Detection and Localization of Features on a Soccer Field with Feedforward Fully Convolutional Neural Networks (FCNN) for the Adult-Size Humanoid Robot Sweaty

Fabian Schnekenburger, Manuel Scharffenberg, Michael Wülker, Ulrich Hochberg, Klaus Dorer
Faculty of Mechanical and Process Engineering
University of Applied Sciences Offenburg
Badstr. 24, 77652 Offenburg, Germany
(Fabian.Schnekenburger, Manuel.Scharffenberg, Michael.Wuelker, Ulrich.Hochberg, Klaus.Dorer)@hs-offenburg.de
https://sweaty.hs-offenburg.de/en

***Abstract*—For the RoboCup Soccer AdultSize League the humanoid robot Sweaty uses a single fully convolutional neural network to detect and localize the ball, opponents and other features on the field of play. This neural network can be trained from scratch in a few hours and is able to perform in real-time within the constraints of computational resources available on the robot. The time it takes to precess an image is approximately $11\,\mathrm{ms}$. Balls and goal posts are recalled in $99\,\%$ of all cases ($94.5\,\%$ for all objects) accompanied by a false detection rate of $1.2\,\%$ ($5.2\,\%$ for all). The object detection and localization helped Sweaty to become finalist for the RoboCup 2017 in Nagoya.**

I. Introduction

The detection of the ball and other objects on the field of play of RoboCup Soccer competitions have become more and more challenging since the introduction of multi-colored balls, white goal posts and of artificial turf. Therefore, achieving reliable results with traditional algorithms based on color segmentation and edge detection within captured images becomes increasingly difficult. Differing and unexpected patterns on the – moving – soccer ball, changing lighting conditions and varying contrast of worn-out lines require constant parameter tuning.

A. Fully Convolutional Neural Networks

In recent years, neural networks, especially convolutional neural networks have revolutionized the field of computer vision. There are already a number of teams in the RoboCup Humanoid Soccer leagues that use neural networks to improve the accuracy of their detection, however most still rely on algorithms based on color segmentation, histograms and detecting shapes to extract a region of interest (ROI) beforehand. The neural network is then applied to these ROIs. For this work, a deep fully convolutional neural network architecture is used for the identification and localization of objects on the field of play analyzing the whole camera image. With this information the location and attitude of the robot on the field can be deduced.

It is possible to completely replace traditional computer vision approaches with a fully convolutional neural network. This is achieved in a manageable time and for training, even when starting from scratch without a pre-trained network. Since the FCNN maps an input image pixel by pixel to a probability map for a particular object class ("heat-map") the image size and resolution can easily be changed. It turned out that not only the ball, but many other object types like the goal posts, opponents and characteristic line elements could be trained and localized. Including training images captured in a wide range of lighting conditions made the object recognition insensitive to variations in lighting. During a game the deployed FCNN worked in real-time.

B. Humanoid Robot Sweaty

The humanoid robot "Sweaty" was finalist at the RoboCup Soccer AdultSize League in 2016 and 2017. Sweaty's name alludes to the innovation that particularly hard working motors are cooled by evaporating water from an enclosing felt [1], like humans sweating. Overall 32 motors actuate Sweaty's joints. Sweaty is $172\,\mathrm{cm}$ tall and weighs $25.6\,\mathrm{kg}$ (Fig. 1).

To perceive items and opponents on the field of play, Sweaty is equipped with two cameras with a resolution of 1280×1024, which cover a field of vision of $160°$ and operate at up to 60 frames per second (Fig. 2). To track its movements on the field, Sweaty uses three inertial measurement units (IMU), each providing three components of angular velocity and three components of linear acceleration. For image processing Sweaty carries a i7-$3.5\,\mathrm{GHz}$-CPU running Ubuntu 16.04 and a GeForce-GTX-760-GPU.

Once objects have been detected their distance is calculated by triangulation. Therefore the base point of obstacles, opponents and goal posts has to be identified. The only preprocessing step is to convert the image data to float values and dividing by 255 to get normalized values between zero and one.

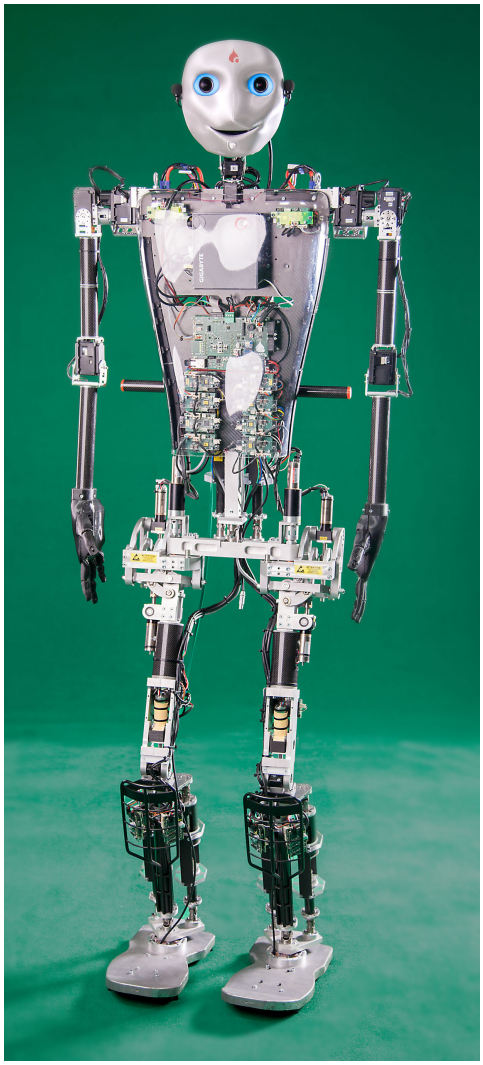

Figure 1. RoboCup AdultSize League humanoid robot Sweaty

II. Related Work

In which way neural networks have been used for the recognition of objects on the field of play within the RoboCup Soccer community is shortly summarized. A few references were taken from the meanwhile vast area of the classification and localization of objects in images and the segmentation of images with the help of convolutional neural networks (CNNs).

A. Neural Network Applications in RoboCup Soccer

In the context of RoboCup Soccer competitions, camera detection of entities on the field of play (ball, goals, field area, lines or players) were identified with the help of neural networks by various teams. In an early stage very small neural networks assisted with the classification from hue histograms [2] or orientation histograms [3] for regions of interest, which had been chosen beforehand by other image processing methods. Unprocessed camera pixel values were first taken from very small sample windows [4], but later from larger regions of interest [5]. In this case up to three convolutional neural network layers and two fully-connected layers were used for the detection of other NAO robots. The size of the neural networks resulted in a computational load, which is too high for real-time deployment on the NAO robots being investigated. The approach by [5] was extended to humanoid robots in general [6] and compared various neural net configurations (LeNet, SqueezeNet, and GoogleLeNet) and published training data.

Recently [7] addressed the problem to reduce the computational load in the case of NAO robots by using an XNOR-net respectively a SqueezeNet achieving a classification performance for NAO robots of approximately $97\,\%$ taking around $1\,\mathrm{ms}$ for a proposal. For the case of detecting the ball [8] benchmarked as many as 252 network designs regarding precision, recall, and execution time. The recall is almost insensitive to the network design and precision drops from around $98\,\%$ to $82\,\%$ in some cases, but is uncorrelated to the execution time.

In all cases discussed in the preceding paragraphs, classified objects are only localized implicitly by the position of the selected region of interest within the image. To localize the soccer ball in full camera images was undertaken by Speck et al. [9], [10] using three convolutional layers followed by two fully connected layers for the horizontal respectively vertical projection of the images. In the horizontal projection $81\,\%$ of the peaks were detected and $75\,\%$ in the vertical direction.

B. Classification and Segmentation by CNNs

The vision system in RoboCup Soccer for humanoid robots aims at identifying and localizing a small number of objects and features on the field of play. Particularly in the case of a fast moving ball it is also important to do so in real-time at

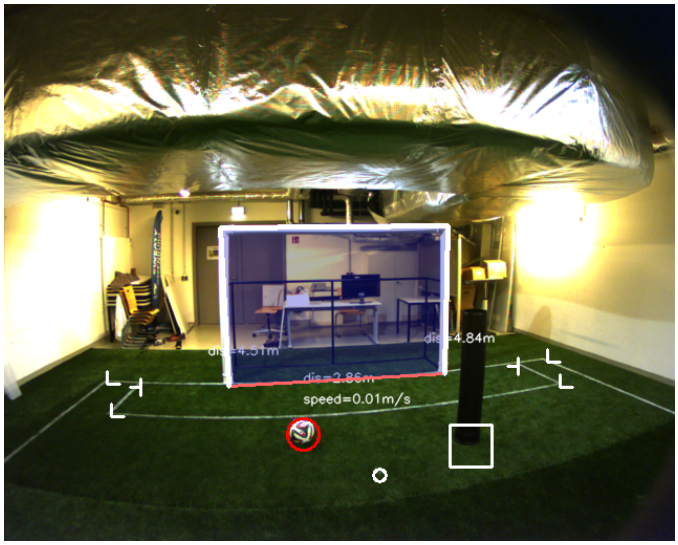

Figure 2. Sample view of Sweaty's vision system with object detections overlaid: ball circled in red, feet of the goal post joint by a goal line, foot of the obstacle indicated by white square, L–line corners, T–line junctions, white circle–penalty point, X–line crossing (not in view)

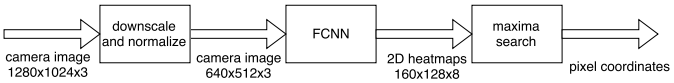

Figure 3. Image processing pipeline

rates of more than 20 Hz. It is also necessary to find several instances of the same object category like line corners (see Fig. 2).

To start with, the use of deep convolutional neural networks followed by one or more fully-connected layers lead to impressive results classifying objects out of up to a thousand object categories. Pure classification was extended to the detection of multiple occurrences of the same object category and localization by bounding boxes. The various approaches (Faster R-CNN, R-FCN, and SSD) with different feature extractors (VGG, Resnet, Inception, and MobilNet) have been benchmarked in [11].

Driven by the needs of image processing for assisted and autonomous driving as well as medical applications semantic segmentation tries to allocate every pixel of the image to an object category. For deep convolutional neural networks the fully-connected layers for classification are dropped and the resolution lost by pooling (down-sampling) is recovered by up-sampling (deconvolution). Details of the image are retained by skip-connections between layers of the same resolution and the resolution of the output is the same as for the input image allowing feedforward end-to-end training by back-propagation. For this work reference [12] served as a starting point. Similar approaches are reported as the SegNet [13], the V-Net [14], and the U-Net [15].

III. Neural Network for Localization

Using a FCNN that can perform in real-time when applied on full resolution images with limited resources requires lightweight models. In contrast to image classification applications with hundreds or even thousands of different object categories, the environment on a RoboCup Soccer field is distinctively less complex. A neural network designed for RoboCup is therefore able to perform well with only a fraction of the number of parameters. Such small lightweight models are generally less prone to over-fitting even when trained on a relatively small dataset.

A. Network Architecture

We propose three similar architectures, with many layers but a comparatively low number of feature maps per layer. All of which have significantly less then one million parameters.

1) SweatyNet-1: The network has an encoder-decoder design similar to SegNet [13]. The architecture is illustrated in Fig. 6. The encoder consists of 12 convolutional layers. Batch normalization and ReLU activation function are applied after each convolution. The number of filters in the first layer is eight and is doubled after each of the four max-pooling layers. The decoder path is shorter, with only six convolutional layers and two upscaling layers. Bilinear upscaling is used instead of transposed convolutions that are known to produce checkerboard artifacts [16]. All convolutions are computationally efficient 3×3 kernels. The input resolution is 640×512, output resolution is four times lower to reduce computational effort and costly data transfers from the GPU. The number of output channels equals the number of classes.

Figure 4. Sample training images demonstrating variations in lighting conditions, white balance and complexity of scene

Skip connections exist between layers of the same resolution from the encoder to the decoder path to have finer grained spatial information from higher layers available in the decoder. In addition there are residual connections between the pooling layers of the encoder path to speed up training and to achieve more efficient parameter usage.

2) SweatyNet-2: Is a variation of SweatyNet-1 with fewer layers and less parameters to reduce inference time. The dotted layers in Fig. 6 are removed.

3) SweatyNet-3: Reduces the number of channels with 1×1 convolutions before every convolution with 3×3 filter kernels to decrease the number of parameters and the inference time (Fig. 7).

B. Training Data and Teaching Signal

Most of the training data was collected on our field of play conforming to the rules of the RoboCup Humanoid Leagues. In total the dataset consists of 2400 images, 2150 are used for training while 250 are used for testing. Images were taken from the robots point of view at arbitrary locations and viewing directions in different light conditions. To be able to detect a wide variety of robot feet, additional training data was gathered from *Youtube* videos of prior RoboCup competitions.

To increase the variance in the training, the following modifications are applied to the images during training: random horizontal flips, random crops, random rotations, random blur, and random brightness adjustments.

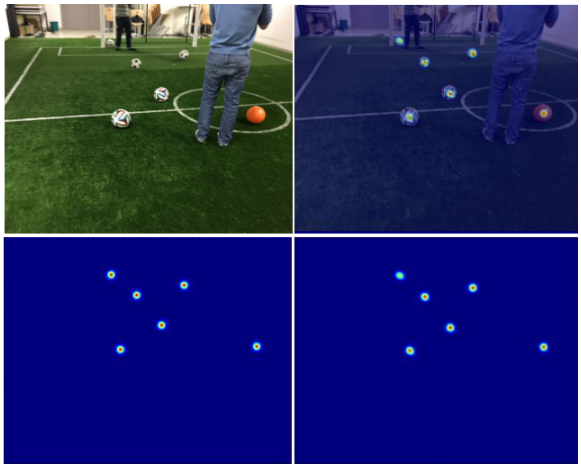

Figure 5. Training data teaching signal and network output (heat-map). The top left image shows a training image and the bottom left accordingly labeled balls. The top right image is the training image overlaid with the heat-map, shown on its own at the bottom right

Currently we discriminate between the following object classes: ball, goal post, crossing lines, penalty mark, corners of lines, T-junctions of lines, opponents, and obstacles.

All these objects are labeled at a single point. Opponents, obstacles and goal posts are marked at the center of their base, the other classes are labeled at their center. To train the network, these labels are transformed to 2D-maps of feature locations. Instead of marking a single pixel a normal distribution with a standard deviation of $\sigma = 4$ is centered around the label coordinates. In this way the training error is reduced, even if the peak at the output of the network is off by a few pixels. A pixel-accurate prediction (standard deviation $\sigma = 0$) would be almost impossible to learn. The left column of Fig. 5 shows a training image and the corresponding 2D-map for marked balls. The labels are saved as text-files; during training the required 2D-maps are created from these labels.

C. Post-Processing

Since the network does not directly predict pixel coordinates but two-dimensional probability maps for each category, pixel coordinates are determined with a simple peak detection algorithm. The pixels around a local maximum are used to determine the actual peak position with sub-pixel accuracy.

Data: matrix for single category in output
while *matrix maximum value* $>$ *threshold* **do**
  get coordinates of maximum;
  mask pixels around maximum;
  determine new matrix maximum value;
end

Algorithm 1: Peak detection algorithm

D. Training

All models are trained by minimizing the mean squared error (MSE) between the output and the teaching signal with

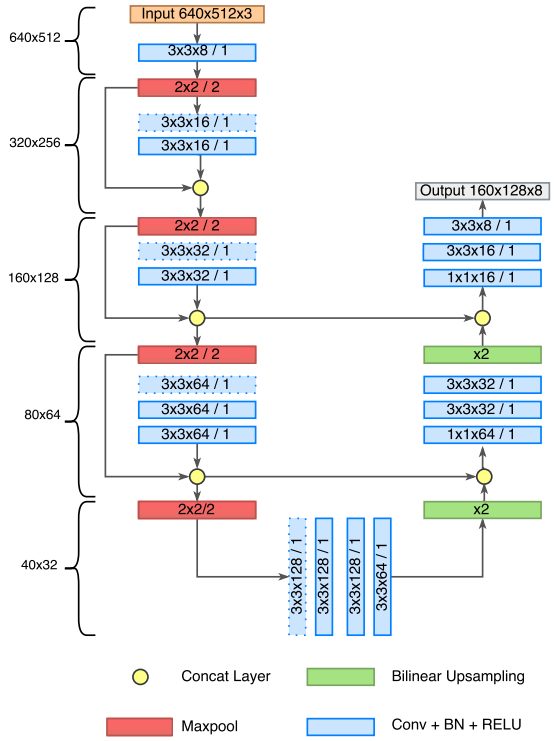

Figure 6. Architecture of SweatyNet-1 and SweatyNet-2. The dotted convolutional layers are removed in SweatyNet-2

the Adam-optimizer [17]. This optimizer has been shown to perform well for deep networks. It automatically adapts its learning rate, so there is no need to manually decrease it in later stages of training. The models are trained end-to-end on the full dataset with a batch-size of 4 and a learning rate of 10^{-3}.

IV. Results

Training the network for 100 epochs from scratch to convergence takes two hours on a single Nvidia GTX-1080-GPU. At that point the recall (RC) on the training set over all classes is around $93\,\%$, while the false detection rate is around $5\,\%$. Recall is defined as true positives (TP) divided by the sum of TP and false negatives (FN)

$$RC = \frac{TP}{TP + FN}. \tag{1}$$

The false detection rate (FDR) is the number of false positives (FP) divided by the number of all detections

$$FDR = \frac{FP}{FP + TP}. \tag{2}$$

A detection is classified as TP if a local maximum with sufficient magnitude is detected within a radius of five pixels around the coordinates of the label. The threshold for a valid detection is determined per class in the testing phase. Defining the threshold as $70\,\%$ of the average magnitude of a maximum over all training data, where at least one object of a class is

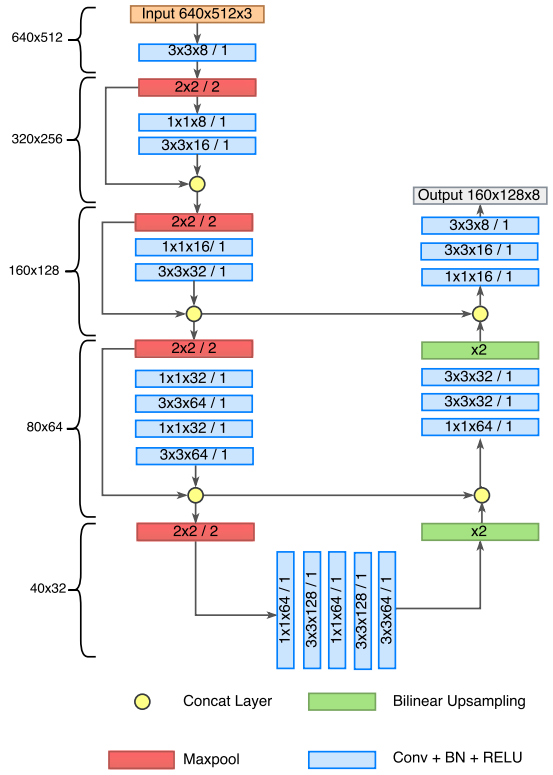

Figure 7. Architecture of SweatyNet-3

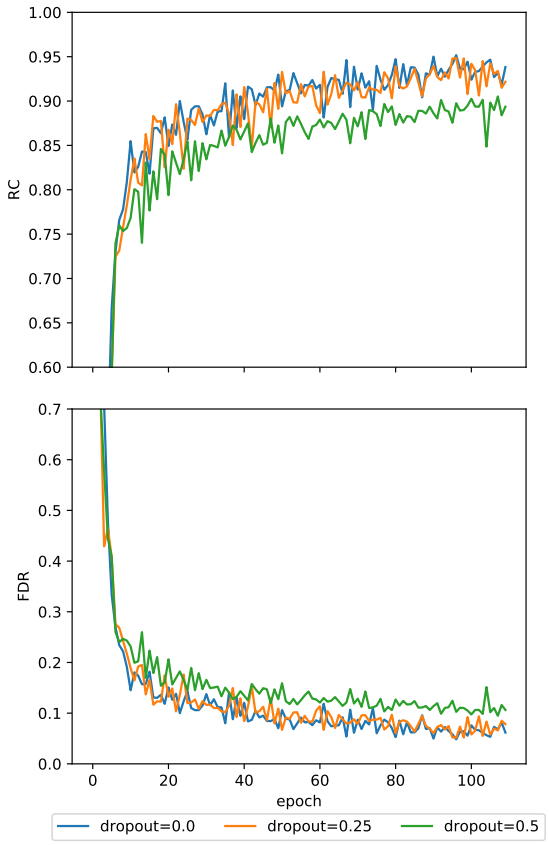

Figure 8. Influence of dropout on recall and false detection rate for SweatyNet-1

present, has shown to be a good trade-off between achieving a high RC and relatively low FDR.

A. Influence of Dropout

Dropout is a widely used method to improve generalization and to prevent over-fitting. A dropout layer randomly zeros a defined percentage of feature maps in the training phase. To examine the effects of dropout a single dropout layer is added at the bottleneck of the network.

Fig. 8 shows the recall and the false detection rate of the SweatyNet-1, trained with dropout probabilities of $p = 0$, $p = 0.25$ and $p = 0.5$. There are no signs of over-fitting without applying dropout; training with $p = 0.25$ leads to comparable RC and slightly higher FDR as with $p = 0$, the network trained with a dropout probability of $p = 0.5$ performs significantly worse.

B. Comparison

The three proposed architectures perform almost equally well at detecting the ball and field entities. SweatyNet-1 performs significantly better at detecting opponents. The complete results are illustrated in Table II. Table I shows a size and speed comparison. GPU and CPU inference time tests were carried out on Sweaty's hardware with an input resolution of 640×512.

C. Performance on Sweaty

Sweaty's vision performed very well at the RoboCup 2017 in Japan trained with only the data collected at our laboratory. Gathering additional training data and retraining on site further improved performance. Running our image processing pipeline in a single thread, frame rates of up to 60 Hz are possible. The network itself needs around 11 ms to process a frame, getting the frame from the camera and preprocessing takes 3 ms. The time it takes to identify maxima and calculate pixel coordinates is dependent on how many objects are detected in the camera image, but is usually around 3 ms for common game situations. Pre- and post-processing are done on the CPU while the network runs on the GPU.

Table I
SPEED AND SIZE COMPARISON

model	parameters	inference GPU	inference CPU
SweatyNet-1	600K	11 ms	69 ms
SweatyNet-2	450K	9 ms	55 ms
SweatyNet-3	300K	9 ms	53 ms

V. Conclusion

It was shown that it is possible to replace Sweaty's old vision tool-chain with separate line and ball detection completely by a single FCNN. The ball, the goal posts and line elements are very reliably detected. The positional resolution needs to be investigated more carefully in order to estimate its

Table II
RECALL AND FALSE DETECTION RATES IN PERCENT AFTER 100 EPOCHS (TOTALS ARE AVERAGES WEIGHTED BY OCCURANCES OF OBJECTS)

model	Balls		Posts		X_Junction		L_Junction		T_Junction		P_Junction		Obstacles		Opponents		Total	
	RC	FDR	RC	FDR	RC	PR	RC	FDR	RC	FDR	RC	FDR	RC	FDR	RC	FDR	RC	FDR
SweatyNet-1	98.7	1.2	**99.1**	**0.8**	**99.2**	**0.7**	94.9	5.1	95.0	5.0	81.5	10.5	97.2	2.8	**84.7**	**15.2**	**94.5**	**5.2**
SweatyNet-1 with dropout $p = 0.5$	**99.2**	**0.8**	**99.1**	**0.8**	97.2	2.8	**95.7**	**4.3**	**98.6**	**1.4**	0.0	100	**98.1**	**2.0**	82.4	17.5	90.5	9.7
SweatyNet-1 with dropout $p = 0.25$	98.7	1.2	98.2	1.7	96.4	3.5	94.8	5.0	97.2	2.8	81.3	18.3	97.2	2.8	75.5	20.1	93.5	5.9
SweatyNet-2	95.3	3.6	96.5	3.4	90.9	6.5	93.6	6.3	94.5	5.4	82.5	12.3	90.0	9.1	69.5	23.2	90.7	8.4
SweatyNet-3	97.0	3.0	96.1	3.1	94.0	5.3	92.1	9.0	94.9	5.1	**92.5**	**7.5**	91.4	8.5	71.3	25.3	91.3	7.3

influence on the localization and attitude of the robot in space, i. e. on the field of play. The detection of opponents and of obstacles (like the referees) is not yet fully satisfactory. In this case a definition by bounding boxes might be better than the current approach.

ACKNOWLEDGMENT

The authors would like to thank Hochschule Offenburg for the financial support of the project, as well as maxon motor GmbH, Becker & Müller GmbH and HOBART GmbH for their sponsorship.

REFERENCES

[1] U. Hochberg, A. Dietsche, and K. Dorer, "Evaporative Cooling of Actuators for Humanoid Robots," in *Proceedings of the 8th Workshop on Humanoid Soccer Robots, IEEE-RAS International Conference on Humanoid Robots*, Atlanta, Oct. 2013.

[2] C. Amoroso, A. Chella, V. Morreale, and P. Storniolo, "A Segmentation System for Soccer Robot Based on Neural Networks," in *RoboCup 1999: Robot Soccer World Cup III*, ser. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, Jul. 1999, pp. 136–147. [Online]. Available: https://link.springer.com/chapter/10.1007/3-540-45327-X_10

[3] U. Kaufmann, G. Mayer, G. Kraetzschmar, and G. Palm, "Visual Robot Detection in RoboCup Using Neural Networks," in *RoboCup 2004: Robot Soccer World Cup VIII*, ser. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, Jun. 2004, pp. 262–273. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-540-32256-6_21

[4] P. R. de Almeida Ribeiro, G. Lopes, and F. Ribeiro, "Neural Network in Computer Vision for RoboCup Middle Size League," *Journal of Software Engineering and Applications*, vol. 09, no. 07, pp. 319–325, 2016. [Online]. Available: http://www.scirp.org/journal/doi.aspx?DOI=10.4236/jsea.2016.97022

[5] D. Albani, A. Youssef, V. Suriani, D. Nardi, and D. D. Bloisi, "A Deep Learning Approach for Object Recognition with NAO Soccer Robots," in *RoboCup 2016: Robot World Cup XX*, Leipzig, Jul. 2016.

[6] M. Javadi, S. M. Azar, S. Azami, S. S. Ghidary, S. Sadeghnejad, and J. Baltes, "Humanoid Robot Detection using Deep Learning: A Speed-Accuracy Tradeoff," in *RoboCup 2017: Robot World Cup XXI*, Jul. 2017.

[7] N. Cruz, K. Lobos-Tsunekawa, and J. Ruiz-del Solar, "Using Convolutional Neural Networks in Robots with Limited Computational Resources: Detecting NAO Robots while Playing Soccer," *arXiv:1706.06702 [cs]*, Jun. 2017, arXiv: 1706.06702. [Online]. Available: http://arxiv.org/abs/1706.06702

[8] S. O'Keeffe and R. Villing, "A Benchmark Data Set and Evaluation of Deep Learning Architectures for Ball Detection in the RoboCup SPL," in *RoboCup 2017: Robot World Cup XXI*, Nagoya, Jul. 2017.

[9] D. Speck, P. Barros, C. Weber, and S. Wermter, "Ball Localization for Robocup Soccer using Convolutional Neural Networks," in *RoboCup 2016: Robot World Cup XX*, Leipzig, Jul. 2016.

[10] D. Speck, "Balltracking for Robocup Soccer using Deep Neural Networks," Hamburg, Jun. 2016, bachelor thesis.

[11] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy, "Speed/accuracy trade-offs for modern convolutional object detectors," *arXiv:1611.10012 [cs]*, Nov. 2016, arXiv: 1611.10012. [Online]. Available: http://arxiv.org/abs/1611.10012

[12] E. Shelhamer, J. Long, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640–651, Apr. 2017.

[13] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *CoRR*, vol. abs/1511.00561, 2015. [Online]. Available: http://arxiv.org/abs/1511.00561

[14] F. Milletari, N. Navab, and S. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," *CoRR*, vol. abs/1606.04797, 2016. [Online]. Available: http://arxiv.org/abs/1606.04797

[15] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *arXiv:1505.04597 [cs]*, May 2015, arXiv: 1505.04597. [Online]. Available: http://arxiv.org/abs/1505.04597

[16] A. P. Aitken, C. Ledig, L. Theis, J. Caballero, Z. Wang, and W. Shi, "Checkerboard artifact free sub-pixel convolution: A note on sub-pixel convolution, resize convolution and convolution resize," *CoRR*, vol. abs/1707.02937, 2017. [Online]. Available: http://arxiv.org/abs/1707.02937

[17] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980