Supplementary Material for: Self-supervised Keypoint Correspondences for Multi-Person Pose Estimation and Tracking in Videos

Umer Rafi[1⋆], Andreas Doering[1⋆], Bastian Leibe[2], and Juergen Gall[1]

[1] University of Bonn, Germany
[2] RWTH Aachen, Germany

A Pose Estimation Framework

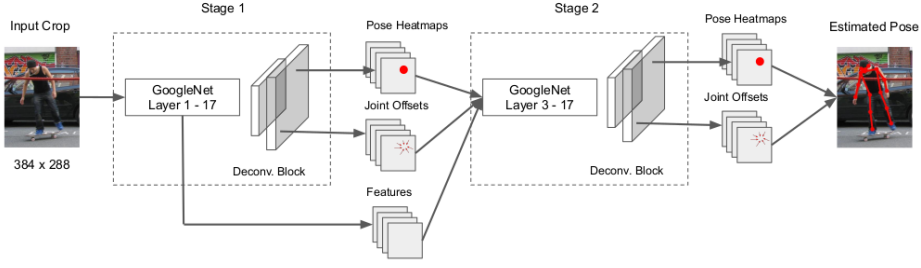

Fig. 1. Our two-stage pose estimation framework. Each stage uses GoogleNet [1] as the backbone. The features extracted by the backbone in the first stage are fed into a deconvolution layer block to produce pose and joint offsets maps. The backbone features, pose heatmaps and joint offsets maps from the first stage are fed into the second stage to produce refined pose and joint offsets maps.

Our two-stage pose estimation framework is shown in Figure 1. Each stage uses a GoogleNet [1] as the backbone. We use layer 1 to layer 17 for the backbone of the first stage while for the second stage we use layer 3 to layer 17 only. The features extracted by the backbone in the first stage are fed into a deconvolution layer block to produce pose and joint offset maps. The backbone features, pose heatmaps and joint offset maps from the first stage are fed into the second stage to produce refined pose and joint offset maps.

Due to pooling used in the backbone, the resolution of the pose heatmaps is reduced by a factor of 4 in height and width dimensions. Consequently, the

⋆ equal contribution

Table 1. Impact of τ_{corr} on mAP and MOTA on the PoseTrack 2017 validation set.

τ_{corr}	MOTA	mAP
0.1	67.9	77.9
0.2	67.9	77.9
0.3	**67.9**	**78.0**
0.4	67.9	78.0
0.5	67.8	78.0

Table 2. Comparison of mAP and MOTA for different design choices on the PoseTrack 2017 validation set.

Design Choices	MOTA	mAP	IDSW
Correspondence Tracking	67.9	78.0	3632
Correspondence Tracking w/o refinement module	66.9	77.7	4304
Correspondence Tracking w/o duplicate removal	64.5	77.9	8288

up-sampled predicted pose is slightly away from the actual pose. Towards this end, we append a joint offset head to predict the deltas, *i.e.*, Δx and Δy for each keypoint. The position of the jth keypoint $(\hat{x}_j, \hat{y}_j)$ at inference is computed as

$$(\hat{x}_j, \hat{y}_j) = (x_j + \Delta x_j,\ y_j + \Delta y_j). \tag{1}$$

where (x_j, y_j) is the up-sampled position from the pose heatmaps. During training, we minimize the L1 loss between the predicted and ground-truth deltas for the joint offset maps and use the binary cross entropy loss for the pose heatmaps.

B Impact of τ_{corr}

We evaluate the impact of τ_{corr} on the pose estimation and tracking performance. As shown in Table 1, the threshold has a low impact. We use $\tau_{corr} = 0.3$ for all our experiments.

C Effect of Refinement Module and Duplicate Removal

We evaluate the effect of the refinement module and duplicate removal on the pose estimation and tracking performance. As shown in Table 2, omitting any of the introduced design choices results in a significant drop in MOTA of at least 1%, and increases the number of identity switches (IDSW). Our proposed correspondence refinement module improves the generated correspondence affinity maps which results in stronger tracking results. This is reflected by the MOTA and mAP scores that drop to 66.9 and 77.7, respectively, if we disable the refinement module. If duplicates are not removed, the MOTA and the mAP scores drop to 64.5 and 77.9, respectively.

D Track Merging

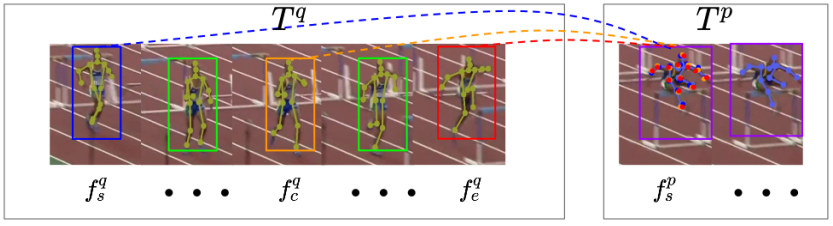

Fig. 2. Tack merging: For the start frame f_s^q, the center frame f_c^q and the last frame f_e^q of track T^q, we estimate poses from keypoint correspondences in the start frame f_s^p of T^p, as illustrated by the colored dashed lines. We use an OKS-based similarity metric to measure the average pose similarity between the poses from correspondences and the pose in the starting frame f_s^p of track T^p.

We propose a post-processing step in which we merge tracks of the same pose instance at different time steps by utilizing keypoint correspondences from multiple frames. Given two tracks T^q and T^p as illustrated in Figure 2, we select three pose instances $\{B_f^q\}$ with $f \in \{f_s^q, f_c^q, f_e^q\}$ at the start, center and end frames of track T^q. For each of the pose instances B_f^q, we compute the pose $\bar{B}_f^q$ for the starting frame f_s^p of track T^p using correspondences, as described in Section 5 of the paper. We then employ OKS as similarity metric and calculate the average similarity between tracks T^q and T^p as

$$S_{match}(T^q, T^p) = \frac{\sum_{f \in \{f_s, f_c, f_e\}} OKS(\bar{B}_f^q, B_{f_s^p}^p)}{3}. \tag{2}$$

E Failure Cases

Existing person detectors sometimes output duplicate detections for the same person. Such duplicate detections are hard to remove using non-maximum suppression. In our experiments, they increase the number of false-positives (FP) and lead to identity-switches. This impacts the overall tracking performance, as the MOTA metric used in PoseTrack heavily penalizes FPs and IDSWs as shown in Table 2. Figure 3 illustrates such failure cases.

Fig. 3. Failure cases. Duplicates by the person detector lead to multiple tracks of the same person and negatively impact the tracking performance.

F Qualitative Results

Fig. 4. Qualitative results for recovering missed detections. Best seen using the zoom function of the PDF viewer.

References

1. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: CVPR (2015)