OpenPose
: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields (2018)

Seho Kim

https://arxiv.org/pdf/1812.08008.pdf

Introduction

- Human 2D pose estimation; localizing keypoints or parts
- Inferring the pose of multiple people in images presents a unique set of challenges
  - Unknown number of people
  - Complex spatial interference
  - Runtime complexity; realtime performance
- Top-down approaches(common approach)
  - Failure of person detector
  - Runtime is proportional to the number of people in image

Introduction

- Bottom-up approaches
  - vs Top-down
    - Robustness to early commitment
    - Potential to decouple runtime complexity
  - Do not directly use global contextual cues
  - Initial bottom-up methods

    ; Did not retain the gains in efficiency

  → An efficient method
    - Part Affinity Fields(PAFs)
    - Contributions
      - PAF refinement; crucial for maximizing accuracy
      - An annotated foot dataset
      - Vehicle keypoint estimation
      - Document the release of Openpose

Method

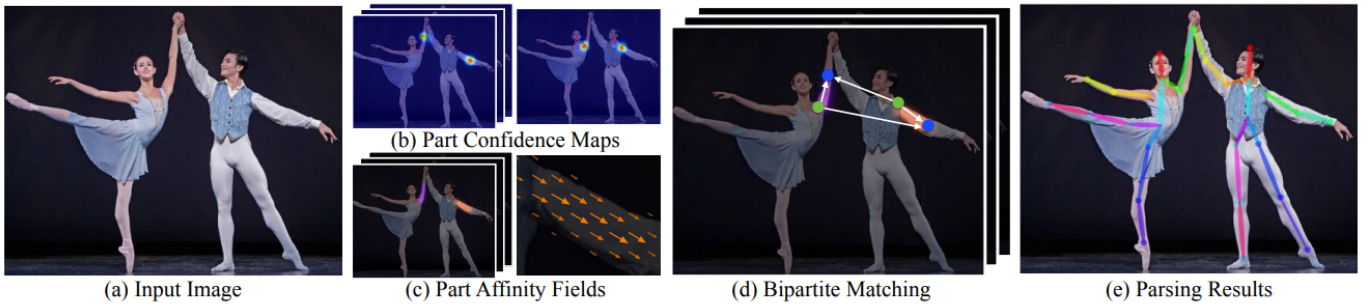


Fig. 2: Overall pipeline. (a) Our method takes the entire image as the input for a CNN to jointly predict (b) confidence maps for body part detection and (c) PAFs for part association. (d) The parsing step performs a set of bipartite matchings to associate body part candidates. (e) We finally assemble them into full body poses for all people in the image.

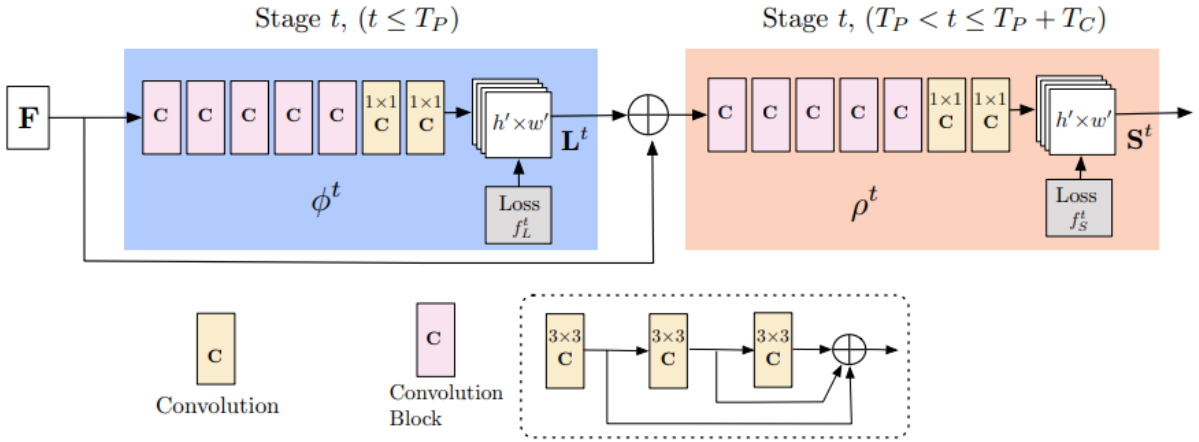


Fig. 3: Architecture of the multi-stage CNN. The first set of stages predicts PAFs $\mathbf{L}^t$, while the last set predicts confidence maps $\mathbf{S}^t$. The predictions of each stage and their corresponding image features are concatenated for each subsequent stage. Convolutions of kernel size 7 from the original approach [3] are replaced with 3 layers of convolutions of kernel 3 which are concatenated at their end.

Method

- Network Architecture
  - Predict affinity fields and detection confidence maps
  - Iterative prediction architecture refines the predictions
  - Original approach; several 7x7 convolutional layers

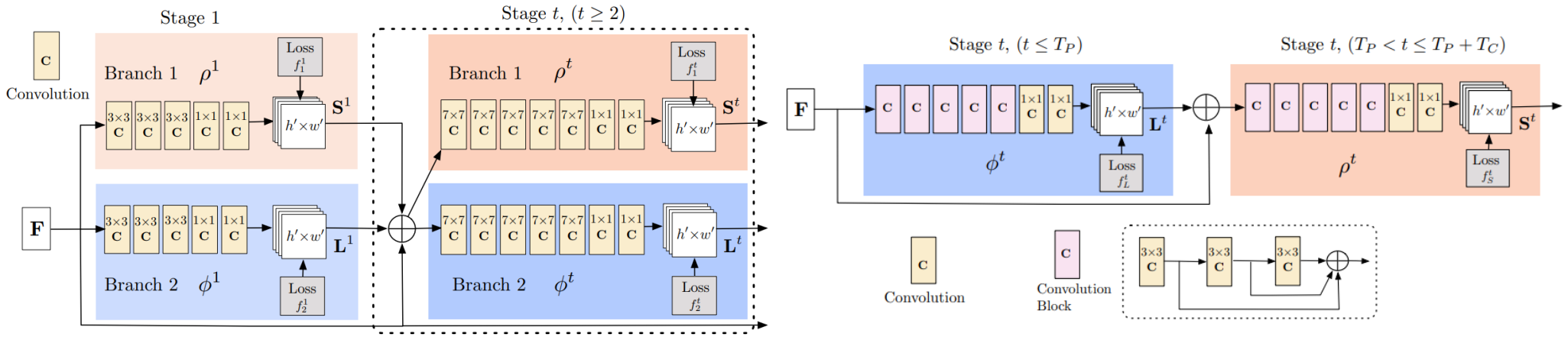
    → Current model; 3 consecutive 3x3 kernels; the receptive field is preserved and the computation is reduced(91→51)
  - Similar to DenseNet

    ; keep both lower level and higher level features

Method

- Simultaneous Detection and Association
  - Initialized by the first 10 layers of VGG-19 and finetuned
  - Computation per stage is reduced by half

$$
\begin{aligned}
\mathbf{L}^t &= \phi^t(\mathbf{F}, \mathbf{L}^{t-1}), \ \forall 2 \leq t \leq T_P, \\
\mathbf{S}^{T_P} &= \rho^t(\mathbf{F}, \mathbf{L}^{T_P}), \ \forall t = T_P, \\
\mathbf{S}^t &= \rho^t(\mathbf{F}, \mathbf{L}^{T_P}, \mathbf{S}^{t-1}), \ \forall T_P < t \leq T_P + T_C,
\end{aligned}
$$



Method

- Simultaneous Detection and Association
  - Use L2 loss between the estimated predictions and the groundtruth maps and fields
  - Weight the loss functions to address not completely labeled datasets issue

$$f_{\mathbf{L}}^{t_i} = \sum_{c=1}^{C} \sum_{\mathbf{p}} \mathbf{W}(\mathbf{p}) \cdot \|\mathbf{L}_c^{t_i}(\mathbf{p}) - \mathbf{L}_c^*(\mathbf{p})\|_2^2,$$

$$f_{\mathbf{S}}^{t_k} = \sum_{j=1}^{J} \sum_{\mathbf{p}} \mathbf{W}(\mathbf{p}) \cdot \|\mathbf{S}_j^{t_k}(\mathbf{p}) - \mathbf{S}_j^*(\mathbf{p})\|_2^2,$$

$$f = \sum_{t=1}^{T_P} f_{\mathbf{L}}^t + \sum_{t=T_P+1}^{T_P+T_C} f_{\mathbf{S}}^t.$$

Method

- Confidence Maps for Part Detection

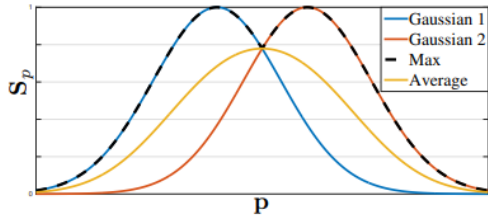
We first generate individual confidence maps $\mathbf{S}^*_{j,k}$ for each person k. Let $\mathbf{x}_{j,k} \in \mathbb{R}^2$ be the groundtruth position of body part j for person k in the image. The value at location $\mathbf{p} \in \mathbb{R}^2$ in $\mathbf{S}^*_{j,k}$ is defined as,

$$\mathbf{S}^*_{j,k}(\mathbf{p}) = \exp\left(-\frac{||\mathbf{p} - \mathbf{x}_{j,k}||_2^2}{\sigma^2}\right), \tag{6}$$

where σ controls the spread of the peak. The groundtruth confidence map to be predicted by the network is an aggregation of the individual confidence maps via a max operator,

$$\mathbf{S}^*_j(\mathbf{p}) = \max_k \mathbf{S}^*_{j,k}(\mathbf{p}). \tag{7}$$

We take the maximum of the confidence maps instead of the average so that the precision of close by peaks remains distinct, as illustrated in the right figure. At test time, we predict confidence maps (as shown in the first row of Fig. 4), and obtain body part candidates by performing non-maximum suppression.



Method

- Part Affinity Fields for Part Association
  - Possible way:
    - Additional midpoint; is likely to support false associations
      - Limitations:
        - only position
        - reduce the region of support of a limb to a single point
    - Part Affinity Fields(PAFs); 2D vector field
      - Preserve both location and orientation information

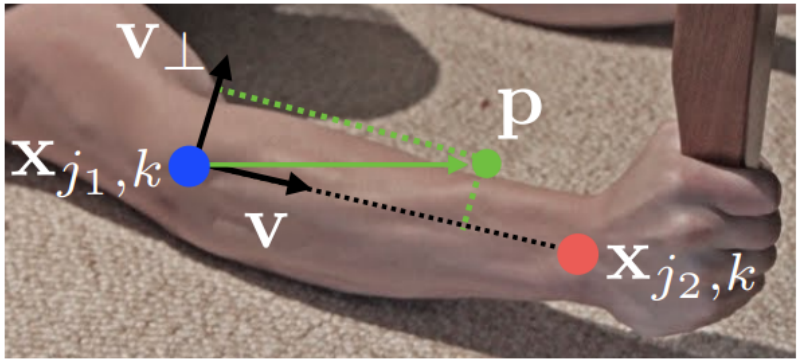
Method

- Part Affinity Fields for Part Association
  - Make groundtruth

$$\mathbf{L}^*_{c,k}(\mathbf{p}) = \begin{cases} \mathbf{v} & \text{if } \mathbf{p} \text{ on limb } c,k \\ \mathbf{0} & \text{otherwise.} \end{cases}$$

$$0 \leq \mathbf{v} \cdot (\mathbf{p} - \mathbf{x}_{j_1,k}) \leq l_{c,k} \text{ and } |\mathbf{v}_\perp \cdot (\mathbf{p} - \mathbf{x}_{j_1,k})| \leq \sigma_l,$$

$$\mathbf{L}^*_c(\mathbf{p}) = \frac{1}{n_c(\mathbf{p})} \sum_k \mathbf{L}^*_{c,k}(\mathbf{p}),$$



  - Measure association(confidence)

$$E = \int_{u=0}^{u=1} \mathbf{L}_c(\mathbf{p}(u)) \cdot \frac{\mathbf{d}_{j_2} - \mathbf{d}_{j_1}}{||\mathbf{d}_{j_2} - \mathbf{d}_{j_1}||_2} du,$$

$$\mathbf{p}(u) = (1-u)\mathbf{d}_{j_1} + u\mathbf{d}_{j_2}.$$

Method

- Multi-Person Parsing using PAFs
  - NMS on the detection confidence maps
  - K-dimensional matching problem(NP-Hard)

    ; Two relaxations to the optimization
    - Choose minimal number of edges
    - Decompose the matching problem and determine the matching

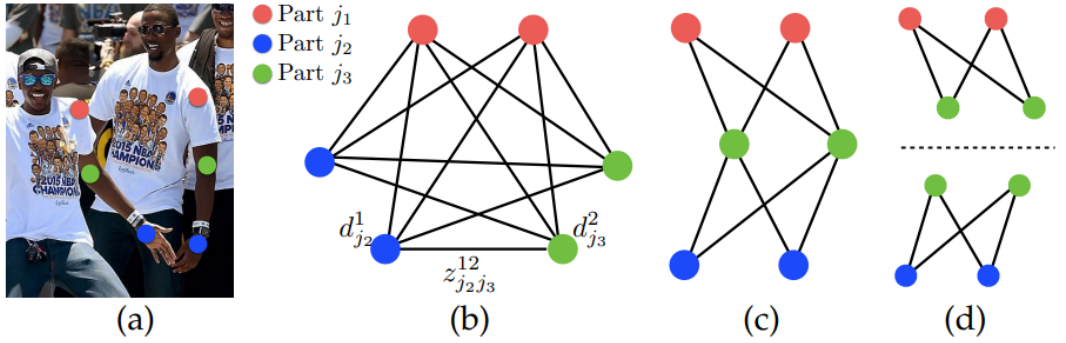


Fig. 6: Graph matching. (a) Original image with part detections. (b) K-partite graph. (c) Tree structure. (d) A set of bipartite graphs.

$$\max_{\mathcal{Z}_c} E_c = \max_{\mathcal{Z}_c} \sum_{m \in \mathcal{D}_{j_1}} \sum_{n \in \mathcal{D}_{j_2}} E_{mn} \cdot z^{mn}_{j_1 j_2},$$

$$\text{s.t.} \quad \forall m \in \mathcal{D}_{j_1}, \sum_{n \in \mathcal{D}_{j_2}} z^{mn}_{j_1 j_2} \leq 1,$$

$$\forall n \in \mathcal{D}_{j_2}, \sum_{m \in \mathcal{D}_{j_1}} z^{mn}_{j_1 j_2} \leq 1,$$

$$\max_{\mathcal{Z}} E = \sum_{c=1}^{C} \max_{\mathcal{Z}_c} E_c.$$

Openpose

- The first real-time multi-person system to jointly detect human keypoints on single images
- System
  - 2D body pose estimation libraries(Maks R-CNN, Alpha-Pose) are not combined, requiring a different library for each purpose
  - Openpose can run on different platforms, and provide support for different hardware… etc
  - 3 different blocks; body+foot(core block), hand, and face
  - Top-down approach(face, hand)
  - Include 3D keypoint detection
  - 22 FPS (with a Nvidia GTX 1080 Ti)
  - Already used by the research community for many vision and robotics topics

Openpose

- Extended Foot Keypoint Detection
  - A small subset of foot instances out of the COCO dataset is labeled using the Clickworker platform
  - Consider 3D coordinate
  - Detect body keypoint → generate foot bbox proposals → training a foot detector

    ; suffer from the top-down problems

    implicitly help the network to more accurately predict some body keypoints(ex. leg, ankle)

Datasets and Evaluations

- Results on the MPII Multi-Person Dataset
  - Use toolkit to measure mAP of all body parts(PCKh)

    ;2D human pose estimation: new benchmark and state of the art analysis

Method	Hea	Sho	Elb	Wri	Hip	Kne	Ank	**mAP**	s/image
Subset of 288 images as in [1]									
Deepcut [1]	73.4	71.8	57.9	39.9	56.7	44.0	32.0	54.1	57995
Iqbal et al. [41]	70.0	65.2	56.4	46.1	52.7	47.9	44.5	54.7	10
DeeperCut [2]	87.9	84.0	71.9	63.9	68.8	63.8	58.1	71.2	230
Newell et al. [48]	91.5	87.2	75.9	65.4	72.2	67.0	62.1	74.5	-
ArtTrack [47]	92.2	**91.3**	80.8	71.4	**79.1**	72.6	67.8	**79.3**	**0.005**
Fang et al. [6]	89.3	88.1	80.7	75.5	73.7	**76.7**	**70.0**	79.1	-
Ours	**92.9**	**91.3**	**82.3**	72.6	76.0	70.9	66.8	79.0	**0.005**
Full testing set									
DeeperCut [2]	78.4	72.5	60.2	51.0	57.2	52.0	45.4	59.5	485
Iqbal et al. [41]	58.4	53.9	44.5	35.0	42.2	36.7	31.1	43.1	10
Levinko et al. [71]	89.8	85.2	71.8	59.6	71.1	63.0	53.5	70.6	-
ArtTrack [47]	88.8	87.0	75.9	64.9	74.2	68.8	60.5	74.3	**0.005**
Fang et al. [6]	88.4	86.5	78.6	**70.4**	74.4	**73.0**	**65.8**	76.7	-
Newell et al. [48]	**92.1**	89.3	78.9	69.8	76.2	71.6	64.7	77.5	-
Fieraru et al. [72]	91.8	**89.5**	**80.4**	69.6	**77.3**	71.7	65.5	**78.0**	-
Ours (one scale)	89.0	84.9	74.9	64.2	71.0	65.6	58.1	72.5	0.005
Ours	91.2	87.6	77.7	66.8	75.4	68.9	61.7	75.6	**0.005**

TABLE 1: Results on the MPII dataset. Top: Comparison results on the testing subset defined in [1]. Middle: Comparison results on the whole testing set. Testing without scale search is denoted as "(one scale)".

Method	Hea	Sho	Elb	Wri	Hip	Kne	Ank	**mAP**	s/image
Fig. 6b	91.8	**90.8**	80.6	69.5	78.9	71.4	63.8	78.3	362
Fig. 6c	92.2	90.8	80.2	69.2	78.5	70.7	62.6	77.6	43
Fig. 6d	92.0	90.7	80.0	69.4	78.4	70.1	62.3	77.4	0.005
Fig. 6d (sep)	**92.4**	90.4	**80.9**	**70.8**	**79.5**	**73.1**	**66.5**	**79.1**	**0.005**

TABLE 2: Comparison of different structures on our custom validation set.

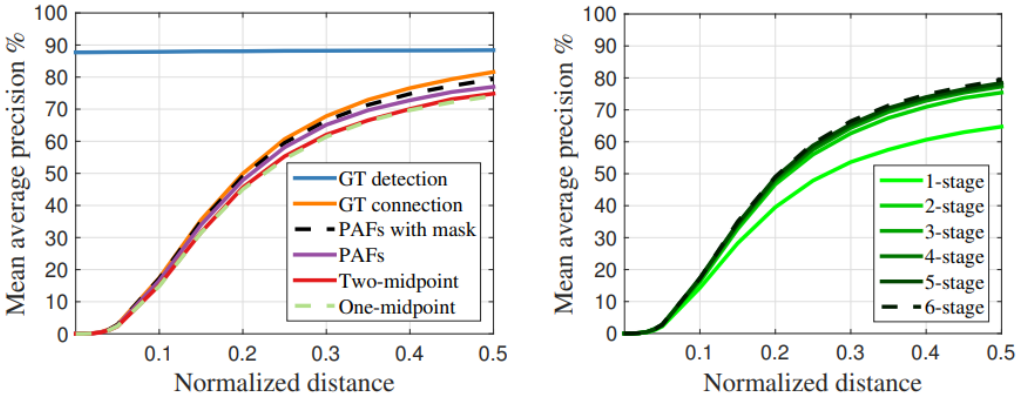


Fig. 11: mAP curves over different PCKh thresholds on MPII validation set. (a) mAP curves of self-comparison experiments. (b) mAP curves of PAFs across stages.

Datasets and Evaluations

- Results on the COCO Keypoints Challenge

Team	**AP**	AP^{50}	AP^{75}	AP^{M}	AP^{L}
Top-Down Approaches					
Megvii [43]	78.1	94.1	85.9	74.5	83.3
MRSA [44]	76.5	92.4	84.0	73.0	82.7
The Sea Monsters*	75.9	92.1	83.0	71.7	82.1
Alpha-Pose [6]	71.0	87.9	77.7	69.0	75.2
Mask R-CNN [5]	69.2	90.4	76.0	64.9	76.3
Bottom-Up Approaches					
METU [50]	70.5	87.7	77.2	66.1	77.3
TFMAN*	70.2	89.2	77.0	65.6	76.3
PersonLab [49]	68.7	89.0	75.4	64.1	75.5
Associative Emb. [48]	65.5	86.8	72.3	60.6	72.6
Ours	64.2	86.2	70.1	61.0	68.8
Ours [3]	61.8	84.9	67.5	57.1	68.2

TABLE 3: COCO test-dev leaderboard [73], "*" indicates that no citation was provided. Top: some of the highest top-down results. Bottom: highest bottom-up results.

Method	**AP**	AP^{50}	AP^{75}	AP^{M}	AP^{L}
GT Bbox + CPM [20]	62.7	**86.0**	69.3	58.5	70.6
SSD [74] + CPM [20]	52.7	71.1	57.2	47.0	64.2
Ours [3]	58.4	81.5	62.6	54.4	65.1
+ CPM refinement	61.0	84.9	67.5	56.3	69.3
Ours	**65.3**	85.2	**71.3**	**62.2**	**70.7**

TABLE 4: Self-comparison experiments on the COCO validation set. Our new body+foot model outperforms the original work in [3] by 6.9%.

Method	**AP**	AP^{50}	AP^{75}	AP^{M}	AP^{L}	Stages
5 PAF - 1 CM	**65.3**	85.2	71.3	62.2	**70.7**	6
4 PAF - 2 CM	65.2	**85.3**	**71.4**	62.3	70.1	6
3 PAF - 3 CM	65.0	85.1	71.2	**62.4**	69.4	6
4 PAF - 1 CM	64.8	**85.3**	70.9	61.9	69.6	5
3 PAF - 1 CM	64.6	84.8	70.6	61.8	69.5	4
3 CM - 3 PAF	61.0	83.9	65.7	58.5	65.3	6

TABLE 5: Self-comparison experiments on the COCO validation set. CM refers to confidence map, while the numbers express the number of estimation stages for PAF and CM. *Stages* refers to the number of PAF and CM stages. Reducing the number of stages increases the runtime performance.

Datasets and Evaluations

- Inference Runtime Analysis

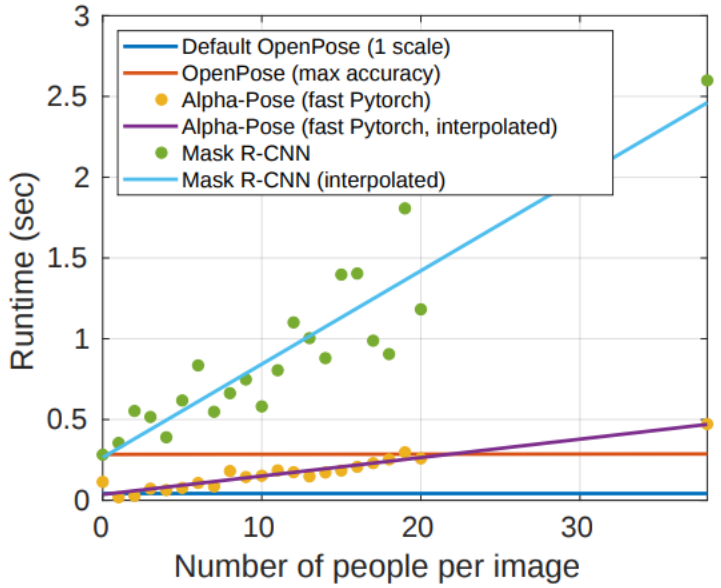


Fig. 12: Inference time comparison between OpenPose, Mask R-CNN, and Alpha-Pose (fast Pytorch version). While OpenPose inference time is invariant, Mask R-CNN and Alpha-Pose runtimes grow linearly with the number of people. Testing with and without scale search is denoted as "max accuracy" and "1 scale", respectively. This analysis was performed using the same images for each algorithm and a batch size of 1. Each analysis was repeated 1000 times and then averaged. This was all performed on a system with a Nvidia 1080 Ti and CUDA 8.

Method	CUDA	CPU-only
Original MPII model	73 ms	**2309 ms**
Original COCO model	74 ms	2407 ms
Body+foot model	**36 ms**	10396 ms

TABLE 6: Runtime difference between the 3 models released in OpenPose with CUDA and CPU-only versions, running in a NVIDIA GeForce GTX-1080 Ti GPU and a i7-6850K CPU. MPII and COCO models refer to our work in [3].

Datasets and Evaluations

- Trade-off between Speed and Accuracy

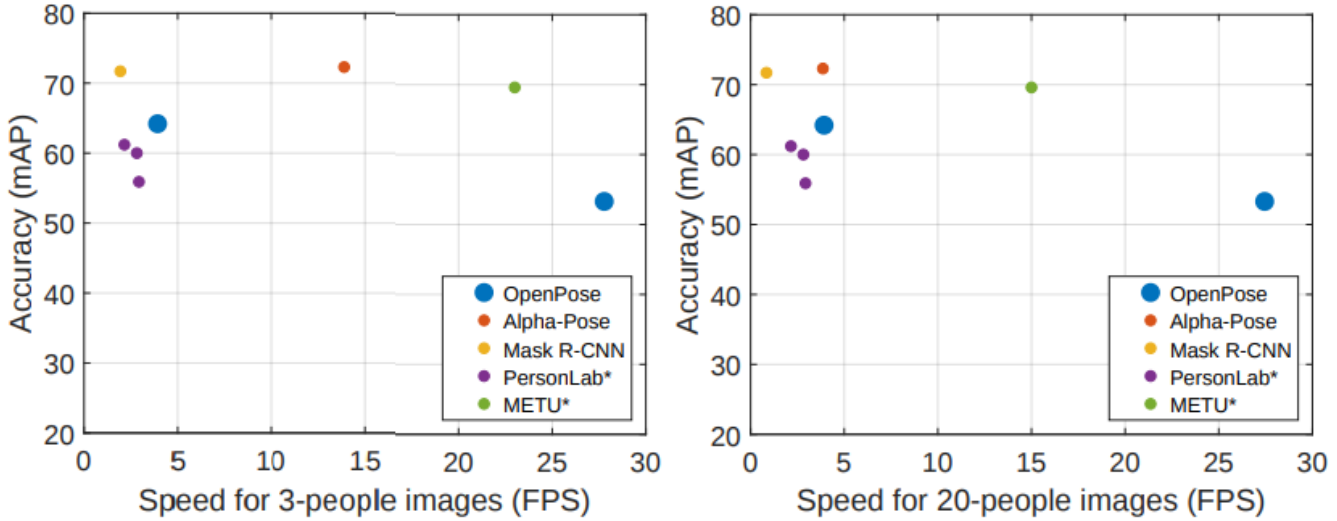


Fig. 13: Trade-off between speed and accuracy for the main entries of the COCO Challenge. We only consider those approaches that either release their runtime measurements (methods with an asterisk) or their code (rest). Algorithms with several values represent different resolution configurations. AlphaPose, METU, and single-scale OpenPose provide the best results considering the trade-off between speed and accuracy. The remaining methods are both slower and less accurate than at least one of these 3 approaches.

Datasets and Evaluations

- Results on the Foot Keypoint Dataset

Method	**AP**	**AR**	AP^{75}	AR^{75}
Body+foot model (5 PAF - 1 CM)	77.9	82.5	82.1	85.6

TABLE 7: Foot keypoint analysis on the foot validation set.

Method	**AP**	AP^{50}	AP^{75}	AP^{M}	AP^{L}
Body-only (5 PAF - 1 CM)	65.2	85.0	70.9	62.1	70.5
Body+foot (5 PAF - 1 CM)	**65.3**	**85.2**	**71.3**	**62.2**	**70.7**

TABLE 8: Self-comparison experiments for body on the COCO validation set. Foot keypoints are predicted but ignored for the evaluation.

Datasets and Evaluations

- Vehicle Pose Estimation

Method	**AP**	**AR**	AP^{75}	AR^{75}
Vehicle keypoint detector	70.1	77.4	73.0	79.7

TABLE 9: Vehicle keypoint validation set.

- Failure Case Analysis
  - Non typical poses and upside-down examples
  - Body occlusion; dataset annotations in which occluded keypoints are not included
  - Animals and statues

Conclusion