

# Real-time Pose Estimation from Images for Multiple Humanoid Robots

Arash Amini<sup>1</sup>, Hafez Farazi<sup>2</sup>, and Sven Behnke<sup>2</sup>

*University of Bonn, Computer Science Institute VI, Autonomous Intelligent Systems  
Friedrich-Hirzebruch-Allee 8, 53115 Bonn, Germany  
amini@uni-bonn.de<sup>1</sup>, {farazi, behnke}@ais.uni-bonn.de<sup>2</sup>*

**Abstract.** Pose estimation commonly refers to computer vision methods that recognize people’s body postures in images or videos. With recent advancements in deep learning, we now have compelling models to tackle the problem in real-time. Since these models are usually designed for human images, one needs to adapt existing models to work on other creatures, including robots. This paper examines different state-of-the-art pose estimation models and proposes a lightweight model that can work in real-time on humanoid robots in the RoboCup Humanoid League environment. Additionally, we present a novel dataset called the HumanoidRobotPose dataset. The results of this work have the potential to enable many advanced behaviors for soccer-playing robots.

## 1 Introduction



Fig. 1: One sample image of the introduced dataset with the estimated poses is shown on the left. Our model’s predicted heatmaps of all keypoints and limbs are displayed in the middle and right, respectively.

The 2D humanoid pose estimation problem aims to detect and localize keypoints and parts and infer the limb connections to reconstruct the existing human poses from images. The human pose estimation problem’s importance arises from the fact that this task has many applications in various areas such as human-computer interaction and action recognition. In this work, we address the real-time pose estimation problem for humanoid robots (see Fig. 1). The shape

similarity between humanoid robots and persons is a double-edged sword. On the one hand, it enables us to start with existing methods designed for persons, but on the other hand, it adds additional difficulty to our problem not to confuse humans with humanoid-robots, especially in adult-size league. In general, the attempts to address the pose estimation problem for multiple persons]can be categorized as either top-down or bottom-up approaches.

In the top-down models, the procedure includes two distinct steps. The first step is to detect individual people, while the particular pose is estimated in the next step. One of these models' disadvantages is that the performance of the model is tightly correlated to the person detector performance. Although the state-of-the-art (SOTA) results are derived from this type of models, including Cascaded Pyramid Network [3] and High-Resolution Net (HRNet) [24], the runtime of such approaches is negatively affected by the number of persons present, as a single-person pose estimator is run for each detection. Hence, the computational cost linearly increases with more persons, so the performance is often not real-time.

In contrast, bottom-up approaches detect body joints and group them into individuals simultaneously; therefore, they are less dependant on the number of persons in the image. One of the bottom-up method's main challenges is to group the detected keypoints in a real-time manner accurately. Recent approaches [2,11,19] utilize a greedy algorithm to group the detected keypoints into individual instances. Moreover, the bottom-up method's performance is more vulnerable to the different scales of the persons in a given image compared to the top-down approaches. To alleviate this issue, previous works exploit the scale search method [2] or rely on high-resolution input size [19]. These solutions are increasing the inference time though. A time-efficient method predicting keypoints at higher resolution was introduced by Cheng et al. [4], narrowing the performance gap between bottom-up and top-down models.

This paper opts for a bottom-up approach designed for 2D pose estimation of multiple humanoid robots. We made this choice because, in top-down methods, the inference time is generally much higher than in bottom-up approaches, so they will not be suitable for RoboCup real-time applications. Furthermore, we wanted to avoid the cost of annotating bounding boxes. We remedy our bottom-up model scale variations problem by using feature pyramid structure [13] through utilizing high-resolution feature maps.

Despite the availability of several large-scale benchmark datasets such as MPII Human Pose [1] and MS COCO [14] for the task of human pose estimation, we cannot fully utilize them because of differences between robots and humans, such as types and sizes. Thus, we present a new pose dataset of robots from the RoboCup Humanoid League. The code and dataset of this paper are publicly available.<sup>1</sup> In summary, we make the following contributions:

- We propose a deep learning model specifically designed to address the 2D pose estimation problem for multiple humanoid robots.

<sup>1</sup> <https://github.com/AIS-Bonn/HumanoidRobotPoseEstimation>.

- We introduce a new dataset, namely the HumanoidRobotPose dataset, consisting of robots from the RoboCup Humanoid League.
- We demonstrate that the proposed real-time light-weight model outperforms the SOTA bottom-up methods in our application.

## 2 Related Work

Although there are some works on the detection and tracking of humanoid robots [6, 7], to the best of our knowledge, there is no previous work that addresses humanoid robot’s pose estimation, which works on a variety of robots. Giambattista et al. [5] propose a gesture-based communication between Nao robot that utilizes OpenPose [2] for Nao robot pose estimation. Note that pose estimation on a single standardized Nao robot type is significantly easier than what we need in the Humanoid League. We have to address various unseen robots with different colors, kinematic shapes, and sizes.

**Top-down:** Most of the existing top-down methods exploit human detector models such as Feature Pyramid Networks [13], and Faster R-CNN [21]. Papan-dreou et al. [20] propose one of the first top-down models which employ the Faster R-CNN for the person detector step and present a new representation for keypoints, which is a mixture of binary activation heatmap and the corresponding offset. The most recent top-down approaches which obtain SOTA accuracy are the Cascaded Pyramid Network (CPN) introduced by Chen et al. [3], where multi-scale feature maps from different layers of the GlobalNet are integrated with an online hard keypoint mining loss for difficult-to-detect joints, and the model presented by Sun et al. [24] that improves the heatmap estimation using high-resolution representations and multiple branches of different resolutions.

**Bottom-up:** The recent architectures [2, 12, 16, 18, 19] take advantage of the confidence maps to detect the keypoints. Kreiss et al. [11] introduce a combination of confidence maps and vectorial parts for keypoints detection. Moreover, there are different approaches for encoding the part association used in the SOTA bottom-up models. OpenPose [2] introduces the Part Affinity Fields (PAFs) method to learn the body parts associations by encoding the location and direction, offset regression that uses the displacements of the keypoints [18, 19], and tag heatmap, which produces a heatmap as a tag for each keypoint heatmap [12, 16]. Pose Partition Networks [18] present a dense regression approach over all the keypoints to generate individuals’ partitions using the embedding spaces.

## 3 Pose Estimation Model

In this section, we present our real-time bottom-up approach to pose estimation of multiple humanoid robots. The aim is to predict the part coordinates and the part associations to build robot poses. In the following, we first describe the

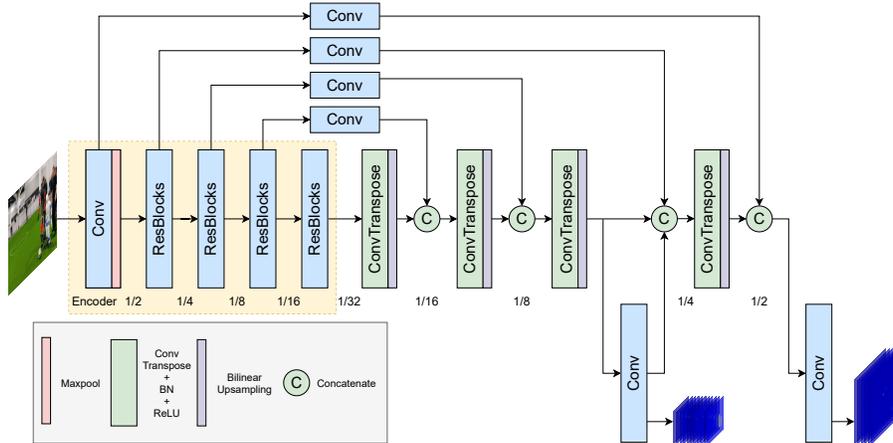


Fig. 2: The architecture of the proposed single-stage encoder-decoder model. The model predicts heatmaps of both keypoints and limbs for scale 1/4 and only heatmaps of keypoints for scale 1/2, where each scale is supervised with an intermediate loss.

model, then explain the keypoint detection and the part association methods in detail.

### 3.1 Network Architecture

Following the successful results of NimbRo-Net [8] and NimbRo-Net2 [22], we decided to utilize a similar architecture. This decision ensures that later we can combine this model with NimbRo-Net2 to have a unified network for multiple tasks related to the humanoid league. The proposed network is depicted in Fig. 2.

Our model is an encoder-decoder network which takes an RGB image of size  $w \times h$ . We observe that it is required to use a deeper encoder than the decoder to create a powerful feature extractor. The encoder is a pre-trained ResNet model [10], in which the last fully connected and global average pooling layers are eliminated. The first layer is a  $7 \times 7$  convolutional with stride 2, followed by a max-pooling layer. The rest of the encoder network consists of four modules of residual blocks with higher depths and lower resolutions as the number of modules increases. Each residual block consists of two or three convolutional layers, depending on the selected ResNet architecture, followed by batch normalization and ReLU activation and a shortcut connection. More fine-grained spatial information is present in the early layers, while in the final layers, the network extracts more semantic information.

In the decoder part, we utilize lateral connections from different parts of the encoder, which allows us to maintain the high-resolution information. For every lateral connection, we apply  $1 \times 1$  convolution to generate a fixed number of channels. The decoder network has a feature pyramid structure involving

four modules. At each level of the pyramid, the previous level’s output is fed to the  $3 \times 3$  transposed convolution followed by a bilinear upsampling to obtain a fixed number of higher-resolution features. These upsampled features are concatenated with the features from the corresponding lateral connection. Similar to the encoder, ReLU and batch normalization is used to get the final output of the module.

As high-resolution feature maps are essential for precise keypoint localization [25], we leverage two scales of the feature pyramid hierarchy, i.e., 1/4 and 1/2 resolutions. As a result, the keypoints heatmaps  $\hat{\mathbf{K}}^s$  at each scale  $s$  is generated by performing a final convolution on extracted features. As depicted in Fig. 2, we have two scales of keypoint heatmaps with intermediate supervision, inspired by HigherHRNet [4]. The final keypoints heatmaps  $\hat{\mathbf{K}}$  are the average over the predictions generated by these two scales after upsampling to the same resolution as the input image in order to achieve accurate high-resolution predictions. Note that only one scale of limbs heatmaps  $\hat{\mathbf{L}}$  is utilized, as we observe that following the same approach as keypoints yields performance drop.

### 3.2 Keypoint Detection and Part Association

The ground truth heatmaps of the keypoints in a given image can be represented as the set  $\mathbf{K} = \{\mathbf{K}_1, \mathbf{K}_2, \dots, \mathbf{K}_P\}$ , where  $\mathbf{K}_p \in \mathbb{R}^{w' \times h'}$ ,  $p \in \{1, 2, \dots, P\}$ , and  $P$  is the total keypoints of a robot instance, which is equal to six for our dataset (see Fig. 3 (left)). The heatmap  $\mathbf{K}_p$  with the resolution of  $w' \times h'$  includes the Gaussian heatmaps of the  $p$ th part of all the robot instances. Let  $x_{p,n}$  and  $y_{p,n}$  be the location of the  $p$ th part of the  $n$ th instance, where  $n \in \{1, 2, \dots, N\}$  and  $N$  is the total number of existing robots with the visible  $p$ th part in an image. To embed the position of the annotated  $p$ th part, we use the 2D unnormalized Gaussian distribution with the center of  $(x_{p,n}, y_{p,n})$  and the standard deviation  $\sigma$ , which is fixed for all the parts:

$$\mathbf{K}_p(x, y) = \exp\left(-\frac{(x - x_{p,n})^2 + (y - y_{p,n})^2}{2\sigma^2}\right). \quad (1)$$

Due to occlusion or proximity of the robot instances in a given image, we utilize the pixel-wise max operation on Eq. 1 to preserve the Gaussian peak of the  $p$ th part for each instance.

For limbs, the ground truth heatmaps in a given image can be expressed as the set  $\mathbf{L} = \{\mathbf{L}_1, \mathbf{L}_2, \dots, \mathbf{L}_C\}$ , where  $\mathbf{L}_c \in \mathbb{R}^{w' \times h'}$ ,  $c \in \{1, 2, \dots, C\}$ , and  $C$  is the total limbs of a robot instance that is five in this work (see Fig. 3 (left)). Note that the intended utility of limbs is only to encode the relations between keypoints, so they do not necessarily lie on actual robot limbs. Therefore, to encode a limb’s position, first, we compute a line segment between two keypoints and mark all of the points that lie on such limb, following the approach proposed by Li et al. [12]. Then having these offsets, the final Gaussian heatmap of each limb is generated by an unnormalized Gaussian distribution with the standard deviation  $4\sigma$  that controls the spread of the Gaussian peak in the same way as for the keypoint heatmap. The final limb heatmap is the average of all the robots’ limb appearing

in the image. In contrast to the PAFs method that encodes each limb in two channels as vector directions, we encode each limb type in a single channel. This simpler approach for encoding the limbs is enough for our application since our experiments show better performance than the PAF method.

### 3.3 Loss

We use the mean square error to compute the loss between the predicted heatmaps and the ground truth heatmaps for both keypoints and the limbs.

$$\mathcal{L}_{keypoints} = \frac{1}{2P} \sum_{s \in \{\frac{1}{4}, \frac{1}{2}\}} \sum_{p=1}^P \mathbf{W} \cdot \left\| \mathbf{K}_p^s - \hat{\mathbf{K}}_p^s \right\|_2^2, \quad (2)$$

$$\mathcal{L}_{limbs} = \frac{1}{C} \sum_{c=1}^C \mathbf{W} \cdot \left\| \mathbf{L}_c - \hat{\mathbf{L}}_c \right\|_2^2, \quad (3)$$

where  $W$  is a binary mask with  $W = 0$  when the annotation is missing in the image, and  $s$  is the scale of the predicted heatmaps. Finally, the total loss used to train the network is the sum of the keypoint loss (2) and the limb loss (3).

### 3.4 Post Processing

By performing Non-Maximum Suppression ( $3 \times 3$  kernel) on the predicted keypoint heatmaps, we obtain the peak of each Gaussian heatmap and the location of its corresponding keypoint for the robot instances. We use the detected limb’s heatmaps to acquire the candidate connections between the keypoints as [2]. As there are multiple robot instances in an image, it is required to group the keypoints to determine the poses corresponding to the correct individuals. Having the set of keypoints and the connection candidates, we employed the proposed greedy algorithm by Cao et al. [2] to solve the assignment problem and obtain the final pose of all robot instances. In this algorithm, instead of considering the fully connected graph, the goal is to obtain the minimum spanning tree of the pose instance and assign the adjacent tree nodes independently, resulting in a well-approximate solution with efficient computational cost.

## 4 Dataset

This section explains the paper’s additional contribution, the HumanoidRobot-Pose dataset, including data collection and annotation procedures and the evaluation metrics used for this dataset.

Our goal was to collect a dataset containing both single and multiple robots to simulate the RoboCup’s real conditions. We gathered many YouTube videos from the RoboCup Humanoid League, as well as some in-house videos and ROS bags. Some videos originate from the qualification videos, which only demonstrate a

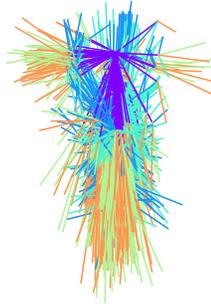
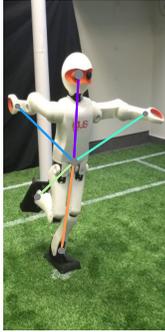


Fig. 3: An example of the annotations depicted in left. Visualization for the distribution of pose variability by considering the head as the origin is shown in right. Note that we did not choose trunk joint as the origin to show the dataset’s variability better.

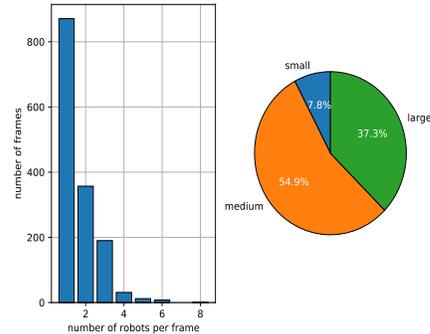


Fig. 4: Statistics for our dataset. The diversity of the number of robot instances is illustrated in left, and the scale proportions of robot instances is shown in right. The definition of small, medium and large scale is identical to the COCO dataset.

specific robot; therefore, they only consist of a single pose. To include videos with multiple robots and increase the diversity of robots in the dataset, we also employ videos from drop-in games and round-robin competitions. These videos are from recent years and contain various view angles, lens distortions, brightness, and robots. Note that in most of the videos, there are humans present in the pictures, e.g., the robot handler, the referee, and audiences around the field. Overall, we annotated over 1.5K manually selected frames from 23 videos with around 2.3k robot instances. These frames include teen and adult sized robots and contain more than ten different robot types. About 30 percent of the dataset was exclusively used for testing. Note that testing frames were collected from different videos than the training videos.

#### 4.1 Data Annotation

For data annotation, we used Supervise.ly<sup>2</sup>, a web-based data annotations and management tool. We decided to ignore the truncated or severely occluded points in the image, which are usually considered invisible keypoints. For each robot, six keypoints are annotated, including head, trunk, hands, and feet. The head keypoint is important, for instance, to estimate the height of the goalie robot. We use these few keypoints to avoid annotation costs; however, they can be easily extended to more keypoints. We define a minimal pose representation by five limbs from these keypoints, which would be sufficient for the current soccer behavior level. The annotation for a robot instance is illustrated in Fig. 3 (left).

<sup>2</sup> <https://supervise.ly>

To show the diversity of our dataset, we visualize the variability of annotated poses in Fig. 3 (right) and statistics of the number and scales of the robot instances are presented in Fig 4. About 60 percent of the collected frames contain single pose instances and robot instances with medium scale, i.e.,  $[32^2 < \text{segment area} < 96^2]$ , where the segment area of a robot instance is measured using the size of the minimum encapsulating rectangle of the annotated keypoints. Our definition of the size scales is identical to the COCO dataset <sup>3</sup>.

## 4.2 Evaluation Metrics

We use the Object Keypoint Similarity (OKS) metric from COCO keypoint dataset [23]. The OKS of a robot  $r$  between the detected keypoint ( $\hat{y}_{i,r}$ ) and its corresponding ground truth ( $y_{i,r}$ ) can be written as follows:

$$OKS(\hat{y}_r, y_r) = \frac{\sum_i e^{-\frac{\|\hat{y}_{r_i} - y_{r_i}\|_2^2}{2a^2k_i^2}} \delta(v_i > 0)}{\sum_i \delta(v_i > 0)}, \quad (4)$$

where  $k_i$  is a constant specific to each keypoint, which is equal for all keypoints in our dataset,  $a$  is the segment area of the robot instance measured in pixels, and  $v_i$  is the keypoint visibility flag in the ground truth ( $v_i = 0$  for the invisible keypoint). The OKS metric is robust to the number of visible keypoints as it gives equal importance to the robot instances with different numbers of visible keypoints. The evaluation metrics used for the proposed dataset are as following: AP (the mean average precision over 10 OKS thresholds = [0.50:0.05:0.95]), AP<sup>50</sup> (AP at OKS threshold = 0.50), AP<sup>75</sup>, AP<sup>M</sup> for medium scale robot instances, AP<sup>L</sup> for large scale instances, and AR (the mean of average recall over 10 OKS thresholds).

## 5 Experiments

We compare the proposed method with SOTA bottom-up approaches on the HumanoidRobotPose dataset. These approaches are OpenPose [2], Associative Embedding (AE) [16], PifPaf [11], and HigherHRNet [4]. OpenPose [2] utilizes confidence maps to localize the keypoints and PAFs to encode the body parts' location and orientation. For grouping the detected keypoints, the greedy algorithm is proposed in which each part is scored, computing the line integral on the corresponding PAF. Associative Embedding (AE) [16], merges the stacked hourglass architecture [17] with associative embedding.

PifPaf [11] proposes Part Intensity Field to detect and localize the keypoints and Part Association Fields to associate body parts with each other.

HigherHRNet [4] is using an adopted top-down model as the backbone with a transposed convolution module to predict higher resolution heatmaps for the keypoints detection. Similar to the AE approach, in HigherHRNet, the associative embedding is employed to parse the poses. Following the configurations

<sup>3</sup> <https://cocodataset.org/#keypoints-eval>



Fig. 5: Four sample results from the test set. Methods used from top to bottom: OpenPose [2], AE [16], PifPaf [11], HigherHRNet [4], and our model.

provided by the original papers, we reported the details of models and the inference time evaluated on NimbRo-OP2X robot hardware [9] in Table 1.

For all methods, the hyperparameters are tuned to achieve the best possible results. Our model is trained using the AdamW [15] optimizer with learning rate of  $10^{-4}$ , batch size 16 and weight decay of  $10^{-4}$  for the total 200 epochs. Note that the encoder is initialized by pre-trained ResNet weights on ImageNet. We conduct data augmentation that includes random horizontal flip, random rotation, random scaling, and random translation during training.

The results on the test set are reported in Table 2. The reported results are achieved without performing the flip test or the multi-scales test for preserving the methods to be real-time. Our proposed method with ResNet18 backbone outperforms the best existing methods in all metrics except for large scale when we train the models from scratch on our dataset (see Table 2). Note that com-

Table 1: Details for the used methods.

Method	Input Size	Backbone	Params	GFLOPs	FPS
OpenPose [2]	368	VGG19	25.8M	159.8	14
AE [16]	512	Hourglass	138.8M	441.6	5
PifPaf [11]	385	ShuffleNetV2	<b>9.4M</b>	46.3	13
HigherHRNet [4]	512	HRNet-W32	28.6M	94.7	13
Ours	384	ResNet18	12.8M	<b>28.0</b>	<b>48</b>

Table 2: Results on the test set.

Method	AP	AP <sup>50</sup>	AP <sup>75</sup>	AP <sup>M</sup>	AP <sup>L</sup>	AR	AR <sup>50</sup>	AR <sup>75</sup>	AR <sup>M</sup>	AR <sup>L</sup>
OpenPose [2]	67.9	80.0	70.0	73.8	73.1	68.7	80.1	70.4	74.8	74.4
AE [16]	62.9	71.9	64.1	64.0	72.9	64.6	73.9	65.6	64.7	76.0
PifPaf [11]	76.1	81.6	75.6	76.0	<b>91.0</b>	77.9	83.6	77.2	77.7	<b>93.0</b>
HigherHRNet [4]	73.4	84.1	75.6	80.3	78.7	76.2	85.3	77.2	81.4	83.0
Ours	<b>78.1</b>	<b>84.6</b>	<b>79.6</b>	<b>87.5</b>	80.2	<b>79.4</b>	<b>85.4</b>	<b>80.6</b>	<b>88.4</b>	81.6

pared to other baselines, our model can utilize our limited dataset better. Based on AP results of medium and large scales, our model can better handle the different scales than the other approaches. Moreover, the strict metric results demonstrate that the predicted pose instances are more accurate compared to the other methods due to the high-resolution predictions. Fig. 5 illustrates some samples of estimated poses for all the approaches.

## 6 Ablation Study

This section investigates different backbones for the encoder part of our model and the importance of employing multi-scale predictions in our approach. As shown in Table 3, although applying a deeper encoder helps achieve better performance, it negatively affects the inference time of the model. Moreover, AP results demonstrate that without multi-scale heatmaps, the accuracy of predicted keypoints drops.

Table 3: Ablation study: the effectiveness of backbones and multi-scale predictions on the test set.

Backbone	Multi-scale	AP	AP <sup>M</sup>	AP <sup>L</sup>	Params	GFLOPs	FPS
ResNet18		76.6	83.6	82.0	<b>12.4M</b>	<b>17.8</b>	<b>83</b>
ResNet18	✓	78.1	<b>87.5</b>	80.2	12.8M	28.0	48
ResNet50	✓	<b>78.6</b>	86.0	<b>82.1</b>	27.0M	47.7	25

## 7 Conclusion

In this paper, we presented a lightweight bottom-up model for estimating multiple humanoid robot poses in real-time. We showed that our proposed model is capable of multi-robot pose estimation on NimbRo-OP2X robot hardware and is more suitable for the RoboCup humanoid league in comparison with other SOTA models. For the future, we will use this model for advanced soccer behavior decisions like recognizing rival robots' actions or anticipating the ball's movement direction before the kicking motion. Since the developed model is very similar to NimbRo-Net2, we will combine them to produce a unified network for diverse perception tasks in RoboCup.

*Acknowledgment:* This work was partially funded by grant BE 2556/16-2 (Research Unit FOR 2535 Anticipating Human Behavior) of the German Research Foundation (DFG).

## References

1. Andriluka, M., Pishchulin, L., Gehler, P., Schiele, B.: 2D human pose estimation: New benchmark and state of the art analysis. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2014)
2. Cao, Z., Hidalgo, G., Simon, T., Wei, S.E., Sheikh, Y.: Openpose: Realtime multi-person 2D pose estimation using part affinity fields. IEEE Transactions on Pattern Analysis and Machine Intelligence (2021)
3. Chen, Y., Wang, Z., Peng, Y., Zhang, Z., Yu, G., Sun, J.: Cascaded pyramid network for multi-person pose estimation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
4. Cheng, B., Xiao, B., Wang, J., Shi, H., Huang, T.S., Zhang, L.: Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
5. Di Giambattista, V., Fawakherji, M., Suriani, V., Bloisi, D.D., Nardi, D.: On field gesture-based robot-to-robot communication with nao soccer players. In: Robot World Cup XXIII. Springer (2019)
6. Farazi, H., Behnke, S.: Real-time visual tracking and identification for a team of homogeneous humanoid robots. In: Robot World Cup XX. Springer (2016)
7. Farazi, H., Behnke, S.: Online visual robot tracking and identification using deep lstm networks. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2017)
8. Farazi, H., Ficht, G., Allgeuer, P., Pavlichenko, D., Rodriguez, D., Brandenburger, A., Hosseini, M., Behnke, S.: NimbRo robots winning RoboCup 2018 humanoid adultsized soccer competitions. In: Robot World Cup XXII. Springer (2018)
9. Ficht, G., Farazi, H., Brandenburger, A., Rodriguez, D., Pavlichenko, D., Allgeuer, P., Hosseini, M., Behnke, S.: NimbRo-OP2X: Adult-sized open-source 3D printed humanoid robot. In: IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids) (2018)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)

11. Kreiss, S., Bertoni, L., Alahi, A.: PifPaf: Composite fields for human pose estimation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
12. Li, J., Su, W., Wang, Z.: Simple Pose: Rethinking and improving a bottom-up approach for multi-person pose estimation. In: AAAI (2020)
13. Lin, T.Y., Dollar, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
14. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common objects in context. In: European Conference on Computer Vision (ECCV) (2014)
15. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: International Conference on Learning Representations (ICLR) (2019)
16. Newell, A., Huang, Z., Deng, J.: Associative embedding: End-to-end learning for joint detection and grouping. In: Advances in Neural Information Processing Systems (2017)
17. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. In: European Conference on Computer Vision (ECCV) (2016)
18. Nie, X., Feng, J., Xing, J., Yan, S.: Pose partition networks for multi-person pose estimation. In: European Conference on Computer Vision (ECCV) (2018)
19. Papandreou, G., Zhu, T., Chen, L.C., Gidaris, S., Tompson, J., Murphy, K.: Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In: European Conference on Computer Vision (ECCV) (2018)
20. Papandreou, G., Zhu, T., Kanazawa, N., Toshev, A., Tompson, J., Bregler, C., Murphy, K.: Towards accurate multi-person pose estimation in the wild. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
21. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems (2015)
22. Rodriguez, D., Farazi, H., Ficht, G., Pavlichenko, D., Brandenburger, A., Hosseini, M., Kosenko, O., Schreiber, M., Missura, M., Behnke, S.: RoboCup 2019 AdultSize winner NimbRo: Deep learning perception, in-walk kick, push recovery, and team play capabilities. In: Robot World Cup XXIII. Springer (2019)
23. Ruggero Ronchi, M., Perona, P.: Benchmarking and error diagnosis in multi-instance pose estimation. In: IEEE International Conference on Computer Vision (ICCV) (2017)
24. Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., Liu, W., Xiao, B.: Deep high-resolution representation learning for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020)
25. Xiao, B., Wu, H., Wei, Y.: Simple baselines for human pose estimation and tracking. In: European Conference on Computer Vision (ECCV) (2018)