PigPose: A Realtime Framework for Farm Animal Pose Estimation and Tracking

Milan Kresovic, Thong Nguyen, Mohib Ullah, Hina Afridi, Faouzi Alaya Cheikh

PigPose: A Realtime Framework for Farm Animal Pose Estimation and Tracking

Milan Kresovic[1], Thong Nguyen[1], Mohib Ullah[1], Hina Afridi[1,2], and Faouzi Alaya Cheikh[1]

[1] Norwegian University of Science and Technology, 2815 Gjøvik, Norway.
[2] Geno SA Hamar, Norway.

Abstract. In industrial farming, livestock well-being is becoming increasingly more important. Animal breeding companies are interested in enhancing the total merit index used in breeding programs. Pigs tracking and behaviour analysis plays a crucial role in breeding programs. To this end, we proposed a tracking-by-detection approach for detecting and tracking indoor farm animals for an extended period. We exploited a modified OpenPose model for the detection where the features from the input frames are extracted through EfficientNet, and the detected Keypoints are associated through a greedy optimization mechanism. Additionally, the attention mechanism is incorporated in the pose estimation framework to refine the input frames' features maps. A bipartite graph is created for every two frames to track the animals over an extended period. The edge cost is defined by the spatial distance between the detected Keypoints of the animals in the temporal domain. We collected and annotated the customized dataset from the pig farm to train the model. The dataset and annotation will be made publicly available to help promote research in the farming industry. The proposed method is evaluated on AP^{OKS} and AR^{OKS}, and promising results are achieved.

Keywords: Attention mechanism · pose estimation · tracking · greedy optimization · bipartite graph.

1 Introduction

The welfare of livestock is becoming increasingly more important in industrial farming. Besides altruistic and humane reasons, good animal welfare also contributes to the better food quality of animal products. Even though many regulations have been introduced to manage industrial farming, the current industrial practices do not address sustainability issues. To improve farm products and comply with the animal welfare regulations, breeding companies can leverage vision-based solutions to monitor the animal living and conceive novel animal traits that can enhance the breeding programs. Compared to manual monitoring of animals, computer vision provides a non-invasive solution. Especially, tracking multiple pigs in a pen for an extended period of time provide invaluable information about animal behavior that could be used in the breed programs to

enhance the total merit indexes (TMI) [1]. Technically, multi-target tracking is considered a challenging problem and has been studied extensively over the past few decades. Tracking multiple targets in a scene has a broad range of applications in different fields such as surveillance of common places [2], crowd flow management [3], home robotics [4], and tracking in MRI-guided radiotherapy for different disease diagnosis [5]. However, in the farming industry, such approaches are not prevalent.

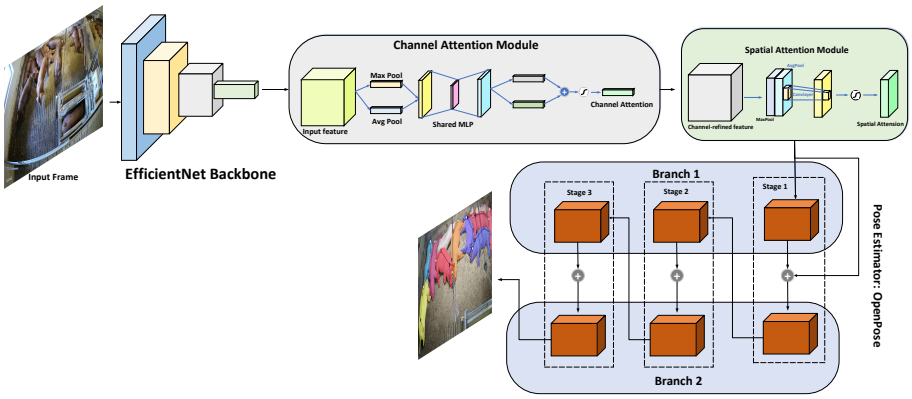


Fig. 1: First, EfficientNet extracts the image features that generate feature maps. The feature maps are refined through channel and spatial attention module. The refined feature maps are given as input to the first stage of Pose estimator. Here the network predicts the PAFs, which presents a degree of association between different animal parts. Later, in stage 2, the network predicts the confidence maps of the animal organs like the nose, left ear, right ear, neck, back, and tail. Finally, the network associate the corresponding organic key points through a greedy optimization.

Thanks to the rapid advancement in video analysis, research on vision-based pig industrial farming has shown a steady increase over the few years. Jaewon et al. [6] proposed an approach for reliable pig detection under various illumination conditions. In addition to RGB, they use additional information like depth maps and infrared images to make detections as trust-worthy as possible. Using non-visual information helps in optimizing the execution and allows for bypassing computationally heavy deep learning models. On the other hand, to obtain this type of information additional hardware is needed to be installed in the pens where the pigs are habituating, which increases the cost of the overall system. Liu et al. [7] used two linear SVM to obtain the candidate regions of pigs on the input image. These candidate regions are then forwarded to the CNN which uses them to classify and identify true pig detection. Furthermore, Ju et al. [8] proposed a hybrid system for segmenting touching-pigs. First, the input data is

obtained using the Kinect depth sensor. Afterward, YOLO [9] is used for pig detection. If the boundary generated by this network is not satisfactory, a more heuristic approach is used where a possible boundary is suggested by analyzing the shape of the pigs. An approach that is addressing the similar problem of pig segmentations is presented by Brunger et al. [10] where instead of using bounding boxes or keypoints location estimation, they focused more on tracing the contours of the pig and obtain the panoptic segmentation. A pipeline for pig detection and tracking in pig farms has been presented in [11]. Intrinsically, the pipeline uses a CNN based detector and Bayesian filters for tracking. Additionally, after a pig has been detected, the tracking algorithm uses features extracted from detected tag-boxes from the previous step. In a similar line of work, Cowton et al. [12] adapted Faster Region-based CNN for pigs with two real-time multi-object tracking algorithms. The caveats of both these approaches are something that is called frame loss which is a tendency of the tracking algorithm to lose some of the tracking frames because usually, the detection part faced some occlusions of the tracking instances. To address this problem, Sun et al. [13] used Faster-Region-CNN to obtain bounding boxes of pigs, where these bounding boxes are forwarded to the SURF algorithm. Together with the background difference method, this algorithm attempts to determine whether the pig will be partially obscured in the next frame and avoid tracking frame loss in that way. As the striving goal of all these methods is to be used for the behavioral analysis of pigs. Methods like [14] try to detect specific behavioral patterns from the input data. Specifically, Li et al. [14] proposed a pipeline for detecting mounting behavior. They used Mask R-CNN [15] is used for detecting pigs in frames and the detected regions of interest are then used to calculate eigenvectors, which are in the end classified with kernel extreme learning machine (KELM) to see if the mounting behavior has happened. The most common tracking paradigm

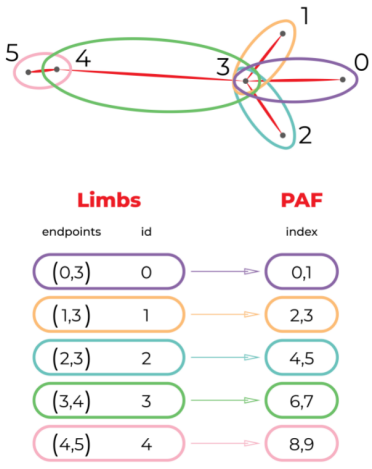


Fig. 2: Description of pig limbs and the corresponding index of PAF.

is known as tracking-by-detection[16][17][18][19]. In a nutshell, the problem is divided into two parts. The first part is obtaining the location of the instances that need to be tracked. While the second part is essentially based on optimization and it aims to associate the corresponding instances for obtaining smooth trajectories in the temporal domain. Historically, the localization of instances in every frame is achieved through a generative [20] or a discriminative [21] model. Different geometric shapes like rectangle, ellipse, circle are used to encapsulate object instances. More recently, methods that localize Keypoints of object instance are getting more popular as it gives the flexibility of detection even in partial occlusion. The detected Keypoints can be reassembled to get a unified skeleton of the target and helped tracked the target for an extended period. This paper focuses on conceiving a multi-target tracking framework for indoor pig tracking. In a nutshell, the contribution of the work is three-fold:

- We proposed an attention based pose estimation framework for pigs. We explored the state-of-the-art CNN model for extracting the features from the input frames.
- We collected the data at pig farm, and did the annotation to training and validating our proposed model.
- We tested the proposed pose estimation framework in tracking multiple pigs and achieved promising results.

The paper is organized as follows. Section 2 elaborate the proposed method. The training strategy and loss function is elaborated in section 3. Tracking module is explained in 4 while section 5 gives the details of the dataset, annotation, implementation, and limitation of the work. Section 6 concludes the paper and gives the future direction.

2 Proposed Method

The pig Keypoint detection pipeline is demonstrated in Figure 1. At input, it takes the RGB image and extracts the deep features through EfficientNet [22]. By exploiting EfficientNet, the spatial resolution is saved by reusing the backbone weights through dilating convolution, which essentially removes the layer $conv4_2/dw$ of the original model [23]. The extracted feature maps are refined through channel and spatial attention module. We used convolution block attention module (CBAM) [24] for refining the features maps. In the next step, the extracted deep features are given to a multi-stage CNN pipeline that extracts part confidence maps and part affinity field. The confidence map is essentially a 2D representation of the probability that a particular pig body part can be found in the given location. Mathematically, it is represented as:

$$S = (S_1, S_2, ..., S_J) \qquad S_J \in R^{w*h} \tag{1}$$

where S represents the confidence map and J the number of the pig body part that is assumed to be six in this work. Similarly, the part affinity fields are is the

set of 2D vectors that represent the location and orientation of limbs of the pigs in the image. It essentially represents the form of pairwise connections between the body parts. Graphically, it is represented in Figure 2. Mathematically, it can be written as:

$$L = (L_1, L_2, ..., L_C) \qquad L_c \in R^{w*h*c} \tag{2}$$

Where L L is the part affinity field and C represents the limb index that is five in the case of pigs. After extracted two pieces of information from the deep features, the parsing between confidence maps and PAFs is conducted to assemble all 2D Keypoints locations into individual body poses for every pig in the input. It is essentially an optimization process and solved through the Hungarian algorithm.

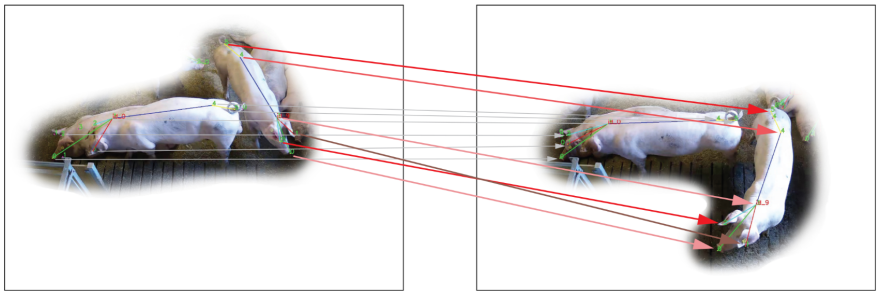


Fig. 3: Temporal association of the detecting keypoints in frame f_{t-1} and f_t.

3 Loss functions

At each time step, after calculating the confidence map and part affinity field, an L2-loss is computed between the predicted confidence maps and Part Affinity fields to the ground truth maps and fields. Mathematically, the loss function component related to the confidence map is computed as:

$$f_S^{t_k} = \sum_{j=1}^{J} \sum_{p} W(p).||S_j^{t_k}(p) - S_j^*(p)||_2^2 \tag{3}$$

where S_j^* is the groundtruth confidence map and $S_j^{t_k}$ is the predicted confidence map by the model. Similarly, the loss function component related to affinity field is computed as:

$$f_L^{t_i} = \sum_{c=1}^{C} \sum_{p} W(p).||L_c^{t_i}(p) - L_c^*(p)||_2^2 \tag{4}$$

Where L_c^* is the groundtruth and $L_c^{t_i}$ is the predicted affinity field, respectively. In addition, we have the parameter W that is used to commentate the loss

when the Keypoint is not visible in the training sample. By combining the two components, we can write the overall loss of the network as:

$$f = \sum_{t=1}^{TP} f_L^t + \sum_{t=Tp+1}^{T_P+T_C} f_S^t \tag{5}$$

where f_S represented the loss related to the confidence map and f_L is the loss related to the part affinity. To improve the model generalization, data augmentations like rotation, scaling, cropping, and Gaussian noise has been applied to the training set.

4 Tracking

After detecting the Keypoints of individual pigs, we use our tracking approach to track the pigs for extended period of time. We followed the tracking-by-detection paradigm [25] and used spatial distance to associate the corresponding pigs. The graphical depiction of pigs association is presented in Fig. 3. The edge cost $c_{i,j}$ between the Keypoints of two pigs is calculated through a statistical similarity metric [26] as follows:

$$c_{i,j} = \frac{1}{2}(D_{KL}(\varphi_i||M) + D_{KL}(\varphi_j||M)) \tag{6}$$

φ_i and φ_j are the spatial position of pig i and j, respectively. M is the average distance of the two pigs while D_{KL} is the Kullback-Leibler divergence. $c_{i,j}$ is the edge cost that shows the affinity between two pigs in the consecutive frames.

Fig. 4: (a) Correction detection and tracking, (b) Scenario with Occlusion and overlapping, (c) Wrong detection and tracking, limb liked to a wall, (d) Wrong PAFs among pigs

5 Experiments

In this section, we presented a brief overview of the implementation, parameter selection, the dataset and annotation, and the limitation of the proposed framework.

5.1 Dataset and annotation

The main contributor of the data is Norsvin that is the largest pig breeding company in Norway. They provided 26GB of different video recordings of pigs in the pens. The camera position is fixed and the lighting is mostly uniform. The frame size is 2688x1520 with 15fps. For the purpose of the videos, two different types of pigs were recorded. The dataset was developed in the standard MS-COCO format. We annotated each pig in every frame for six keypoints i.e. nose, left and right ear, neck, back, and tail. In addition to Keypoints, a binary mask of each pig is generated which assists in model training. An annotated sample can be seen in 5.

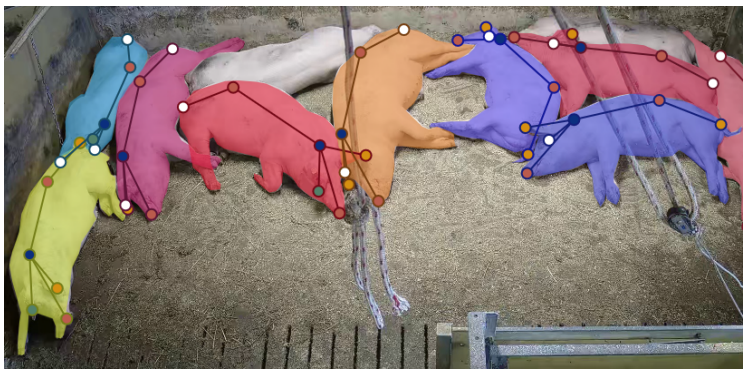
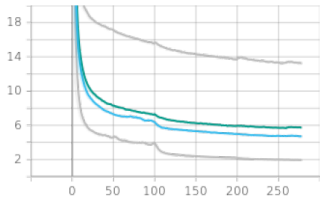


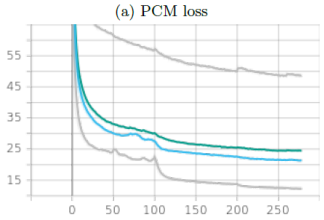
Fig. 5: Annotated sample frame

5.2 Implementation

Models were trained with 4 GPUs NVIDIA Tesla V100 16Gb, with 870 training and 217 validation annotated images. The training PCM and PAF loss for the last training test for different tested models can be seen in Figure 6. Models were trained in three steps. First, we train the model using the pretrained backbone weights - initialization. The second step was to train the one-stage model with the re-learned backbone weights from the previous step - finetuning the one-stage network. In the last step, we increased the number of stages to 3 and train the model using re-learned weights from the second step - training the three-stage network. The same process was done for all tested networks. On average, each frame is annotated in 12 minutes 5. The details of training parameters are listed in table 1. The empirical parameters PAF threshold (σ) and success ration (τ) are similar to [23].



(a) PCM loss



(b) PAF loss

Fig. 6: Training curves of different backbone architectures: Top gray - MobileNet V1, green -EfficientNet-B3-a, blue - EfficientNeet-B3, bottom gray - EfficientNet-B6

Learning rate	4e-5
Number of Epochs	279
Training time	126 mints
Batch size	32
Inference time	0.8 sec
PAF threshold σ	0.05 sec
Success ratio τ	0.8 sec

Table 1: Empirical parameters, training and inference time

5.3 Results of Pose estimation

For pose estimation, to calculate the average precision score, instead of using intersection over union, a new similarity called object keypoint similarity (OKS) is created. OKS between the predicted pig pose $\hat{\theta^p}$ and the ground truth annotation θ^p of the pig p is calculated as follows: Simply put, OKS between predicted

$$ks(\hat{\theta}_i^{(p)}, \theta_i^{(p)}) = e^{-\frac{\left\|\hat{\theta}_i^{(p)} - \theta_i^{(p)}\right\|_2^2}{2s^2k_i^2}}$$

$$OKS(\hat{\theta}^{(p)}, \theta^{(p)}) = \frac{\sum_i ks(\hat{\theta}_i^{(p)}, \theta_i^{(p)})\delta(W_i > 0)}{\sum_i \delta(W_i > 0)}$$

and true pig pose represents averaged keypoint similarity ks measure over all visible keypoints ($W_i > 0$) in a predicted pose with the corresponding true pose keypoints. To calculate ks similarity for predicted keypoint position, we are using an unnormalized Gaussian function that has a center in true keypoint location. The spread of this Gaussian is determined by a standard deviation k_i specific to a keypoint type. Scale s is related to a pig pose bounding box area so that k_s is size-sensitive. It's important to note that k_i is experimentally determined. To do so, multiple observers are needed to annotate the same set of images, and for each keypoint type i a standard deviation σ_i^2 is calculated. k_i is then calculated as:

$$k_i = 2\sigma_i \tag{7}$$

As we couldn't redundantly annotate single images multiple times, we have chosen a constant k_i value of 0.107 for all keypoints.

	$\mathbf{AP^{OKS}}$ **(m.p.p. = 20)**			$\mathbf{AR^{OKS}}$ **(m.p.p. = 20)**		
OKS	0.5 : 0.05 : 0.95	0.5	0.75	0.5 : 0.05 : 0.95	0.5	0.75
MobileNet V1	0.374	0.675	0.375	0.43	0.704	0.445
EfficientNet-B3a	0.395	0.715	0.393	0.465	0.751	0.476
EfficientNet-B3	0.41	0.723	0.418	0.484	0.770	0.505
EfficientNet-B6	**0.448**	**0.768**	**0.456**	**0.519**	**0.796**	**0.546**

Fig. 7: AP^{OKS} and AR^{OKS} values for different models and different OKS thresholds.

5.4 Results and Discussion

From the values of AP^{OKS} and AR^{OKS}, we can see that the EfficientNet-B6 is performing the best. This is not surprising if we compare the number of parameters for each model. EfficientNet-B6 as such has almost 4 times more parameters than the next best models, EfficientNet-B3 and EfficientNet-B3a, thus increasing the training and inference time. On the other hand, if we compare EfficientNet-B3 with EfficientNet-B3a (a version with the spatial attention proposed by us), we can see that the EfficientNet-B3 is performing a little bit better. Therefore, we can conclude that our modification has not produced any improvements. This could be explained by the fact that the weights for the spatial attention layer have not been initialized by the training on the bigger datasets as all of the EfficientNet original models have been. Overall, we can see that by increasing the number of parameters of the backbone model and by using different architecture we obtained much better results than the framework with MobileNet V1 backbone. If we have to optimize for the model size and its performance, we could choose the EfficientNet-B3 model as it best balances gains and losses. We believed that a very small training set was used by extending the training set with more scenarios, the performance could be improved. Additionally, an ablation study and comparison with the state-of-the-art will highlight the strengthens of the proposed method and will be included in future work.

6 Conclusion

In this paper, we have proposed an improved and optimized Pig Pose framework and tested different backbone CNN models. We have found that by using EfficientNet architecture instead of traditional CNN models, we can acquire significant improvement in the performance. We also showed that adding the attention mechanism didn't provide better results. Additionally, by using a bigger and more diverse dataset, we can managed to create a more robust model. Furthermore, we conducted a more thorough evaluation in terms of AP^{OKS} and AR^{OKS} scores for different models. Future work should include the evaluation of more diverse feature-extractor networks. The pig pose estimation doesn't have to be in realtime, thus, by using bigger models we could improve the prediction. We tested a basic tracker on the proposed model and found a robust pose estimator can yield good tracking results.

References

1. Ramona Weishaar, Robin Wellmann, Amelia Camarinha-Silva, Markus Rodehutscord, and Jörn Bennewitz. Selecting the hologenome to breed for an improved feed efficiency in pigs?a novel selection index. *Journal of Animal Breeding and Genetics*, 137(1):14–22, 2020.
2. Michael Beard, Ba Tuong Vo, and Ba-Ngu Vo. A solution for large-scale multi-object tracking. *IEEE Transactions on Signal Processing*, 2020.

3. Mohib Ullah, Habib Ullah, Nicola Conci, and Francesco GB De Natale. Crowd behavior identification. In *IEEE international conference on image processing*, pages 1195–1199, 2016.
4. Berat A Erol, Abhijit Majumdar, Jonathan Lwowski, Patrick Benavidez, Paul Rad, and Mo Jamshidi. Improved deep neural network object tracking system for applications in home robotics. In *Computational Intelligence for Pattern Recognition*, pages 369–395. Springer, 2018.
5. Jennifer Dhont, Jef Vandemeulebroucke, Davide Cusumano, Luca Boldrini, Francesco Cellini, Vincenzo Valentini, and Dirk Verellen. Multi-object tracking in mri-guided radiotherapy using the tracking-learning-detection framework. *Radiotherapy and Oncology*, 138:25–29, 2019.
6. Jaewon Sa, Younchang Choi, Hanhaesol Lee, Yongwha Chung, Daihee Park, and Jinho Cho. Fast pig detection with a top-view camera under various illumination conditions. *Symmetry*, 11(2):266, 2019.
7. Yan LIU, Longqing SUN, Bing LUO, Shuaihua CHEN, and Yue LI. Multi-target pigs detection algorithm based on improved cnn. *Transactions of the Chinese Society for Agricultural Machinery*, page S1, 2019.
8. Miso Ju, Younchang Choi, Jihyun Seo, Jaewon Sa, Sungju Lee, Yongwha Chung, and Daihee Park. A kinect-based segmentation of touching-pigs for real-time monitoring. *Sensors*, 18(6):1746, 2018.
9. Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.
10. Johannes Brünger, Maria Gentz, Imke Traulsen, and Reinhard Koch. Panoptic segmentation of individual pigs for posture recognition. *Sensors*, 20(13):3710, 2020.
11. Lei Zhang, Helen Gray, Xujiong Ye, Lisa Collins, and Nigel Allinson. Automatic Individual Pig Detection and Tracking in Pig Farms. *Sensors*, 19(5):1188, mar 2019.
12. Jake Cowton, Ilias Kyriazakis, and Jaume Bacardit. Automated individual pig localisation, tracking and behaviour metric extraction using deep learning. *IEEE Access*, 7:108049–108060, 2019.
13. Longqing Sun, Yuanbing Zou, Yan Li, Zhengda Cai, Yue Li, Bing Luo, Yan Liu, and Yiyang Li. Multi target pigs tracking loss correction algorithm based on faster r-cnn. *International Journal of Agricultural and Biological Engineering*, 11(5):192–197, 2018.
14. Dan Li, Yifei Chen, Kaifeng Zhang, and Zhenbo Li. Mounting Behaviour Recognition for Pigs Based on Deep Learning. *Sensors*, 19(22):4924, nov 2019.
15. Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.
16. Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. *arXiv preprint arXiv:2101.02702*, 2021.
17. Mohib Ullah, Faouzi Alaya Cheikh, and Ali Shariq Imran. Hog based real-time multi-target tracking in bayesian framework. In *IEEE international conference on advanced video and signal based surveillance*, pages 416–422, 2016.
18. Wei-Chih Hung, Henrik Kretzschmar, Tsung-Yi Lin, Yuning Chai, Ruichi Yu, Ming-Hsuan Yang, and Drago Anguelov. Soda: Multi-object tracking with soft data association. *arXiv preprint arXiv:2008.07725*, 2020.

19. Mohib Ullah, Ahmed Kedir Mohammed, Faouzi Alaya Cheikh, and Zhaohui Wang. A hierarchical feature model for multi-target tracking. In *IEEE international conference on image processing*, pages 2612–2616, 2017.
20. Yancheng Bai, Yongqiang Zhang, Mingli Ding, and Bernard Ghanem. Sod-mtgan: Small object detection via multi-task generative adversarial network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 206–221, 2018.
21. Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10781–10790, 2020.
22. Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.
23. Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *IEEE transactions on pattern analysis and machine intelligence*, 43(1):172–186, 2019.
24. Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.
25. Mohib Ullah and Faouzi Alaya Cheikh. A directed sparse graphical model for multi-target tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1816–1823, 2018.
26. Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE transactions on information theory*, 37(1):145–151, 1991.
27. Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *European Conference on Computer Vision*, pages 17–35. Springer, 2016.