Simple, efficient and effective keypoint tracking

Rohit Girdhar[1]*, Georgia Gkioxari[2], Lorenzo Torresani[2,3]*, Deva Ramanan[1], Manohar Paluri[2], Du Tran[2]
[1]The Robotics Institute, Carnegie Mellon University [2]Facebook [3]Dartmouth College

Abstract

We propose a simple, efficient and effective approach to multi-person body keypoint tracking in videos. Our proposed method builds upon the state-of-the-art single image pose estimation systems (e.g. Mask-RCNN), and adds a lightweight tracking module on top of the frame level predictions to generate keypoint predictions linked in time. We conduct extensive ablative experiments on the newly released multi-person video pose estimation benchmark, PoseTrack, to validate various design choices of our model. Our final model achieves an accuracy of 55% on the validation and 51.8% on the test set using the Multi-Object Tracking Accuracy (MOTA) metric, and won **first** *position in the ICCV'17 PoseTrack keypoint tracking challenge*[1].

1. Introduction

Human keypoint estimation and tracking are important problems in computer vision with various downstream applications, including human action recognition, human-object interaction, etc. Recently proposed methods [3, 5, 8, 12] use deep CNNs to regress for human keypoints, and have shown strong performance on standard benchmarks such as MPII Pose [1] and COCO [11]. While previous methods have mostly focused on single-image keypoint estimation, there has been relatively less work in extending those ideas to continuous video streams. Most recent video pose estimation methods [7, 9, 13] have used used hand-designed graphical models or integer program optimizations on top of frame-based keypoint predictions to compute the final predictions over videos. While such approaches have shown good performance, they require hand-coding of optimization constraints and may not be scalable beyond short video clips due to the computational complexity of IP optimization.

In this work, we propose a much simpler, yet powerful approach to multi-person keypoint tracking in videos. Instead of concurrently optimizing the keypoint locations over space and time as have been done in some of previous works [7, 9, 13], we split it into two stages: frame-level keypoint estimation and tracking the keypoints over frames. This reduces the complexity of our model significantly as we only deal with a single frame for keypoint estimation, allowing us to use very deep state-of-the-art models (e.g. ResNet 101). Our tracking stage employs a lightweight frame-by-frame optimization, allowing our model to be scalable to virtually any length videos. We perform extensive ablation experiments of various design choices of our model, and show that it is a strong baseline, competitive against more complex models involving heavy optimization machinery [7, 9].

2. Related Work

Our proposed approach is related to previous works involving human pose estimation and tracking, as described next.

Multi-person pose estimation in images: The application of deep convolutional neural networks (CNNs) to keypoint estimation has shown significant improvements over previous non-deep approaches [14]. Most recent approaches to multi-person keypoint estimation in still images can be classified into a bottom-up or a top-down architecture. Top-down approaches [5, 12] involve first detecting a human bounding box, followed by estimation of the body joint keypoints within that box. In contrast to the top-down approaches, bottom-up methods like [3, 8] involve detecting individual keypoints, and in some cases the affinities between those keypoints, and then splitting those predictions into keypoints belonging to specific persons in the image.

Multi-person pose estimation in videos: The naive solution to video pose estimation is to apply image-based models as described above to each frame of the video. While this obtains strong performance, it misses out on two key factors. First, it does not exploit the additional temporal information in videos that can help disambiguate a lot of keypoints, specially in frames with large amounts of motion or occlusions. And second, it does not solve the problem of assigning identity to each of those keypoints. In videos with multiple interacting people, understanding trajectories of individual

*Work done as a part of RG's internship at Facebook
[1] https://posetrack.net/iccv-challenge/

keypoints belonging to a person can be very useful in understanding actions and interactions. A recent approach, Thin-Slicing networks [13], proposes a model combining a CNN and a CRF to jointly optimize frame-level keypoint estimator with the CRF that smooths the predictions over space and time. Although their results showed an improvement over the frame-level predictions, their method does not directly address keypoint tracking problem. Other recent approaches [7, 9] have also been proposed for video pose estimation by solving an integer program optimization on top of single frame predictions. While these methods can handle both space-time smoothing and identity assignment, it is difficult to apply these methods to long videos due to the NP-hardness of the Integer Programming optimization.

3. Our Approach

We propose a two-stage approach for human keypoint tracking in videos. First, we compute frame level keypoint estimates using a CNN based model. Although our approach can use any frame-based keypoint estimation system, in this submission, we use the Mask R-CNN [5] framework due to its simple formulation and robust predictions. Mask R-CNN is a top-down keypoint estimation model that builds upon Faster R-CNN object detection framework. It consists of a standard base CNN, typically a ResNet [6] variant to extract features from the images. These features are then used to regress for potential boxes in the image containing humans using a Region Proposal Network (RPN). The obtained boxes are then used to crop the previously extracted feature maps by an operation known as RoIAlign, which scales a region of the feature map to a fixed size. This transformed feature map is then passed through two "head" CNNs, one of which is to classify and regress a tight bounding box around the person, and the other is to compute joint heatmaps of the person.

Given these keypoint predictions linked in space by person identity (i.e. pose estimation), we need to link them in time for tracking. We represent these detections as a graph, where every detected person bounding box in every frame is a node. We define edges to connect each box in a frame to each box in the next frame. The cost of each edge is defined as the negative likelihood of the two boxes linked on that edge to belong to the same person. To compute tracks, we simplify the problem to bipartite matching between a pair of frames, and propagate the labels forward, one frame at a time, starting from the first frame to the last. Any boxes that do not get matched to an existing track, including all boxes in the first frame, instantiate a new track. As shown in Sec. 4, this simple approach is very effective in getting good tracks, is highly scalable, and can run on any length videos. We present more details of our approach, including the definition of edge cost and matching algorithm in Sec. 4.

4. Experiments and Results

We now experimentally evaluate the various design choices of our final submission. Unless otherwise specified, we perform all evaluations on the validation subset of the PoseTrack dataset. Due to the gradual release of the PoseTrack dataset, some of the experiments were performed on an initial subset of the complete dataset, which we specify with each set of results. For test submissions, we trained our models on the final train+val sets and report results as obtained by submitting the predictions to the challenge evaluation server.

4.1. Multi-frame pose model

We build upon the state-of-the-art frame-level pose estimation system, Mask-RCNN [5]. We extend that system to a video input by taking inspiration from recent work, I3D, inflating 2D CNN models for action recognition [4]. We inflate the ResNet backbone of the network using 3D filters to handle 3D input (multiple frames), and collapse the final feature map back to 2D by averaging across time. The object classification and keypoint prediction heads then operate on the RoIAlign output of the 2D feature map. We compare this model on frame-level keypoint prediction task using the mAP metric in Tab. 1. We experimented with two types of weight inflations: "center", in which we set the center slice of the 3D kernel with the 2D kernel; and "mean", where we replicate the 2D kernel T times and divide by T (as used in [4]). All models were initialized with a model trained on the COCO [11] dataset and then finetuned on PoseTrack. We observed best performance with the 2D model, and thus use that for further experiments.

4.2. Thresholding initial detections

Before we can track the detections through the video, we would want to drop the incorrect detections. This helps prevent the tracks from drifting and reduces false positives. Tab. 2 shows the effect of thresholding the detections. As expected, the MOTA [2] tracking metric improves with higher thresholds, while the keypoint mAP performance decreases due to missing out on certain detections. Since we primarily focus on the tracking task, we threshold our detections at 0.95 for our final model.

4.3. Deeper frame-level model

As with most vision problems, we noticed an improvement in frame level pose estimation by moving to a deeper model. The improvements also directly transferred to the tracking performance. Replacing ResNet-50 in Mask-RCNN with Resnet-101 gave us about 2% improvement. Thus, we use Resnet-101 in our submission and all later experiments, otherwise stated.

Table 1. **Comparison between 2D model with 3D inflated models**. Performance of 2D model vs. 3D inflated models using mAP metric on v0.7 validation release. All models are based on Resnet50.

Model	Init	Head	Shou	Elbo	Wri	Hip	Knee	Ankle	Total
ResNet-50	-	69.6	73.6	60.0	49.1	65.6	58.3	46.0	60.9
ResNet-50 3D	center	69.8	73.6	59.9	49.1	65.5	58.2	45.6	60.9
ResNet-50 3D	mean	66.2	68.6	54.3	42.3	61.3	52.3	40.7	55.8

Table 2. **Effect of the detection cut-off threshold**. Effect of thresholding initial detections before matching the detections to compute tracks, on pose mAP and tracking MOTA. While mAP goes down, the MOTA increases as there are fewer spurious detections to confuse the tracker. On v0.7 release with Res-50 base.

Threshold	**mAP Total**	mAP Head	mAP Shou	mAP Elbo	mAP Wri	mAP Hip	mAP Knee	mAP Ankl	MOTA Head	MOTA Shou	MOTA Elb	MOTA Wri	MOTA Hip	MOTA Knee	MOTA Ankl	**MOTA Total**	MOTP Total	Prec Total	Rec Total
Dummy tracks	61	69.7	73.9	60.1	49	66	58	45.8	26.5	35.2	1.9	-19.5	18.9	0.4	-23.2	7.1	82.8	53	77.6
0.0	61	69.7	73.9	60.1	49	66	58	45.8	21	29.7	-3.3	-24.8	14	-4.5	-28.5	1.9	82.7	53	77.6
0.5	61	69.7	73.9	60.1	49	66	58	45.8	22.1	30.8	-2.3	-24	15	-3.5	-27.8	2.8	82.6	53	77.6
0.9	58.3	66	69.8	57.8	47.5	63.4	56	44	48.1	54.5	31.1	12.6	43.4	28.9	8.1	33.4	82.9	67.4	72.1
0.95	56.3	63.1	66.8	55.9	46.1	61.1	54.6	42.9	49.6	55.4	35.2	18.5	45.8	34	14.8	37.1	83.1	70.9	69

4.4. Matching algorithm

We experimented with two bipartite matching algorithms: the Hungarian algorithm [10] and a greedy algorithm. While the Hungarian algorithm computes a optimal matching given a edge cost matrix, the greedy algorithm takes inspiration from the evaluation algorithms for object detection and tracking. We start from the highest confidence match, select the edge and remove the two connected nodes out of consideration. The process of connecting each predicted box in the current frame with previous frame is repeatedly applied from the first to the last frame of the video. Table 3 compares the two algorithms, using a "bounding box overlap" cost function (details in Sec. 4.5). We observe that the Hungarian method performs slightly better, thus we use it as our final submission model.

4.5. Cost criterion

We experimented with three cost criteria. First, we use bounding box overlap over union (IoU) as the similarity metric. This metric expects the person to move and deform little from one frame to next, and boxes should mostly overlap. Second, we used pose PCKh [14] as the similarity metric, as the pose of the same person would change only a little between consecutive frames. Finally, we used cosine similarity between CNN features extracted from the image cropped using the person bounding box as a similarity metric. Specifically, we use ResNet-18 pre-trained on ImageNet. Tab. 4 shows that the performance is relatively stable across different cost criteria. We also experimented with different layers of the CNN but obtained similar performance. Combinations of the different cost criteria did not give an improvement either. For simplicity, we use the bounding box overlap cost for our final model.

4.6. Dropping low-confidence keypoint predictions

We observed that the MOTA criterion used in the challenge to evaluate tracking is not sensitive to the score values of each keypoint. Since our frame-level keypoint predictor produces a confidence score apart from the location, we can use that information to drop keypoints we are not confident about. We cross-validate for the threshold and obtain about 10% improvement in validation performance by dropping low-conf keypoints (Tab. 5).

4.7. Comparison with state of the art

Finally, we compare our approach to previously published work on this dataset. Since this data was only released in full few weeks before the challenge, there do not exist published baselines on the complete dataset. However, a previous work [9], from the authors of the challenge, reports results on a subset of this data. We compare our performance on a mini test set to their reported performance in Tab. 5. We note that the numbers are not directly comparable. Our final performance on the full test set was 51.8 (revealed during the workshop), and won first position in the challenge.

5. Conclusion

We have presented a simple, yet efficient approach to human keypoint tracking in videos. Our approach combines the state-of-the-art frame-level pose estimation with a fast and effective person-level tracking module to connect keypoints over time. Through extensive ablative experiments, we explore different design choices for our model, and present strong results on the PoseTrack challenge benchmark.

Table 3. **Comparison between Hungarian and Greedy algorithm for matching**. Effect of matching algorithm in tracking performance on v0.75 validation set. All numbers computed at 0.95 initial detection threshold using bounding-box overlap cost criterion.

Method	MOTA Head	MOTA Shou	MOTA Elb	MOTA Wri	MOTA Hip	MOTA Knee	MOTA Ankl	MOTA Total	MOTP Total	Prec Total	Rec Total
Hungarian	55.4	60.4	44.6	29.3	50.8	43.4	25.9	45	83.9	77.7	68.9
Greedy	57.1	62.9	44.2	28.1	50.8	41.8	22.9	44.9	83.8	75.2	72.5

Table 4. **Comparison between different similarity cost criteria**. Impact of cost criterion on keypoint tracking performance. v0.75 release at 0.95 inital detection threshold with R-101 base model.

Method	MOTA Head	MOTA Shou	MOTA Elb	MOTA Wri	MOTA Hip	MOTA Knee	MOTA Ankl	MOTA Total	MOTP Total	Prec Total	Rec Total
Bbox overlap	57.2	63.0	44.2	28.2	50.9	42.0	23.0	45.0	83.8	75.2	72.5
Pose PCK	56.2	61.9	43.2	27.1	49.8	40.8	21.8	43.9	83.8	75.2	72.5
CNN cos-dist	57.4	63.1	44.4	28.4	51.0	42.1	23.2	45.1	83.8	75.2	72.5

Table 5. **Effect of thresholding detected keypoints**. Thresholding the predicted keypoints used for MOTA evaluation gives a significant improvement in performance, as MOTA does not use score values. We also compare our method with the previously reported method on a subset of this dataset [9]. Performance reported with R101 base model.

Method	Dataset	MOTA Head	MOTA Shou	MOTA Elb	MOTA Wri	MOTA Hip	MOTA Knee	MOTA Ankl	MOTA Total	MOTP Total	Prec Total	Rec Total
Without drop	Val v0.75	57.2	63.0	44.2	28.2	50.9	42.0	23.0	45.0	83.8	75.2	72.5
Drop threshold=1.95	Val v0.75	61.5	65.2	57	45.6	54	52.8	45.3	55	84.5	87.9	66.3
Final model	(Mini) Test v1.0	55.9	59.0	51.9	43.9	47.2	46.3	40.1	49.6	34.1	81.9	67.4
PoseTrack [9]	Test (subset)	-	-	-	-	-	-	-	28.2	55.7	64.8	63.0

References

[1] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *CVPR*, 2014. 1

[2] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: the CLEAR MOT metrics. *EURASIP Journal on Image and Video Processing*, 2008. 2

[3] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017. 1

[4] J. Carreira and A. Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. In *CVPR*, 2017. 2

[5] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *ICCV*, 2017. 1, 2

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2

[7] E. Insafutdinov, M. Andriluka, L. Pishchulin, S. Tang, B. Andres, and B. Schiele. Articulated multi-person tracking in the wild. In *CVPR*, 2017. 1, 2

[8] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *ECCV*, 2016. 1

[9] U. Iqbal, A. Milan, and J. Gall. Pose-track: Joint multi-person pose estimation and tracking. In *CVPR*, 2017. 1, 2, 3, 4

[10] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics (NRL)*, 1955. 3

[11] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 1, 2

[12] G. Papandreou, T. Zhu, N. Kanazawa, A. Toshev, J. Tompson, C. Bregler, and K. Murphy. Towards accurate multi-person pose estimation in the wild. In *CVPR*, 2017. 1

[13] J. Song, L. Wang, L. Van Gool, and O. Hilliges. Thin-slicing network: A deep structured model for pose estimation in videos. In *CVPR*, 2017. 1, 2

[14] Y. Yang and D. Ramanan. Articulated human detection with flexible mixtures of parts. *PAMI*, 2013. 1, 3