methods for both pose estimation and tracking(simple and effective)

- Based on a few deconvolutional layers added on a backbone network, ResNet
- SOTA(mAP 73.7 on COCO test-dev)
- Pose tracking
  - Follow a similar pipeline of the winner of ICCV’17 PoseTrack Challenge + use optical flow based pose propagation and similarity measurement
  - SOTA mAP score of 74.6 and a MOTA score of 57.8

Pose Estimation Using A Deconvolution Head Network

- Simply add a few deconvolutional layers over the last convolution stage in the ResNet
- Three deconvolutional layers with batch normalization and ReLU activation are used
- Each layer has 256 filters with 4x4 kernel, stride 2
- 1x1 convolutional layer is added at last to generate predicted heatmaps for all k key points
- Mean Squared Error(MSE) loss between the predicted heatmaps and target heatmaps(2D gaussian centered)

Pose Estimation Using A Deconvolution Head Network

- Hourglass
  - The basis for all leading methods
  - A multi-stage architecture with repeated bottom-up, top-down processing and skip layer feature concatenation
- Cascaded Pyramid Network(CPN)
  - Leading method on COCO 2017 keypoint challenge
  - Involve skip layer feature concatenation and an online hard keypoint mining step
- Simplebaseline combines the upsampling and convolutional parameters into deconvolutional layers → Much simpler way
- Commonality of the three methods ; three upsampling and three non-linearity
- Preliminary and heuristic

Pose Estimation Using A Deconvolution Head Network

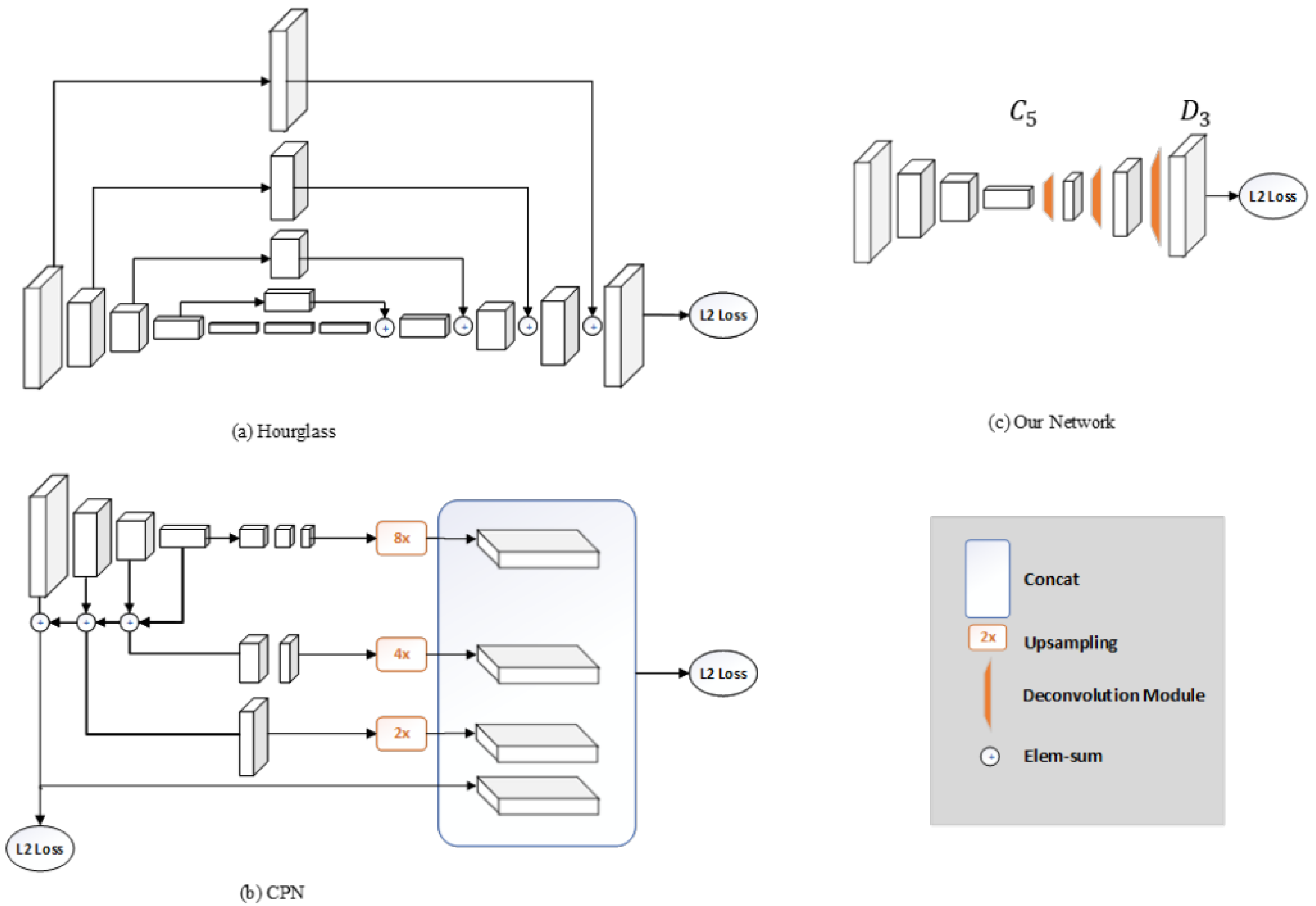

Fig. 1. Illustration of two state-of-the-art network architectures for pose estimation (a) one stage in Hourglass [22], (b) CPN [6], and our simple baseline (c).

Pose Tracking Based on Optical Flow

- Multi-person pose tracking in videos
  - Estimate human poses → Tracks the human pose by assigning id
  - Follow the winner of ICCV’17 PoseTrack Challenge(Mask R-CNN + Greedy bipartite matching algorithm) with two differences
    - Two different kinds of human boxes

      ; from human detector and generated from previous frames using optical flow
    - Similarity metric used by the greedy matching algorithm

      ; flow-based pose similarity metric

Pose Tracking Based on Optical Flow

- Joint Propagation using Optical Flow
  - Missing detections and false detections due to motion blur and occlusion
  - Given one human instance with joints + optical flow field

    → Estimate the corresponding joints

      ; compute a bounding of the propagated joints and expand box by some extend(15% in experiments) as the candidated box

Pose Tracking Based on Optical Flow

- Flow-based Pose Similarity
  - Using bbox IoU as the similarity metric to link instances

    ; Problematic because of fast movement and crowd scene
  - Pose similarity using OKS(more fine-grained metric)

    ; Problematic because of pose changing
  - Propose to use a flow-based pose similarity metric

$$S_{Flow}(J_i^k, J_j^l) = OKS(\hat{J}_i^l, J_j^l),$$

Pose Tracking Based on Optical Flow

- Flow-based Pose Tracking Algorithm
  - Solve the pose estimation problem
    - Complement; the boxes
  - Solve the tracking problem

Table 1. Notations in Algorithm 1.

I^k	k^{th} frame
Q	tracked instances queue
L_Q	max capacity of Q
$\mathcal{P}^k$	instances set in k^{th} frame
$\mathcal{J}^k$	instances set of body joints in k^{th} frame
P_i^k	i^{th} instance in k^{th} frame
J_i^k	body joints set of i^{th} instance in k^{th} frame
$F_{k\rightarrow l}$	flow field from k^{th} frame to l^{th} frame
M_{sim}	similariy matrix
B^k_{det}	boxes from person detector in k^{th} frame
B^k_{flow}	boxes generated by joint propagating in k^{th} frame
B^k_{unified}	boxes unified by box NMS in k^{th} frame
$\mathcal{N}_{\text{det}}$	person detection network
$\mathcal{N}_{\text{pose}}$	human pose estimation network
$\mathcal{N}_{\text{flow}}$	flow estimation network
$\mathcal{F}_{\text{sim}}$	function for calculating similarity matrix
$\mathcal{F}_{\text{NMS}}$	function for NMS operation
$\mathcal{F}_{\text{FlowBoxGen}}$	function for generating boxes by joint propagating
$\mathcal{F}_{\text{AssignID}}$	function for assigning instance id

Algorithm 1 The flow-based inference algorithm for video human pose tracking

input: video frames $\{I^k\}$, $Q = []$, Q's max capacity L_Q.
$B^0_{\text{det}} = \mathcal{N}_{\text{det}}(I^0)$
$\mathcal{J}^0 = \mathcal{N}_{\text{pose}}(I^0, B^0_{\text{det}})$
$\mathcal{P}^0 = (\mathcal{J}^0, id)$ ▷ initialize the id for the first frame
$Q = [\mathcal{P}_0]$ ▷ append the instance set $\mathcal{P}_0$ to Q
for $k = 1$ **to** ∞ **do**
  $B^k_{\text{det}} = \mathcal{N}_{\text{det}}(I^k)$
  $B^k_{\text{flow}} = \mathcal{F}_{\text{FlowBoxGen}}(\mathcal{J}^{k-1}, F_{k-1\rightarrow k})$
  $B^k_{\text{unified}} = \mathcal{F}_{\text{NMS}}(B^k_{\text{det}}, B^k_{\text{flow}})$ ▷ unify detection boxes and flow boxes
  $\mathcal{J}^k = \mathcal{N}_{\text{pose}}(I^k, B^k_{\text{unified}})$
  $M_{\text{sim}} = \mathcal{F}_{\text{sim}}(Q, \mathcal{J}^k)$
  $\mathcal{P}^k = \mathcal{F}_{\text{AssignID}}(M_{\text{sim}}, \mathcal{J}^k)$
  append $\mathcal{P}^k$ to Q ▷ update the Q
end for

Pose Tracking Based on Optical Flow

- Flow-based Pose Tracking Algorithm
  - Solve the pose estimation problem
    - Complement; the boxes(human detector and propagating joints) are unified by using NMS
  - Solve the tracking problem
    - Store the tracked instances in a Deque with fixed length

$$Q = [\mathcal{P}_{k-1}, \mathcal{P}_{k-2}, ..., \mathcal{P}_{k-L_Q}]$$

Table 1. Notations in Algorithm 1

I^k	k^{th} frame
Q	tracked instances queue
L_Q	max capacity of Q
$\mathcal{P}^k$	instances set in k^{th} frame
$\mathcal{J}^k$	instances set of body joints in k^{th} frame
P_i^k	i^{th} instance in k^{th} frame
J_i^k	body joints set of i^{th} instance in k^{th} frame
$F_{k \to l}$	flow field from k^{th} frame to l^{th} frame
M_{sim}	similariy matrix
B_{det}^k	boxes from person detector in k^{th} frame
B_{flow}^k	boxes generated by joint propagating in k^{th} frame
B_{unified}^k	boxes unified by box NMS in k^{th} frame
$\mathcal{N}_{\text{det}}$	person detection network
$\mathcal{N}_{\text{pose}}$	human pose estimation network
$\mathcal{N}_{\text{flow}}$	flow estimation network
$\mathcal{F}_{\text{sim}}$	function for calculating similarity matrix
$\mathcal{F}_{\text{NMS}}$	function for NMS operation
$\mathcal{F}_{\text{FlowBoxGen}}$	function for generating boxes by joint propagating
$\mathcal{F}_{\text{AssignID}}$	function for assigning instance id

Algorithm 1 The flow-based inference algorithm for video human pose tracking

```
input: video frames {I^k}, Q = [], Q's max capacity L_Q.
B^0_det = N_det(I^0)
J^0 = N_pose(I^0, B^0_det)
P^0 = (J^0, id)                                   ▷ initialize the id for the first frame
Q = [P_0]                                         ▷ append the instance set P_0 to Q
for k = 1 to ∞ do
    B^k_det = N_det(I^k)
    B^k_flow = F_FlowBoxGen(J^{k-1}, F_{k-1→k})
    B^k_unified = F_NMS(B^k_det, B^k_flow)        ▷ unify detection boxes and flow boxes
    J^k = N_pose(I^k, B^k_unified)
    M_sim = F_sim(Q, J^k)
    P^k = F_AssignID(M_sim, J^k)
    append P^k to Q                               ▷ update the Q
end for
```

Experiments

- Pose Estimation on COCO
  - Training
  - Testing
  - Ablation Study
  - Comparison with Other Methods on COCO val2017
  - Comparison on COCO test-dev dataset

Table 2. Ablation study of our method on COCO val2017 dataset. Those settings used in comparison are in **bold**. For example, (a, e, f) compares backbones.

Method	Backbone	Input Size	#Deconv. Layers	Deconv. Kernel Size	AP
a	**ResNet-50**	$\mathbf{256 \times 192}$	**3**	**4**	70.4
b	ResNet-50	256×192	**2**	4	67.9
c	ResNet-50	256×192	3	**2**	70.1
d	ResNet-50	256×192	3	**3**	70.3
e	**ResNet-101**	256×192	3	4	71.4
f	**ResNet-152**	256×192	3	4	72.0
g	ResNet-50	$\mathbf{128 \times 96}$	3	4	60.6
h	ResNet-50	$\mathbf{384 \times 288}$	3	4	72.2

Experiments

- Pose Estimation on COCO

Table 3. Comparison with Hourglass [22] and CPN [6] on COCO val2017 dataset. Their results are cited from [6]. OHKM means Online Hard Keypoints Mining.

Method	Backbone	Input Size	OHKM	AP
8-stage Hourglass	-	256 × 192	✗	66.9
8-stage Hourglass	-	256 × 256	✗	67.1
CPN	ResNet-50	256 × 192	✗	68.6
CPN	ResNet-50	384 × 288	✗	70.6
CPN	ResNet-50	256 × 192	✓	69.4
CPN	ResNet-50	384 × 288	✓	71.6
Ours	ResNet-50	256 × 192	✗	70.4
Ours	ResNet-50	384 × 288	✗	72.2

Table 4. Comparisons on COCO test-dev dataset. **Top**: methods in the literature, trained only on COCO training dataset. **Middle**: results submitted to COCO test-dev leaderboard [9], which have either extra training data (*) or models ensamled ($^+$). **Bottom**: our single model results, trained only on COCO training dataset.

Method	Backbone	Input Size	AP	AP_{50}	AP_{75}	AP_m	AP_l	AR
CMU-Pose [5]	-	-	61.8	84.9	67.5	57.1	68.2	66.5
Mask-RCNN [12]	ResNet-50-FPN	-	63.1	87.3	68.7	57.8	71.4	-
G-RMI [24]	ResNet-101	353 × 257	64.9	85.5	71.3	62.3	70.0	69.7
CPN [6]	ResNet-Inception	384 × 288	72.1	91.4	80.0	68.7	77.2	78.5
FAIR* [9]	ResNeXt-101-FPN	-	69.2	90.4	77.0	64.9	76.3	75.2
G-RMI* [9]	ResNet-152	353 × 257	71.0	87.9	77.7	69.0	75.2	75.8
oks* [9]	-	-	72.0	90.3	79.7	67.6	78.4	77.1
bangbangren*$^+$ [9]	ResNet-101	-	72.8	89.4	79.6	68.6	**80.0**	78.7
CPN$^+$ [6,9]	ResNet-Inception	384 × 288	73.0	**91.7**	80.9	69.5	78.1	**79.0**
Ours	ResNet-152	384 × 288	**73.7**	**91.9**	**81.1**	**70.3**	**80.0**	**79.0**

Experiments

- Pose Estimation and Tracking on PoseTrack
  - Training
  - Testing
  - Effect of Joint Propagation
  - Effect of Flow-based Pose Similarity
  - Comparison with State-of-the-Art

Table 5. Ablation study on PoseTrack Challenge validation dataset. **Top**: Results of ResNet-50 backbone using R-FCN detector. **Middle**: Results of ResNet-50 backbone using FPN-DCN detector. **Bottom**: Results of ResNet-152 backbone using FPN-DCN detector.

Method	Backbone	Detector	With Joint Propagation	Similarity Metric	mAP Total	MOTA Total
a_1	ResNet-50	R-FCN	✗	S_{Bbox}	66.0	57.6
a_2	ResNet-50	R-FCN	✗	S_{Pose}	66.0	57.7
a_3	ResNet-50	R-FCN	✓	S_{Bbox}	70.3	61.4
a_4	ResNet-50	R-FCN	✓	S_{Pose}	70.3	61.8
a_5	ResNet-50	R-FCN	✓	S_{Flow}	70.3	61.8
a_6	ResNet-50	R-FCN	✓	$S_{Multi-Flow}$	**70.3**	**62.2**
b_1	ResNet-50	FPN-DCN	✗	S_{Bbox}	69.3	59.8
b_2	ResNet-50	FPN-DCN	✗	S_{Pose}	69.3	59.7
b_3	ResNet-50	FPN-DCN	✓	S_{Bbox}	72.4	62.1
b_4	ResNet-50	FPN-DCN	✓	S_{Pose}	72.4	61.8
b_5	ResNet-50	FPN-DCN	✓	S_{Flow}	72.4	62.4
b_6	ResNet-50	FPN-DCN	✓	$S_{Multi-Flow}$	**72.4**	**62.9**
c_1	ResNet-152	FPN-DCN	✗	S_{Bbox}	72.9	62.0
c_2	ResNet-152	FPN-DCN	✗	S_{Pose}	72.9	61.9
c_3	ResNet-152	FPN-DCN	✓	S_{Bbox}	76.7	64.8
c_4	ResNet-152	FPN-DCN	✓	S_{Pose}	76.7	64.9
c_5	ResNet-152	FPN-DCN	✓	S_{Flow}	76.7	65.1
c_6	ResNet-152	FPN-DCN	✓	$S_{Multi-Flow}$	**76.7**	**65.4**

Table 6. Multi-person Pose Estimation Performance on PoseTrack Challenge dataset. "*" means models trained on train+validation set. **Top**: Results on PoseTrack validation set. **Bottom**: Results on PoseTrack test set

Method	Dataset	Head mAP	Sho. mAP	Elb. mAP	Wri. mAP	Hip mAP	Knee mAP	Ank. mAP	Total mAP
Girdhar et al. [11]	val	67.5	70.2	62.0	51.7	60.7	58.7	49.8	60.6
Xiu et al. [32]	val	66.7	73.3	68.3	61.1	67.5	67.0	61.3	66.5
Ours:ResNet-50	val	79.1	80.5	75.5	66.0	70.8	70.0	61.7	72.4
Ours:ResNet-152	val	81.7	83.4	80.0	72.4	75.3	74.8	67.1	**76.7**
Girdhar et al.* [11]	test	-	-	-	-	-	-	-	59.6
Xiu et al. [32]	test	64.9	67.5	65.0	59.0	62.5	62.8	57.9	63.0
Ours:ResNet-50	test	76.4	77.2	72.2	65.1	68.5	66.9	60.3	70.0
Ours:ResNet-152	test	79.5	79.7	76.4	70.7	71.6	71.3	64.9	**73.9**

Experiments

Table 7. Multi-person Pose Tracking Performance on PoseTrack Challenge dataset. "*" means models trained on train+validation set. **Top**: Results on PoseTrack validation set. **Bottom**: Results on PoseTrack test set

Method	Dataset	MOTA Head	MOTA Sho.	MOTA Elb.	MOTA Wri.	MOTA Hip	MOTA Knee	MOTA Ank.	MOTA Total	MOTP Total	Prec Total	Rec Total
Girdhar et al. [11]	val	61.7	65.5	57.3	45.7	54.3	53.1	45.7	55.2	61.5	66.4	88.1
Xiu et al. [32]	val	59.8	67.0	59.8	51.6	60.0	58.4	50.5	58.3	67.8	70.3	87.0
Ours:ResNet-50	val	72.1	74.0	61.2	53.4	62.4	61.6	50.7	62.9	84.5	86.3	76.0
Ours:ResNet-152	val	73.9	75.9	63.7	56.1	65.5	65.1	53.5	**65.4**	85.4	85.5	80.3
Girdhar et al.* [11]	test	-	-	-	-	-	-	-	51.8	-	-	-
Xiu et al. [32]	test	52.0	57.4	52.8	46.6	51.0	51.2	45.3	51.0	16.9	71.2	78.9
Ours:ResNet-50	test	65.9	67.0	51.5	48.0	56.2	54.6	46.9	56.4	45.5	81.0	75.7
Ours:ResNet-152	test	67.1	68.4	52.2	48.9	56.1	56.6	48.8	**57.6**	62.6	79.4	79.9

Table 8. Results of Mulit-Person Pose Tracking on PoseTrack Challenge Leaderboard. "*" means models trained on train+validation set.

Entry	Additional Training Dataset	mAP	MOTA
ProTracker [11]	COCO	59.6	51.8
PoseFlow [26]	COCO+MPII-Pose	63.0	51.0
MVIG [26]	COCO+MPII-Pose	63.2	50.7
BUTD2 [17]	COCO	59.2	50.6
SOPT-PT [26]	COCO+MPII-Pose	58.2	42.0
ML-LAB [34]	COCO+MPII-Pose	70.3	41.8
Ours:ResNet152*	COCO	**74.6**	**57.8**

Conclusions