Erschienen in: Pattern Recognition : 44th DAGM German Conference, DAGM GCPR 2022, Konstanz, Germany, September 27–30, 2022, Proceedings / Andres, Björn et al. (Hrsg.). - Cham : Springer, 2022. - (Lecture Notes in Computer Science ; 13485). - S. 513-528. - ISSN 0302-9743. - eISSN 1611-3349. - ISBN 978-3-031-16787-4
https://dx.doi.org/10.1007/978-3-031-16788-1_31

I-MuPPET: Interactive Multi-Pigeon Pose Estimation and Tracking

Urs Waldmann[1,2](✉), Hemal Naik[1,2,3,4], Nagy Máté[1,2,3,5,6], Fumihiro Kano[2,3], Iain D. Couzin[1,2,3], Oliver Deussen[1,2], and Bastian Goldlücke[1,2]

[1] University of Konstanz, Konstanz, Germany
[2] Centre for the Advanced Study of Collective Behaviour, University of Konstanz, Konstanz, Germany
urs.waldmann@uni-konstanz.de
[3] Max Planck Institute of Animal Behavior, Konstanz, Germany
[4] Technische Universität München, Munich, Germany
[5] Hungarian Academy of Sciences, Budapest, Hungary
[6] Eötvös Loránd University, Budapest, Hungary

Abstract. Most tracking data encompasses humans, the availability of annotated tracking data for animals is limited, especially for multiple objects. To overcome this obstacle, we present I-MuPPET, a system to estimate and track 2D keypoints of multiple pigeons at interactive speed. We train a Keypoint R-CNN on single pigeons in a fully supervised manner and infer keypoints and bounding boxes of multiple pigeons with that neural network. We use a state of the art tracker to track the individual pigeons in video sequences. I-MuPPET is tested quantitatively on single pigeon motion capture data, and we achieve comparable accuracy to state of the art 2D animal pose estimation methods in terms of Root Mean Square Error (RMSE). Additionally, we test I-MuPPET to estimate and track poses of multiple pigeons in video sequences with up to four pigeons and obtain stable and accurate results with up to 17 fps. To establish a baseline for future research, we perform a detailed quantitative tracking evaluation, which yields encouraging results.

Keywords: Pose estimation · Multi-object tracking · Animals · Applications

We acknowledge funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2117 - 422037984, the Office of Naval Research (Research Grant to I.D.C, #N00014-64019–1-2556) and the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement (to I.D.C; #860949). Ethics: Animal experiments were approved by the Regierungspräsidium Freiburg under the permit number Az. 35–9185.81/G-19/107. Full author affiliations are in our supplemental material.

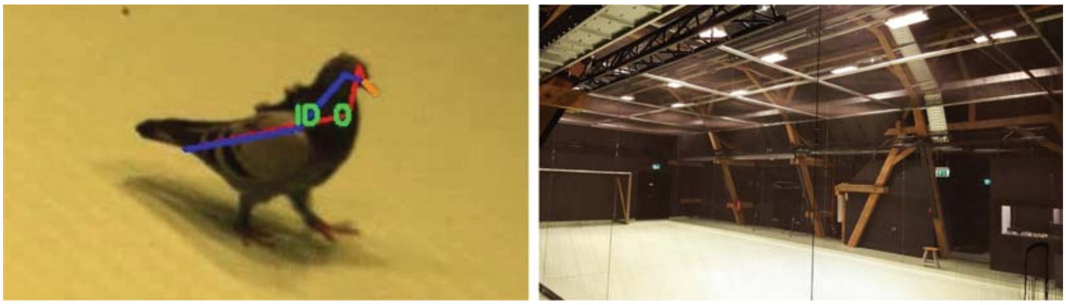

Fig. 1. *Interactive multi-pigeon pose estimation and tracking (I-MuPPET). Left*: Estimated complex pose (beak tip, nose, left and right eye, left and right shoulder and tail) of a pigeon with its ID from tracking. Left body side of pigeon in red, right one in blue and beak in orange. *Right*: Facility with cameras and Vicon motion capture system where our pigeon datasets were recorded. © CASCB Uni Konstanz.

1 Introduction

Accurate quantification of behavior is critical to understand the underlying principles of social interaction and the neural and cognitive underpinnings of animal behaviour [1,5,7,31,38]. While researchers conventionally analyzed animal behaviour manually using a predefined catalogue of behaviours called ethograms, the recent advances in computer vision, as well as the increasing demands for a large data set involving the analysis on the fine-scaled and rapidly-changing behaviours of animals, encourage automated tracking methods [2,12,18,38]. The CVPR workshop in 2021 on “Computer Vision for Animal Behavior Tracking and Modeling” [43] emphasized the increasing interest in computer vision tools in the field of animal behaviour. While existing automatic methods range from object detection [16], behavior analysis [10,42], segmentation [11], 3D shape and pose fitting [3,9] to pose estimation [19,33] and tracking [45,49], reliable tracking of multiple moving animals in real-time and estimating their pose remains a challenging task.

One of the limiting factors in the field of animal pose estimation is the small amount of annotated training data compared to its human counterpart (for example 3.6 million in [25]). DeepLabCut [38], LEAP [46] and DeepPoseKit [20] overcome this lack of training data by introducing a method to manually label few data that is then used to train a neural network. With that network the authors predict body parts of additional unlabeled material creating more and more annotated training data for animals. Creatures Great and SMAL [9] instead creates synthetic silhouettes for training and extracts silhouettes [52,53] from real data for inference. We are aware of only three data sets for birds [3,50,54]. Clearly, methods need to be developed that exploit few training data in an efficient way.

In this paper, we present I-MuPPET, an interactive multi-pigeon pose estimation and tracking system. We can acquire training data for a single pigeon in a semi-automated way and demonstrate that training on an annotated dataset containing only a single pigeon is sufficient for our framework to predict seven keypoints of a complex pose for multiple pigeons and track the individuals at interactive speed ($\geqslant 1$ fps). We track up to four pigeons (at the moment the upper limit in our dataset) with $12-17$ fps, and report detailed results for speed and accuracy. Our framework is comparable with state of the art 2D animal pose estimation methods in terms of Root Mean Square Error (RMSE) .

2 Related Work

2.1 Animal Pose Estimation

2D Single Animal Pose Estimation. With the huge success of DeepLabCut [38] and LEAP [46], animal pose estimation has been developing into its own research branch parallel to human pose estimation. DeepLabCut and LEAP both introduce a method for labelling animal body parts and training a deep neural network for predicting 2D body part positions. DeepPoseKit [20] improved the inference speed by a factor of approximately two, while maintaining the accuracy of DeepLabCut. In 3D Bird Reconstruction [3], they predict 2D keypoints to estimate the pose and shape of cowbirds from a single view. Since manual annotations are time-consuming, labor-intensive, and prone to errors, we use a framework that uses semi-automatically labeled data.

2D Multi-Animal Pose Estimation. DeepLabCut is extended in [34] to predict 2D body parts of multiple animals. This extension uses training data with annotations of multiple animals. The authors will release four datasets with annotations containing mice, pups, marmosets and fish. Similarly SLEAP [47] provides several architectures to estimate 2D body parts of multiple animals. These two approaches [34,47] work well but are trained on multi-animal annotated data. Since this kind of data is limited in its availability, we overcome this limitation by training a framework on annotated single animal data and still predict complex poses of multi-animal video sequences at interactive speed.

3D Animal Pose Estimation. In [15] Dunn *et al.* use a 3D CNN similar to [26] to infer 3D poses of single rodents from multi-view. This approach comes at the cost of longer run times. In [4,21,28,30,40] the authors use a 2D pose estimator (e.g. [38,41]) to predict 2D keypoints that they then triangulate to 3D. We notice that all these 3D frameworks exploit 2D keypoints, and can thus also use our method as a base.

2.2 Animal Tracking

Romero-Ferrero *et al.* in [49] and Heras *et al.* in [24] use the software idtracker.ai [17] to track up to 100 zebrafish at once. The software needs to know the number of individuals beforehand since it performs an individual identification in each frame. Idtracker.ai does not predict keypoints of the individuals, whereas TRex [51] estimates 2D head and rear of bilateral animals while tracking up to 256 individuals in real-time using background subtraction.

2D Animal Pose Estimation and Tracking. DeepLabCut [38] is further extended in [34] to track multiple mice, pups, marmosets and fish. The authors split the workflow in local and global animal tracking. For local animal tracking they build on SORT [8], a simple online tracking approach. For animals that are closely interacting or in case of occlusions they introduce a global tracking method by optimizing the local tracklets with a global minimization problem using multiple cost functions on the basis of the animals' shape or motion for example. In contrast to this work, we focus on online tracking thus using only SORT [8]. In principal our method can also be post-processed to optimize the local tracklets obtained from SORT [8].

SLEAP [47] uses a tracker based on Kalman filter or flow shift inspired by [55] for candidate generation to track multiple individuals. As mentioned beforehand they also do 2D multi-animal pose estimation.

3 Technical Framework

We will explain the data acquisition of our pigeon data, introduce briefly the two datasets with which we train I-MuPPET in order to compare our framework to [38] and [3] (cf. Sects. 4.2 and 4.3), describe the technical framework behind I-MuPPET and discuss several ablation studies.

3.1 Datasets

Data Acquisition. Our pigeon data is recorded with a Vicon motion capture system. The system consists of six Vue 2, four Vantage 5 and 26 Vero 2.2 sensors covering a volume of approximately 15 × 7 × 4 meters, see Fig. 1 and the supplemental material. Two Vue cameras were used to record the RGB video sequences of single pigeons, while the 30 infrared sensors captured the positions of the pigeons within an area of approximately 5 × 5 meters.

Single Pigeon Data. In total we have 27730 annotated RGB frames (13532 frames from one camera view, 14198 from the other) with a resolution of 1920 × 1080 × 3 pixels available on which only one single pigeon is present (cf. Figure 2, first row). The annotated frames contain the 2D positions of seven distinct body landmarks (beak tip, nose, left and right eye, left and right shoulder and tail, cf. Figure 1) plus the coordinates of a bounding box containing the object. These annotations are obtained in a semi-automatic manner (cf. [39]). For more details regarding our data see our supplemental material.

Multi-Pigeon Data. In addition, we have RGB video sequences of multiple pigeons available (cf. Figure 2, second row). At the moment we do not have ground truth annotations for individual keypoints of the multi-pigeon video sequences. We do have bounding boxes for the multi-pigeon sequences that we use as ground truth for a quantitative tracking evaluation (cf. Sect. 4.4). To

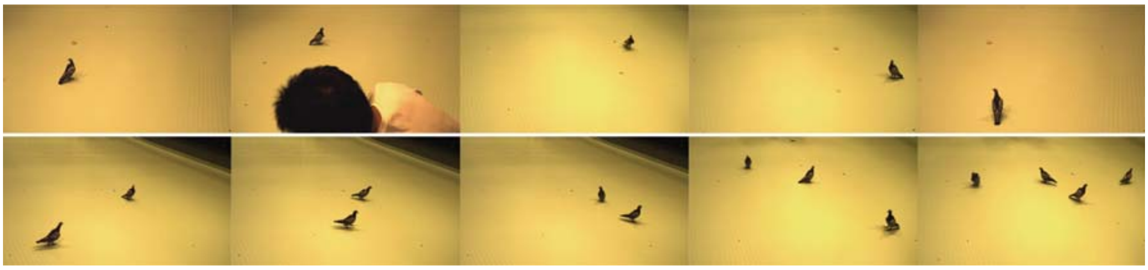

Fig. 2. *Pigeon Data.* Sample frames of our data set. First row shows images from our annotated single pigeon data set. Second row from our multi-pigeon data. Best viewed in color version online.

obtain these bounding boxes we perform a simple background subtraction and validate the bounding boxes with the 3D Vicon positions of the pigeons projected into the camera images.

Data set (upon request) along with accompanying source code to reproduce the results of this paper are publicly available at https://urs-waldmann.github.io/i-muppet/.

Odor Trail Tracking Data. This data from [38] contains single mice following an odor and contains 1080 manually annotated samples. The samples are random, distinct frames from multiple sessions observing seven different mice [38] and the resolution of the images is 640 × 480 or 800 × 800 since the data were recorded with two different monochromatic cameras.

Cowbird Data. This data from [3] contains single cowbirds. Their original images have a maximum resolution of 1920 × 1200 containing multiple birds. For 2D pose estimation they use 1000 cropped samples of single individuals from a subset of 18 moments across six of the 10 days [3] with a resolution of 256 × 256.

For more details on these two datasets we refer to [3,38]. We use them to train I-MuPPET in order to compare the accuracy in pose estimation to that of [38] and [3].

3.2 Pose Estimation and Tracking

The core components of our framework are a Keypoint R-CNN [22] and the SORT tracker [8], see Fig. 3.

The Keypoint R-CNN is a PyTorch [44] implementation of a Mask R-CNN [22], which is modified to output seven keypoints (beak tip, nose, left and right eye, left and right shoulder and tail) for each detected instance, in addition to a score, label (background vs. object) and bounding box. Like DeepLabCut [38], our network has a ResNet-50-FPN [23,36] backbone that was pretrained on ImageNet [14], similar to [22]. For details, we refer to [22]. The input to the network are RGB images (cf. Figure 3) normalized to mean and standard deviation of 0.5. The network is trained in a fully supervised manner using

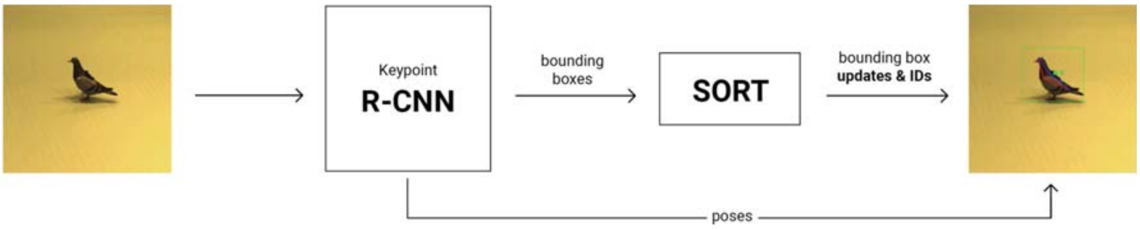

Fig. 3. *I-MuPPET.* For inference the input image (here one pigeon cropped for better view) is passed to the Keypoint R-CNN [22] that predicts bounding boxes and poses of all instances. The bounding boxes are passed to SORT [8] that returns bounding box updates with their associated ID.

stochastic gradient descent with learning rate decay, momentum and weight decay. For training, the network expects ground truth labels, bounding boxes and keypoints in addition to the normalized RGB images.

We also implement data augmentation for training in order to avoid overfitting and to mimic other conditions than those present in the single pigeon data. This expands the training set, which turns out to also lead to better results in multi-pigeon pose estimation when trained on data with only single pigeons present. Specifically, our input data has a specific probability to be flipped, scaled within a specified range, and changed in brightness or sharpness.

The SORT tracker [8] accepts the bounding boxes from all pigeon instances in every frame that exceed a given score threshold and outputs updated bounding boxes with their associated ID using a combination of Kalman Filter [29] and Hungarian algorithm [32]. We chose this method since we are primarily interested in online tracking and high inference speed, and SORT [8] can run up to 260 fps. We use standard parameters and refer to [8] for details.

3.3 Network Training and Ablation Studies

Data Augmentation for Pigeons. For data augmentation we find that changing brightness, flipping or scaling do not enhance performance, but changing sharpness with a probability of 0.2 results in the best performance (for numbers cf. supplemental material). This is intuitive since we train and test on the single pigeon data (cf. Figure 2, first row) where the training data already contains a wide range of different pigeon positions and lightning conditions, and thus covers most of the scaling and brightness. Also the training data already include most body orientations (with respect to the camera), thus flipping does not improve test accuracy. Since the depth of field of the cameras is limited the pigeons are sometimes slightly out of focus and therefore blurring the input image with a small probability of 0.2 improves the accuracy of the test set.

In case of multi-pigeon video sequences, however, we find that the best data augmentation parameters are not the same as for the single pigeon data set. We keep the parameters from the single pigeon analysis but find that randomly jittering brightness by a factor chosen uniformly from $[0.4, 1.6]$ and a flipping probability of 0.5 is best. This is intuitive because the single pigeon data (cf.

Figure 2, first row) does not cover the range of brightness found in the multi-pigeon data (cf. Figure 2, second row) plus the flipping makes the pose estimation in new situations more robust. A small scaling range of ±5% is sufficient since the single pigeon data covers already a large range of pigeon sizes. Also, if the scaling range is too large, we find multiple (mis-)detections if pigeons are nearby. This is also the case in situations where the pigeons occlude or are close to each other even if we do not apply scaling.

Data Augmentation for Cowbirds. The cowbird data set is recorded in outdoor aviaries [3]. Thus different day light and season conditions are present. To consider these different conditions inherent in the data, we use different data augmentation parameters. We find that randomly changing brightness by a factor chosen uniformly from $[0.7, 1.3]$, and a sharpness probability of 0.1, works best (for numbers cf. supplemental material).

Training Hyperparameters. To find out the best network configuration for I-MuPPET we perform several experiments (see supplemental material). From this analysis we find that using a learning rate of 0.005 and reducing it by $\gamma = 0.5$ every given step size to reach a final learning rate of 0.0003 at the end of training works best.

4 Evaluation

We quantitatively evaluate I-MuPPET on our annotated single pigeon data (RMSE in Sect. 4.2, PCK in Sect. 4.3). In addition we evaluate our framework on the odor trail tracking data set from [38] and the cowbird data set from [3]. In this way we can compare the performance of I-MuPPET to the 2D pose estimators used in [3,38]. We also evaluate the I-MuPPET tracking performance in terms of accuracy, precision and speed on a workstation with an nVidia Titan RTX, 64 GB DDR4 RAM, an Intel Xeon E5-2620 at 2.10 GHz and a 2 TB Samsung SSD 850.

4.1 Metrics

Pose Estimation. Two widely used metrics, also in human pose estimation, are the Root Mean Square Error (RMSE), in human pose estimation better known as Mean Per Joint Position Error (MPJPE, cf. e.g. [26]), and the Percentage of Correct Keypoints (PCK, cf. e.g. [56]). DeepLabCut [38] uses the former, 3D Bird Reconstruction [3] the latter. Note that PCK properly takes into account scale, and thus this accuracy measure is more meaningful than RMSE. Both metrics assume that all keypoints in all frames can be predicted.

Tracking. There are three sets of tracking performance measures that are widely used in the literature [13]: the CLEAR-MOT metrics introduced in [6], the metrics introduced in [35] to measure track quality, and the trajectory-based metrics proposed in [48]. Additionally, we report the new Higher Order Tracking

Accuracy (HOTA). It was introduced in [37] because the other metrics overemphasize the importance of either detection or association. HOTA measures how well the trajectories of matching detections align, and averages this over all matching detections, while also penalising detections that do not match [37].

For further details we refer to [13,37]. We use [27] for evaluation.

4.2 Comparison with DeepLabCut

DeepLabCut [38] is state of the art for 2D animal pose estimation. In the article the authors evaluate and report numbers in terms of RMSE on their odor trail tracking data where they estimate the pose (snout, left and right ear and tail base) of single mice. That is why we also report RMSE only in this section. The networks are trained a total of 650K iterations with batch size 1 for three splits of 0.8/0.2 (training/test) and evaluated every 50K iterations. The authors also report the average of the three splits. For more details see [38].

Table 1. *Comparison with DeepLabCut (DLC).* RMSE on the odor trail tracking test set from [38]. Values for DLC from [38]. We report precision within ±0.2 because we have to read values from Fig. 2c in [38].

Model, iterations	RMSE [px]
I-MuPPET, 200K iterations	4.2
DLC, 200K iterations	3.6 ± 0.2
DLC, 350K/600K iterations	$\mathbf{3.2} \pm \mathbf{0.2}$

In order to compare I-MuPPET to DeepLabCut, we train their odor trail tracking data set with our framework. In addition we randomly sample 1000 frames from our full single pigeon data set. This sub data set represents our four sessions in the same way as our full single pigeon data set. We train our framework on the DeepLabCut and our sub-sampled single pigeon data with the configuration that we report in Sect. 3.3. We train for 250 epochs with a batch size of 20 instead of 1 to exploit our hardware and fine-tune twice for another 250 epochs with training configurations that lower the learning rate further to compare our results to those of DeepLabCut after 200K, 400K and 600K iterations.

Table 1 compares results for DeepLabCut from [38] with our framework. We obtained the results for DeepLabCut from Fig. 2c in [38]. These results were achieved with a network based on ResNet-50. We report their values for 200K iterations and their absolute lowest RMSE on test set averaged over the three 0.8/0.2 splits. For our framework we report numbers with the same precision as we are able to read for DeepLabCut. We report numbers for 200K iterations only because our network does not improve the accuracy of pose estimation in the test set when trained for more iterations: 4.2 px@200K (cf. Table 1) on the odor trail tracking test set from DeepLabCut averaged over the three splits, 3.2 px@200K on our sub-sampled single pigeon test set averaged over the three splits.

I-MuPPET is comparable with DeepLabCut in terms of RMSE meaning that we also achieve a RMSE of about 4 px on the odor trail tracking test set. In addition, we achieve a RMSE of about 3 px for our sub-sampled single pigeon data set, both after 200K training iterations. Overall, this comparison shows that I-MuPPET achieves performance on par with state-of-the-art.

4.3 Comparison with 3D Bird Reconstruction

3D Bird Reconstruction [3] is state of the art for 3D bird shape recovery, and they also report on accuracy on 2D bird pose estimation. The authors evaluate and report numbers in terms of PCK (cf. Sect. 4.1) on their cowbird data, where they estimate the pose (bill tip, right and left eyes, neck, nape, right and left wrists, right and left wing tips, right and left feet and the tail tip) of single cowbirds. Their network is trained for 60 epochs (private e-mail communication with the authors) with a train/test split of 0.75/0.25. For more details see [3].

Table 2. *Comparison with 3D Bird Reconstruction (3DBR).* PCK on the cowbird test set from [3]. Values for 3DBR from [3].

Model, epochs	@0.05	@0.1
I-MuPPET, 45 epochs	0.39	0.56
I-MuPPET, 60 epochs	0.36	0.54
3DBR, 60 epochs	**0.46**	**0.64**

In order to compare I-MuPPET to 3D Bird Reconstruction, we train their single cowbird data with our framework. In addition we take the same single pigeon sub data set containing 1000 frames (cf. Sect. 4.2) to report PCK on our pigeon data. We remind the reader that PCK properly takes into account scale, and thus this accuracy measure is more meaningful than RMSE (cf. Sect. 4.1) reported in Sect. 4.2. We train our framework (cf. Sect. 3.2) on the cowbird and our sub-sampled pigeon data with the configuration that we report in Sect. 3.3. We train for 60 epochs with a batch size of 20 to compare our results to those of 3D Bird Reconstruction. Our framework achieves best performance on the cowbird data after 45 epochs, which is why we report PCK for these as well.

Sect. 2 compares results for 3D Bird Reconstruction from [3] with our framework. While I-MuPPET achieves lower accuracy by 7% (PCK@0.05) and 8% (PCK@0.1) on the cowbird data set than 3D Bird reconstruction, I-MuPPET converges faster (45 epochs vs. 60 epochs). In addition, we achieve a PCK of 0.94@0.05 and 0.97@0.1 for our sub-sampled single pigeon data set after 60 training epochs.

Table 3. *Combined Quantitative Tracking Evaluation.* We test 24 video sequences quantitatively with the metrics specified in Sect. 4.1. Here we report the combined results for different detection confidence scores of the Keypoint R-CNN (cf. Sect. 3.2). The space is unfortunately not sufficient to explain all abbreviations and metrics in detail, please refer to our supplemental material.

conf. score	HOTA↑	MOTA↑	MOTP↑	Rcll↑	Prcn↑	MT↑	ML↓	FPF↓	IDS↓	Frag↓	IDF1↑
0	0.53	0.48	**0.61**	**0.83**	0.70	**0.64**	**0.01**	0.99	24	292	0.75
0.5	**0.57**	0.65	**0.61**	**0.83**	0.83	**0.64**	**0.01**	0.49	**8**	278	0.82
0.75	**0.57**	0.67	**0.61**	**0.83**	0.84	**0.64**	**0.01**	0.44	11	280	**0.83**
0.9	0.56	**0.68**	**0.61**	0.82	**0.85**	**0.64**	**0.01**	**0.39**	14	**277**	**0.83**

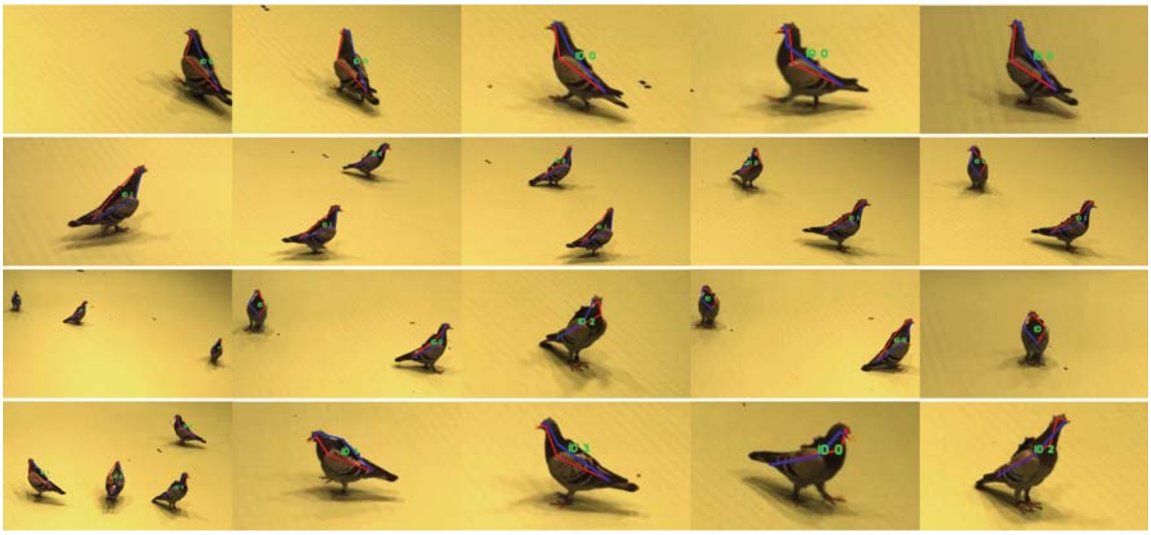

Fig. 4. *Qualitative Results of I-MuPPET.* Cropped sample frames of our pipeline. Left body side of pigeon in red, right one in blue and beak in orange. First and second, third and fourth row are from video sequences with one, two, three and four pigeons present respectively. Sometimes not all pigeons are present in cropped frame for a better view.

4.4 I-MuPPET Tracking Performance

The availability of annotated data from animals is limited, especially for multiple individuals. To overcome this obstacle, we train our Keypoint R-CNN (cf. Sect. 3.2) on our single pigeon data (cf. Sect. 3.1) and infer 2D keypoints on multi-pigeon video sequences. In addition we track the individuals with SORT (cf. Sect. 3.2). We do so for up to four pigeons present in the videos. The video sequence with one pigeon present is not from our labeled single pigeon data set.

Figure 4 shows results of the 2D pose estimation and tracking task for multiple pigeons. The 2D keypoint locations show a very good accuracy even though I-MuPPET was trained on single pigeon data only. The individuals are tracked correctly. See also our supplementary video sequences.

Quantitative Tracking Evaluation. We test I-MuPPET quantitatively on 24 video sequences recorded with 50 fps. They contain between one and four pigeons and 7872 frames and 70 objects in total. For evaluation we use the metrics specified in Sect. 4.1. In Table 3 we report the combined results of the 24 video sequences for different detection confidence scores. We see that tracking does not improve much when setting the detection score from 0.5 to 0.75. We get the best tracking results for a confidence score of 0.9. Detailed results for this detection confidence score of 0.9 are shown in Table 4. We achieve an overall good result with I-MuPPET on the video sequences (HOTA: 0.56, MOTA: 0.68, MOTP: 0.61, Recall: 0.82, Precision: 0.85 and IDF1: 0.83).

By far the worst sequence with respect to tracking accuracy is 4*_pigeons_*8. In this sequence the four pigeons walk towards the edge of the facility that in the camera view appears darker than it does otherwise. In cases of high fragments (frag), e.g. sequences 2*_pigeons_*3 and 4*_pigeons_*9, the video sequences show the same darker regions. Please note that this can be solved by simply setting the SORT [8] parameter to keep alive a track without associated detections to a value higher than 1. We leave it at 1 since we are interested in online tracking and not in re-identification. Thus early deletion of lost targets improves efficiency.

Table 4. *Detailed Quantitative Tracking Evaluation*. We test 24 video sequences quantitatively with the metrics specified in Sect. 4.1. The threshold for the confidence score of the Keypoint R-CNN (cf. Sect. 3.2) is set to 0.9.

Video seq.	HOTA↑	MOTA↑	MOTP↑	Rcll↑	Prcn↑	MT↑	ML↓	FPF↓	IDS↓	Frag↓	IDF1↑
1_pigeon_1	0.64	1	0.65	1	1	1	0	0	0	0	1
1_pigeon_2	0.58	0.89	0.62	0.94	0.94	1	0	0.06	0	2	0.94
1_pigeon_3	0.57	0.96	0.59	0.98	0.98	1	0	0.02	0	7	0.98
2_pigeons_1	0.53	0.57	0.56	0.78	0.78	0.50	0	0.43	0	1	0.78
2_pigeons_2	0.56	0.99	0.57	0.99	0.99	1	0	0.01	0	2	0.99
2_pigeons_3	0.57	0.76	0.58	0.88	0.88	0.50	0	0.24	0	30	0.88
2_pigeons_4	0.60	0.94	0.60	0.97	0.97	1	0	0.06	0	2	0.97
2_pigeons_5	0.65	0.99	0.66	0.99	1	1	0	0	1	1	0.99
2_pigeons_6	0.69	1	0.69	1	1	1	0	0	0	0	1
3_pigeons_1	0.57	0.50	0.60	0.75	0.75	0.67	0	0.74	0	27	0.75
3_pigeons_2	0.57	0.81	0.59	0.90	0.91	0.67	0	0.28	0	11	0.91
3_pigeons_3	0.59	0.91	0.60	0.96	0.96	1	0	0.13	0	7	0.96
3_pigeons_4	0.64	0.73	0.66	0.87	0.87	0.67	0	0.40	0	17	0.87
3_pigeons_5	0.62	0.82	0.64	0.91	0.91	0.67	0	0.27	0	7	0.91
4_pigeons_1	0.47	0.49	0.56	0.73	0.75	0.50	0	0.95	2	23	0.72
4_pigeons_2	0.46	0.32	0.55	0.60	0.68	0.25	0	1.14	2	19	0.64
4_pigeons_3	0.48	0.75	0.57	0.84	0.90	0.75	0	0.36	3	16	0.82
4_pigeons_4	0.59	0.62	0.65	0.80	0.82	0.75	0	0.73	1	6	0.80
4_pigeons_5	0.63	1	0.64	1	1	1	0	0	0	0	1
4_pigeons_6	0.54	0.77	0.60	0.85	0.91	0.50	0	0.33	0	5	0.88
4_pigeons_7	0.55	0.53	0.63	0.76	0.76	0.50	0	0.93	1	12	0.76
4_pigeons_8	0.29	−0.09	0.55	0.26	0.42	0	0	1.38	1	25	0.29
4_pigeons_9	0.50	0.47	0.59	0.73	0.74	0.25	0	1	1	52	0.71
4_pigeons_10	0.46	0.44	0.58	0.69	0.74	0.75	0.25	0.97	2	5	0.63
Combined	0.56	0.68	0.61	0.82	0.85	0.64	0.01	0.39	14	277	0.83

Inference Speed. We also benchmark the inference speed of I-MuPPET (cf. Sect. 3.2) with the four videos from our supplemental material. The benchmark includes the complete pipeline except for loading the model. It includes also I/O times reading the images from our AVI video sequences (encoded with libx264). We loop three times over the full video sequence, repeat this procedure three times and calculate the average. We obtain an interactive speed of about $12 - 13$ fps (cf. Table 5) for our full pipeline. Interestingly, speed is almost independent from the number of pigeons present in the video.

Table 5. *I-MuPPET Inference Speed.* Benchmark for our complete pipeline. We process our pipeline frame by frame which also includes I/O times reading the images from our AVI video sequences. Values for different number of pigeons differ by 1 fps at most.

	1 pigeon	2 pigeons	3 pigeons	4 pigeons
Frame rate [fps]	13.1	13.0	12.5	12.1

We also benchmark the scenario where we preload the video sequence in memory and are thus independent of disk I/O, with otherwise the same procedure, see Table 6 for results. We report values for batch sizes up to 32, after which we do not observe any speed-up. The speed of our pipeline increases for a batch size of 1 by about 1 fps (comparing Table 5 with Table 6) if we preload the video to memory. The maximum speed is at batch size 16 and 32 with an interactive speed of about $16 - 17$ fps depending on the number of pigeons present in the video sequence.

There are two frameworks which also perform 2D keypoint prediction of complex poses and tracking: maDLC [34] and SLEAP [47]. maDLC [34] does not report numbers on inference speed. SLEAP [47] instead reports numbers and also compares to a SLEAP version of a DLC ResNet model for multi-instance pose estimation. Their benchmark procedure and hardware is comparable to

Table 6. *I-MuPPET Inference Speed.* Benchmark for our in-memory pipeline. We benchmark our pipeline with our AVI video sequences preloaded in memory and report values for different batch sizes.

Batch size	frame rate [fps]			
	1 pigeon	2 pigeons	3 pigeons	4 pigeons
1	14.5	14.1	13.8	13.5
2	15.2	14.4	14.6	14.4
4	15.6	15.3	14.9	14.8
8	16.1	15.5	15.5	15.0
16	17.1	16.8	**16.3**	**16.1**
32	**17.4**	**16.9**	16.2	15.9

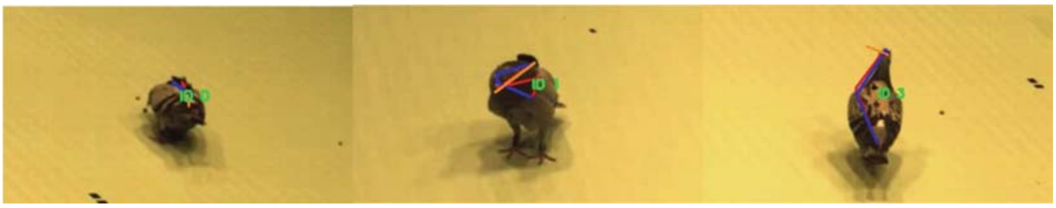

Fig. 5. *Limitations.* Cropped frames of failure cases. See Fig. 1 for an explanation of colors and labels.

ours. For details we refer to [47]. A rough comparison yields that I-MuPPET is comparable in inference speed with the DLC ResNet version of SLEAP. SLEAP [47] instead is about an order of magnitude faster than our framework (numbers read off from [47], Figs. 2b, 3e and Extended Data Fig. 6c; considering the fact that the pigeon image resolution is higher than the one of the flies and mice (open field) and thus we process more data through the whole pipeline). While I-MuPPET solves the substantially harder task of a 'generalist' approach of training a single model that works on all datasets, SLEAP uses a 'specialist' paradigm where small, lightweight models have just enough representational capacity to generalize to the low variability typically found in scientific data [47]. The approach of I-MuPPET comes with an additional cost of compute resource requirements. Albeit with I-MuPPET we want to offer a framework that works with both low and high variability data at the same time, depending on the application, one can easily change the pose estimator of I-MuPPET to achieve frame rates comparable to SLEAP.

4.5 Limitations and Future Work

From Fig. 5 we see that in some frames of the multi-pigeon video sequences, pose estimation is not accurate. In addition the bounding box detector fails in cases where pigeons are too close together, or occlude each other, since we trained it only on single pigeon data. This also affects the pose estimation in this case. Both of these situations can probably be improved by exploiting labeled multi-instance data. Since availability is limited one approach is to synthetically exploit the single instance data. There currently are no instances of occlusions in our multi-pigeon video sequences, we intend to create more varied datasets to assess performance in more complex scenarios.

5 Conclusion

In this work we present I-MuPPET, an interactive multi-pigeon pose estimation and tracking system. While training a neural network only on single pigeon training data, we demonstrate that we can still predict keypoints of a complex pose (seven distinct keypoints) for multiple pigeons and track the individuals at interactive speed of 12.1 – 17.4 fps. I-MuPPET has also a comparable accuracy with DeepLabCut [38] in terms of RMSE with respect to the estimation of

2D animal keypoints. Furthermore we perform a quantitative tracking evaluation on 24 video sequences and obtain good results (HOTA: 0.56, MOTA: 0.68, MOTP: 0.61, Recall: 0.82, Precision: 0.85 and IDF1: 0.83). We hope that this work inspires researchers to improve upon our baseline and pose estimation and tracking of multiple animals in general. As discussed above, we have strived to give a fair comparison, but due to the limitations in the reported data and the different domains of the methods, this comparison can not be fully rigorous. However, it gives, in our opinion, sufficient information to indicate the competitive performance of the proposed framework. Nevertheless, our future work will additionally focus on more datasets for a more comprehensive quantitative performance comparison of animal pose estimation and tracking across different species, which we believe is necessary to make further systematic progress.

References

1. Altmann, J.: Observational study of behavior: sampling methods. Behaviour **49**(3–4), 227–266 (1974)
2. Anderson, D., Perona, P.: Toward a science of computational ethology. Neuron **84**(1), 18–31 (2014)
3. Badger, M., et al.: 3d bird reconstruction: a dataset, model, and shape recovery from a single view. In: ECCV, pp. 1–17 (2020)
4. Bala, P.C., Eisenreich, B.R., Yoo, S.B.M., Hayden, B.Y., Park, H.S., Zimmermann, J.: Automated markerless pose estimation in freely moving macaques with openMonkeyStudio. Nat. Commun. **11**, 4560 (2020)
5. Berman, G.J.: Measuring behavior across scales. BMC Biol. **16**(23), 1–11 (2018)
6. Bernardin, K., Stiefelhagen, R.: Evaluating multiple object tracking performance: the clear mot metrics. EURASIP J. Image Video Process. **2008**, 1–10 (2008)
7. Bernshtein, N.: The Co-ordination and Regulation of Movements. Pergamon Press (1967)
8. Bewley, A., Ge, Z., Ott, L., Ramos, F., Upcroft, B.: Simple online and realtime tracking. In: ICIP, pp. 3464–3468 (2016)
9. Biggs, B., Roddick, T., Fitzgibbon, A., Cipolla, R.: Creatures great and SMAL: recovering the shape and motion of animals from video. In: Jawahar, C.V., Li, H., Mori, G., Schindler, K. (eds.) ACCV 2018. LNCS, vol. 11365, pp. 3–19. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-20873-8_1
10. Bolaños, L.A., et al.: A three-dimensional virtual mouse generates synthetic training data for behavioral analysis. Nat. Methods **18**, 378–381 (2021)
11. Chen, X., Zhai, H., Liu, D., Li, W., Ding, C., Xie, Q., Han, H.: SiamBOMB: a real-time AI-based system for home-cage animal tracking, segmentation and behavioral analysis. In: IJCAI, pp. 5300–5302 (2020)
12. Dell, A.I., et al.: Automated image-based tracking and its application in ecology. Trends Ecol. Evol. **29**(7), 417–428 (2014)
13. Dendorfer, P., et al.: MOTChallenge: a benchmark for single-camera multiple target tracking. Int. J. Comput. Vis. **129**(4), 845–881 (2020). https://doi.org/10.1007/s11263-020-01393-0
14. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: CVPR, pp. 248–255 (2009)
15. Dunn, T.W., et al.: Geometric deep learning enables 3D kinematic profiling across species and environments. Nat. Methods **18**(5), 564–573 (2021)

16. Duporge, I., Isupova, O., Reece, S., Macdonald, D.W., Wang, T.: Using very-high-resolution satellite imagery and deep learning to detect and count African elephants in heterogeneous landscapes. Remote Sens. Ecol. Conserv. **7**(3), 369–381 (2021)
17. Ferrero, F.R., Bergomi, M.G., Heras, F.J., Hinz, R., de Polavieja, G.G.: The champalimaud foundation: idtracker.ai (2017). https://idtrackerai.readthedocs.io/en/latest
18. Gomez-Marin, A., Paton, J.J., Kampff, A.R., Costa, R.M., Mainen, Z.F.: Big behavioral data: psychology, ethology and the foundations of neuroscience. Nat. Neurosci. **17**, 1455–1462 (2014)
19. Gosztolai, A., et al.: Liftpose3D, a deep learning-based approach for transforming two-dimensional to three-dimensional poses in laboratory animals. Nat. Methods **18**, 975–981 (2021)
20. Graving, J.M., et al.: Deepposekit, a software toolkit for fast and robust animal pose estimation using deep learning. eLife **8**, e47994 (2019)
21. Günel, S., Rhodin, H., Morales, D., Campagnolo, J., Ramdya, P., Fua, P.: Deepfly3D, a deep learning-based approach for 3D limb and appendage tracking in tethered, adult Drosophila. eLife **8**, e48571 (2019)
22. He, K., Gkioxari, G., Dollar, P., Girshick, R.: Mask R-CNN. In: ICCV (2017)
23. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
24. Heras, F.J.H., Romero-Ferrero, F., Hinz, R.C., de Polavieja, G.G.: Deep attention networks reveal the rules of collective motion in zebrafish. PLOS Comput. Biol. **15**(9), 1–23 (2019)
25. Ionescu, C., Papava, D., Olaru, V., Sminchisescu, C.: Human3.6M: large scale datasets and predictive methods for 3D human sensing in natural environments. IEEE Trans. Pattern Anal. Mach. Intell. **36**(7), 1325–1339 (2014)
26. Iskakov, K., Burkov, E., Lempitsky, V., Malkov, Y.: Learnable triangulation of human pose. In: ICCV (2019)
27. Jonathon Luiten, A.H.: Trackeval. https://github.com/JonathonLuiten/TrackEval (2020)
28. Joska, D., et al.: AcinoSet: a 3D pose estimation dataset and baseline models for cheetahs in the wild. In: 2021 IEEE International Conference on Robotics and Automation (ICRA), pp. 13901–13908 (2021). https://doi.org/10.1109/ICRA48506.2021.9561338
29. Kalman, R.E.: A new approach to linear filtering and prediction problems. J. Basic Eng. **82**(1), 35–45 (1960)
30. Karashchuk, P., et al.: Anipose: a toolkit for robust markerless 3D pose estimation. Cell Rep. **36**(13), 109730 (2021)
31. Kays, R., Crofoot, M.C., Jetz, W., Wikelski, M.: Terrestrial animal tracking as an eye on life and planet. Science **348**(6240), aaa2478 (2015)
32. Kuhn, H.W.: The Hungarian method for the assignment problem. Naval Res. Logist. Q. **2**(1–2), 83–97 (1955)
33. Labuguen, R., et al.: MacaquePose: a novel "in the wild" macaque monkey pose dataset for markerless motion capture. Front. Behav. Neurosci. **14**, 268 (2021)
34. Lauer, J., et al.: Multi-animal pose estimation, identification and tracking with DeepLabCut. Nat. Methods **19**, 496–504 (2022)
35. Li, Y., Huang, C., Nevatia, R.: Learning to associate: HybridBoosted multi-target tracker for crowded scene. In: CVPR, pp. 2953–2960 (2009)
36. Lin, T.Y., Dollar, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: CVPR (2017)

37. Dendorfer, P., et al.: HOTA: a higher order metric for evaluating multi-object tracking. Int. J. Comput. Vis. **129**(2), 548–578 (2021). https://doi.org/10.1007/s11263-020-01375-2
38. Mathis, A., et al.: DeepLabCut: markerless pose estimation of user-defined body parts with deep learning. Nat. Neurosci. **21**, 1281–1289 (2018)
39. Naik, H.: XR For all: Closed-loop Visual Stimulation Techniques for Human and Non-Human Animals. Dissertation, Technische Universität München, München (2021)
40. Nath, T., Mathis, A., Chen, A.C., Patel, A., Bethge, M., Mathis, M.W.: Using DeepLabCut for 3D markerless pose estimation across species and behaviors. Nat. Protoc. **14**, 2152–2176 (2019)
41. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. In: ECCV, pp. 483–499 (2016)
42. Nourizonoz, A., et al.: EthoLoop: automated closed-loop neuroethology in naturalistic environments. Nat. Methods **17**, 1052–1059 (2020)
43. Park, H.S., Rhodin, H., Kanazawa, A., Neverova, N., Nobuhara, S., Black, M.: Cv4Animals: computer vision for animal behavior tracking and modeling (2021). https://www.cv4animals.com/
44. Paszke, A., et al.: PyTorch: an imperative style, high-performance deep learning library. In: NeurIPS (2019)
45. Pedersen, M., Haurum, J.B., Bengtson, S.H., Moeslund, T.B.: 3D-ZeF: a 3D zebrafish tracking benchmark dataset. In: CVPR (2020)
46. Pereira, T.D., et al.: Fast animal pose estimation using deep neural networks. Nat. Methods **16**, 117–125 (2019)
47. Pereira, T.D., et al.: SLEAP: a deep learning system for multi-animal pose tracking. Nat. Methods **19**, 486–495 (2022)
48. Ristani, E., Solera, F., Zou, R., Cucchiara, R., Tomasi, C.: Performance measures and a data set for multi-target, multi-camera tracking. In: ECCV, pp. 17–35 (2016)
49. Romero-Ferrero, F., Bergomi, M.G., Hinz, R.C., Heras, F.J.H., de Polavieja, G.G.: idtracker.ai: tracking all individuals in small or large collectives of unmarked animals. Nat. Methods **16**, 179–182 (2019)
50. Van Horn, G., et al.: Building a bird recognition app and large scale dataset with citizen scientists: the fine print in fine-grained dataset collection. In: CVPR (2015)
51. Walter, T., Couzin, I.D.: Trex, a fast multi-animal tracking system with markerless identification, and 2D estimation of posture and visual fields. eLife **10**, e64000 (2021)
52. Wang, J., Yuille, A.L.: Semantic part segmentation using compositional model combining shape and appearance. In: CVPR (2015)
53. Wang, P., Shen, X., Lin, Z., Cohen, S., Price, B., Yuille, A.L.: Joint object and part segmentation using deep learned potentials. In: ICCV (2015)
54. Welinder, P., et al.: Caltech-UCSD Birds 200. Tech. Rep. CNS-TR-2010-001, California Institute of Technology (2010)
55. Xiao, B., Wu, H., Wei, Y.: Simple baselines for human pose estimation and tracking. In: ECCV (2018)
56. Yang, Y., Ramanan, D.: Articulated human detection with flexible mixtures of parts. IEEE Trans. Pattern Anal. Mech. Intell. **35**(12), 2878–2890 (2013)