20th RoboCup International Symposium, Leipzig, Germany, July 2016.

A Deep Learning Approach for Object Recognition with NAO Soccer Robots

D. Albani, A. Youssef, V. Suriani, D. Nardi, and D.D. Bloisi

Department of Computer, Control, and Management Engineering,
Sapienza University of Rome,
via Ariosto, 25, 00185 Rome (Italy)
<lastname>@diag.uniroma1.it

Abstract. The use of identical robots in the RoboCup Standard Platform League (SPL) made software development the key aspect to achieve good results in competitions. In particular, the visual detection process is crucial for extracting information about the environment. In this paper, we present a novel approach for object detection and classification based on Convolutional Neural Networks (CNN). The approach is designed to be used by NAO robots and is made of two stages: image region segmentation, for reducing the search space, and Deep Learning, for validation. The proposed method can be easily extended to deal with different objects and adapted to be used in other RoboCup leagues. Quantitative experiments have been conducted on a data set of annotated images captured in real conditions from NAO robots in action. The used data set is made available for the community.

Keywords: Robot Vision, Deep Learning, RoboCup SPL, NAO Robots

1 Introduction

Starting from 2008, the RoboCup Standard Platform League (SPL) involves Aldebaran NAO as the common robot for the competitions. Due to the limited computational power available, Computer Vision techniques adopted by the different teams are mostly based on color segmentation approaches. The success of those solutions has been facilitated by special expedients, for example the use of a red ball, controlled illumination, and yellow coloured goals. However, the current trend in using more and more realistic game fields, with white goal posts, natural light, and a ball with black and white patches, as well as personalized jersey shirts, imposes the adoption of robust detection and classification methods, which can deal with a more realistic and complex environment.

In this paper, we propose a method for validating the results provided by color segmentation approaches. In particular, we present a novel algorithm for merging an adaptive segmentation procedure with a Deep Learning based validation stage, which exploits Convolutional Neural Networks (CNN). Our approach is designed to work in scenes with changes in the lighting conditions that can affect significantly the robot perception. The segmentation procedure follows a color based approach with a mechanism for automatically adapting the exposure and white balance parameters of the camera. The validation step consists in a supervised image classification stage based on CNN.

The main contributions of this paper are: i) A dynamic white balance and exposure regularization procedure; ii) A Deep Learning based validation step; iii) A novel fully annotated data set, containing images from multiple game fields captured in real conditions. We present quantitative results obtained with different network architectures, from the simplest one (three layers) to more complex ones (five layers). To test and train our networks, we have created a specific data set, containing 6,843 images, coming from the upper camera of different NAO robots in action. The data set is fully annotated and publicly available.

The rest of the paper is organized as follows. Section 2 presents an overview of the object detection methods presented during the last years of Robocup competition together with a survey of the recent Deep Learning classification approaches. Section 3 contains the details of our method. The data set for training and testing is described in Section 4, while the quantitative experimental results are reported in Section 5. Conclusions are drawn in Section 6.

2 Related Work

The RoboCup 2050 challenge consists in creating a team of fully autonomous humanoid robot soccer players able to win a real soccer game against the winner of the FIFA World Cup. To deal with such a difficult goal, it is necessary to develop a vision system that is able to provide all the environmental information necessary to play soccer on a human level [3]. This means that: 1) a pure color based approach cannot be a feasible solution for the RoboCup 2050 challenge and 2) solutions able to provide articulated information about the scene are needed. In particular, the presence of different views and varying color information produced by the camera of a moving robot makes recognition a hard task, which cannot be solved with *ad hoc* solutions.

In this section, we briefly discuss the specific techniques currently used in the RobocCup SPL and then we provide an overview of more general, recent Deep Learning methods developed in the Computer Vision field.

The method described in [6] exploits color similarities for ball and goal recognition. The recognition routine is invariant to illumination changes and it is not computationally expensive, making it suitable for real-time applications on NAO robots. The detection of the objects (lines, ball, penalty marks, and goal posts) is obtained by merging geometrical (line detection and curve fitting [13]) and color information [12,17]. In addition, the ball motion model is combined with contextual information [3] to enhance the performance of the ball tracker and to carry out event understanding.

In the attempt of developing more accurate human-like vision system, recent approaches tend to work on the middle level of features, where the low level features are aggregated in order to increase scene understanding [1,10]. Convolutional Neural Network (CNN), as a hierarchical model based on low level features, has showed impressive performance on image classification [9], object detection [2], and pedestrian detection [11]. In the group of CNN based methods, DetectorNets [16] and OverFeat [14] perform the object detection on a coarse set of sliding-windows. On the other hand, R-CNN [4,5] works on the extraction of proposal regions, over the entire image at multiple scales, to be classified using a support vector machine (SVM) trained on CNN features.

Recognition approaches use machine learning methods based on powerful models to address the large number of object categories present in general scenes. Deep Models outperformed hand-engineering features representation in many domains [8]. For example, a Deep Learning model based on large and depth network (5 convolution layers and 3 fully connected ones) described in [9] performed image classification on the ImageNet ILSVRC-2010 data set better than the previous state-of-the-art methods. Moreover, single object detection based on deep Convolutional Neural Network [2] demonstrate to be more effective on ILSVRC-2012 and DeepMultiBox data sets with respect to previous object localization approach, thanks to the capacity of handling the presence of multiple instances of the same object in the processed image.

Here, our aim is to combine fast color segmentation techniques already in use in the SPL with more computationally demanding classification methods, in order to use CNN for NAO robot detection, even in presence of a limited hardware. We believe that the adoption of Deep Learning techniques adds an additional level of robustness to the NAO vision system and can lead to the use of context information for higher-level computation, e.g., behaviours.

3 Proposed Approach

Our functional architecture is shown in Fig. 1, where the CNN block is placed at the end. In our case, the use of CNN for classification purposes is important to obtain a validation of the extracted image regions, carried out in the initial part of the pipeline. It is worth noting that, even if Deep Learning procedures spread across various research fields thanks to the release of powerful and cheap hardware (e.g., GPUs for gaming), running those procedures on less advanced

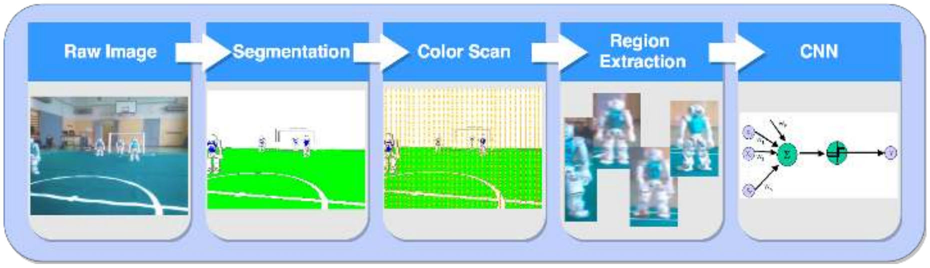


Fig. 1: Our pipeline for the Deep Learning NAO detection module.

hardware (e.g., the ATOM CPU offered by the NAO platform) can overwhelm the available computation power. As a possible approach for Deep Learning techniques to work on NAOs, we propose to have a pre-processing step that allow to reduce the amount of information to be tested by the CNN. In other words, we want to reduce the possible search space. This reduction is provided by the Segmentation, Color Scan, and Region Extraction functions in the pipeline.

During the last few years, different papers proposed solutions for using Deep Learning also for object detection without being able to merge it with a classification step [2,16]. Our method adopts the same approach, i.e., detection followed by classification, with the difference that the detection process does not involve Deep Learning techniques.

Segmentation. The input for the pipeline is an RGB image acquired by the camera of the robot at each cycle. The image is converted to the HSI (Hue, Saturation, and Intensity) color space and is processed to extract regions according to their color. In particular, all the regions that fall inside a given range on the Hue channel are replaced by the corresponding color (see step 2 in Fig. 1). A dynamic update of the white balance and the exposure is used to cope with possible illumination changes (e.g., light from windows) and to increase the robustness to lightning changes. To keep the acquisition rate of the camera consistent with real-time applications, the update of the camera settings is limited to once per second.

The adaptive camera setting update procedure is initialized with values calculated on contextual information. For the white balance, our approach uses the *green world assumption* (Assumption 1) to set the initial values.

Assumption 1. *Given an input image, it is possible to establish an horizon line (knowing the robot structure) and to segment the part of the image placed below the horizon according to a neighbours color clustering procedure. The biggest segmented regions is green-coloured.*

Nevertheless, there are cases in which the above assumption does not hold (e.g., the robot is looking outside the field border or another robot is standing close to the camera). To deal with possible inconsistencies, we use a threshold that allows for discarding images that present large variations in the Hue values with respect to the previously received images. Eq. 1 illustrates the details of the white balance update procedure, where $max(.)$ and $min(.)$ represent the pixels with higher and lower Hue values, respectively.

$$wb_{t1} = (\frac{max(H^n_{t0}) - min(H^n_{t1})}{n}) \cdot 100 \quad (1)$$

H^n_{ti} is the average value in the set H of n pixels coming from the images captured at time ti. Thus, wb_{t1} represents the variation in percentage over the Hue channel of the green region extracted according to Assumption 1.

Algorithm 1 shows the validation procedure for the pixels belonging to the set H, while Fig. 2 contains an example where the region selection process is applied for white balancing. The procedure for updating the camera exposure is

Algorithm 1 Validation

```
procedure VALIDATEREGIONS($Regions$, $H^{t_{i-1}}_{average}$)
    $H_{low} \leftarrow 60°$                                          ▷ high boundary for green
    $H_{high} \leftarrow 120°$                                        ▷ low boundary for green
    for all $r$ in $Regions$ do
        $H^{r,t_i}_{avg} \leftarrow average(r)$
        if $(H^{t_{i-1}}_{avg} - H^{r,t_i}_{avg}) < threshold$ then
            $append(r, Accepted)$
            $H^{t_i}_{avg} \leftarrow H^{t_i}_{avg} + H^{r,t_i}_{avg}$        ▷ update $H_{avg}$ for comparisons at time $t_{i+1}$
        end if
    end for
    return $Accepted$, $H^{t_i}_{avg}$
end procedure
```

carried out by sampling green pixels from the image below the robot horizon line (thus avoiding possible outliers or strong sources of illumination). A mask with fixed weights (see Fig. 3) is used to compare samples from each region and to compute an overall value that is then tested against a predefined threshold. Such threshold is based on samples collected every 4 seconds and stored in memory. If a significant variation in the intensity of the current pixels with respect to the average of the stored ones is detected, then the camera exposure parameter is re-calculated accordingly.

Color Scan. Once the image has been corrected for possible changes in the illumination conditions, the next step is to scan it for detecting color changes. Our approach is based on the work presented in [13]. Due to the introduction of the realistic ball (a ball with black and white patches) in the SPL matches, the focus is on white-coloured areas: Regions with a sufficient number (i.e., greater than a fixed threshold) of white pixels are grouped together.

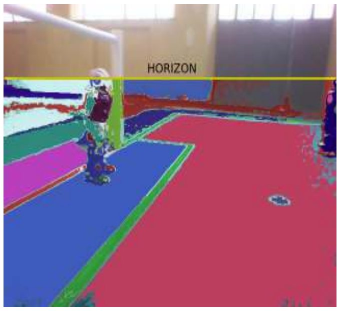
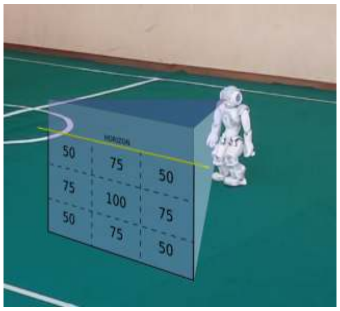


Fig. 2: Region selection process for white balance compensation.



50	75	50
75	100	75
50	75	50

Fig. 3: Adaptive exposure compensation according to the robot's interest areas.

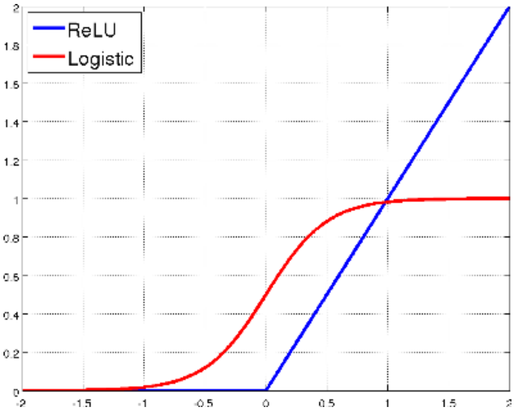


Fig. 4: The ReLU $f(x) = max(0, x)$ and the Logistic function $f(x) = \frac{1}{1+\exp(-x)}$. Thanks to [18] for the original image.

Region Extraction. The image of a NAO robot in standing position can be enclosed in a rectangular bounding box with a larger height with respect to the width, while the image of a ball can be enclosed in a square. Therefore, regions are extracted and fed to a pre-trained Convolutional Neural Network.

CNN. Different CNNs have been tested to obtain a good trade-off between accuracy and computational load. A first CNN presents three layers, being the smallest and thus the fastest. Other networks reach up to five layers, as presented in [9]. Although inner parameters may change, all the convolutional layers present the same inner structure: Convolutional, Pooling, and Normalization layers. For limiting the computational needs, we have decided to use a single-scale CNN and to avoid more complex structures, such as multi-scale CNN.

As suggested by Zeiler et al. in [18], to allow non-linearity for the CNNs, we use a rectified linear unit (ReLU) as shown in Fig. 4. The ReLU replaces the traditional neuron's activation function, it is not affected by the gradient vanishing problem (as for sigmoid and tanh function) and it offers high efficiency even without pre-training. To implement the networks, we have used the open source framework TensorFlow[1], recently released by Google Brain. TensorFlow is a Deep Learning framework similar to other frameworks like Caffe or Torch.

4 Dataset Description

To properly train the recognition network, a big amount of data from a real scenario is needed. The creation of a data set is not trivial, since it may contain images taken in varying conditions, and it requires accurate ground-truth annotations. Due to the lack of a public data set concerning the RoboCup SPL environment, we have decided to collect and share a set of images taken from

[1] `tensorflow.org`

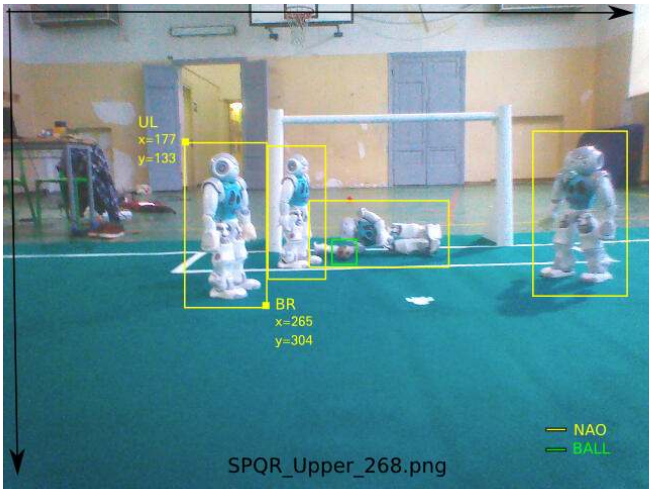


Fig. 5: One of the annotated images from the SPQR NAO image data set.

different real fields, called the **SPQR NAO image data set**. We strongly believe that the introduction of supervised techniques in robotic platforms like NAOs can lead to great improvements also in scientific fields that lie outside the soccer competitions. To encourage the development of similar approaches from other authors, we have decided to made the data set publicly available at: `www.diag.uniroma1.it/~labrococo/?q=node/6`.

The SPQR NAO image data set is built over images captured by different players (i.e., NAO robots) in varying environments. Indeed, the images have been captured during tests made on fields subject to natural and artificial illumination at the same time, thus different areas of the scene may be subject to high contrast and differences in brightness. A particular mention goes to the Multi-Sensor Interactive Group from the University of Bremen, which provided a part of the images in the data set.

All the images in the data set are annotated and have been processed accordingly to a pre-sampling phase, where images that are temporally adjacent have been discarded. Three different classes, namely *ball*, *nao*, and *goal*, have been considered for the ground-truth data. All the annotations are provided in a text file named ”annotations.txt” that contains one annotation per line. Annotations entries are divided into *ImageName*, *UpperRightCorner*, *BottomLeftCorner*, *Class*. *ImageName* is the full image name, including the extension. The *Class* information is set accordingly to the three above listed classes. The remaining two parameters are the image coordinates of the upper-left and bottom-right corners of the bounding box that wraps the corresponding object. The origin of the image coordinates is placed in the upper-left corner. Since more than one object is usually present in an image, bounding boxes could overlap. Fig. 5 shows an example of the annotations available.

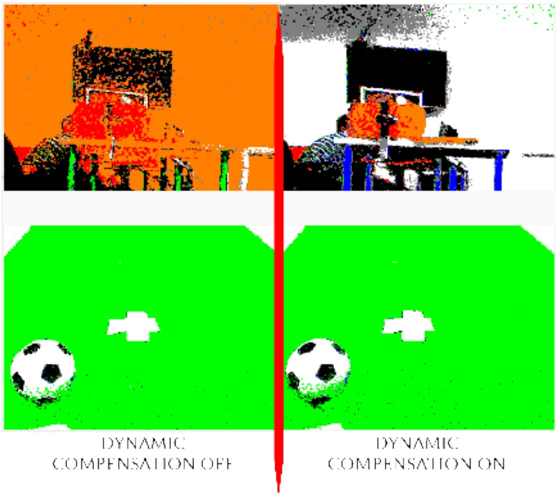


Fig. 6: Dynamic exposure and white balance results.

5 Experimental Results

This section contains the quantitative experimental results obtained by our method. It is worth noting that, due to the lack of publicly available data sets containing images captured by NAO in real SPL matches, we were not able to compare our results with other methods. For such a reason, we have decided to make publicly available our data set in order to provide other authors a way for comparing their results with ours.

During a typical SPL match, the robots can capture a number of different objects (both in the field and from outside). Moreover, in addition to the other NAOs, also humans may be present on the field (e.g., the referees and the audience). Thus, we trained our networks to generate a binary classification between the two classes "NAO robot" and "not-NAO robot". It is worth noting that, for this task, CNN based approaches are very powerful and, as shown in rest of this section, optimal results have been achieved. However, the computational cost required by a CNN based approach can be a problem. The aim of this work is to investigate the possibility of reducing the computational requirements to be compatible with the limited hardware of the NAO robots.

5.1 Dynamic Exposure and White Balance

The initial manual camera calibration constitutes a very restrictive limit for any robotic applications. The NAO vision system is not very robust to illumination variations and the two auto-white balance and auto-exposure available options can drastically influence the performance. The above considerations led us to consider the environment lighting conditions as one of the most influencing factor during a match.

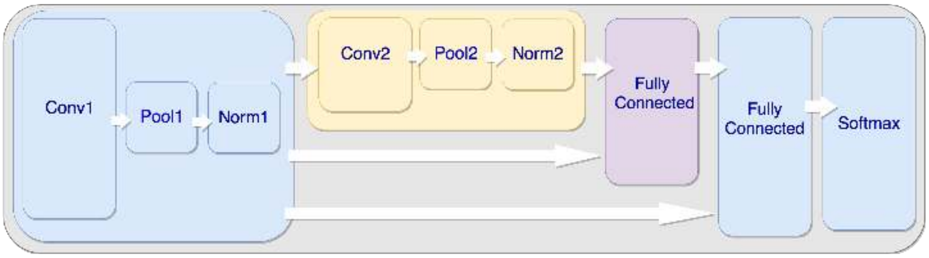


Fig. 7: Overview of the three proposed architectures.

Changes in the camera settings requires a non-trivial recovery of the stable conditions. Fig. 6 shows the dynamic compensation of the camera settings as previously explained. The raw and the segmented image in the first column do not present any dynamic update, while those in the second column do. The comparison between the two is straightforward and, when enabled, the dynamic procedure allows to continuously adapt the parameters to environmental changes, still maintaining real-time performance (25 frames per second).

5.2 CNN Training and Evaluation

We trained our classifier on the SPQR NAO image data set, where original annotations have been subject to the data augmentation process proposed by Sermanet and LeCun in [15]. 24,236 overlapping crops are extracted and labelled with the *nao* class. More than a single crop could derive from an annotations; such crops present at max 0.5 Jaccard overlap similarity as shown in Eq. 2, where A and B are defined as different crops.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{2}$$

37,156 negatives are then extracted from the remaining part of the image.

Evaluation has been made by using an independent and non augmented subset of 1000 images coming from the SPQR NAO image data set.

5.3 CNN Results

We have tested three different architectures as shown in Fig. 7, starting from a three inner-layers networks up to a five inner-layers. In all of the proposed networks, the learning rate present a decay factor and the Adagrad online optimization has been applied [7]. The same kernel $[5,5]$ has been used for every convolutional layer; dropout is not present. The output of the first convolutional layer present a depth of 64, the second and the third, if present, respectively 128 and 384. We use $SAME$ padding with a stride of $[1,1,1,1]$ for the convolutional layer and $[1,3,3,1]$ for the pooling.

Table 1: Quantitative results on the SPQR NAO image data set. Results are expressed in terms of accuracy and frames per second (FPS)

Architecture	Accuracy (%)	FPS
3-Layers	100	14-22
4-Layers	100	13-20
5-Layers	100	11-19

First, we present quantitative results for the three tested networks. These are computed over the SPQR data set and reported in Table 1. Meaningful results about the execution time come from tests ran on the NAO platform. Due to the lack of computational power, all the process not related to the proposed work were disabled during this phase. Last column of Table 1 reports the results for the number of images processed per second over the NAO. The time spent for the evaluation of a single image via the relative network is approximately 7 seconds if the whole network has to be initialized.

Results from Table 1 shows a perfect accuracy on all the tested networks. Such results do not come unexpected because of the simplicity of the classification task and of its domain of application (see Fig. 8). On the other hand, the analysis of the computational load results show margin for real-time applications.

6 Conclusions

We have presented a Deep Learning method for NAO detection to be used in the RoboCup Standard Platform League. In particular, our pipeline contains two main stages: 1) a pre-processing step to segment the image, in order to reduce the

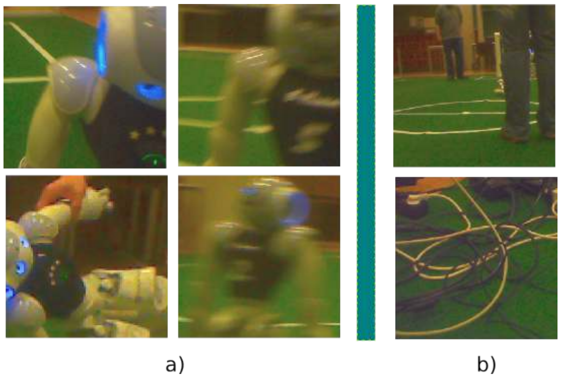


Fig. 8: Examples of the samples used for testing. a) Four correctly classified positives labelled as NAO class. b) Two correctly classified negatives. Images courtesy of the B-Human Team.

search space; 2) a validation phase based on the Convolutional Neural Networks (CNN), which is useful to confirm that the extracted regions actually contains objects of interest.

An important contribution of this work is the creation of a novel data set, called SPQR NAO image data set. The data set, containing images captured from NAOs in action on a regular field, is fully annotated and publicly available.

The proposed approach performs very well under a pure accuracy point of view. However, applicability problems arise when considering the computational time even with a binary classification. Even if the NAO platform is limited in computational power and not feasible for reaching a high frame rate, we believe that Deep Learning techniques have to be considered to deal with the Robocup 2050 challenge. This is also true for RoboCup leagues with the possibility of exploiting greater computational power.

We are currently working on using simpler networks outside the existing frameworks that may reduce the overall execution time. Moreover, we are planning to extend the classification task to more classes and to be able to classify also other objects typical of this environment. As future work, we intend to develop a complete object detection procedure based on custom CNN.

7 Acknowledgment

We wish to acknowledge the Multi-Sensor Interactive Systems Group Faculty 3 - Mathematics and Computer Science University of Bremen for providing a big part of the images used in the SPQR NAO image data set.

References

1. N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition*, volume 1, pages 886–893 vol. 1, 2005.
2. D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. Scalable object detection using deep neural networks. In *Computer Vision and Pattern Recognition*, pages 2155–2162, 2014.
3. U. Frese, T. Laue, O. Birbach, and T. Röfer. (A) vision for 2050 - context-based image understanding for a human-robot soccer match. *ECEASST*, 62, 2013.
4. R. Girshick. Fast R-CNN. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2015.
5. R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
6. A. Härtl, U. Visser, and T. Röfer. *RoboCup 2013: Robot World Cup XVII*, chapter Robust and Efficient Object Recognition for a Humanoid Soccer Robot, pages 396–407. Springer Berlin Heidelberg, 2014.
7. Duchi J., Hazan E., and Singer Y. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159, 2011.

8. Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22Nd ACM International Conference on Multimedia*, MM '14, pages 675–678, 2014.
9. A. Krizhevsky, I. Sutskever, and G.E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1106–1114. 2012.
10. R. Lienhart, A. Kuranov, and V. Pisarevsky. *Pattern Recognition: 25th DAGM Symposium*, chapter Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection, pages 297–304. Springer Berlin Heidelberg, 2003.
11. W. Ouyang and X. Wang. Joint deep learning for pedestrian detection. In *Computer Vision (ICCV)*, pages 2056–2063, 2013.
12. T. Röfer. *RoboCup 2007: Robot Soccer World Cup XI*, chapter Region-Based Segmentation with Ambiguous Color Classes and 2-D Motion Compensation, pages 369–376. Springer Berlin Heidelberg, 2008.
13. T. Röfer, T. Laue, J. Richter-Klug, M. Schünemann, J. Stiensmeier, A. Stolpmann, A. Stöwing, and F. Thielke. B-Human team report and code release 2015, 2015. Only available online: `http://www.b-human.de/downloads/publications/2015/CodeRelease2015.pdf`.
14. P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *CoRR*, abs/1312.6229, 2013.
15. P. Sermanet and Y. LeCun. Traffic sign recognition with multi-scale convolutional networks. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 2809–2813, 2011.
16. C. Szegedy, A. Toshev, and D. Erhan. Deep neural networks for object detection. In *Advances in Neural Information Processing Systems 26*, pages 2553–2561. 2013.
17. S. Volioti and M.G. Lagoudakis. *Artificial Intelligence: Theories, Models and Applications: 5th Hellenic Conference on AI, SETN 2008, Syros, Greece, October 2-4, 2008. Proceedings*, chapter Histogram-Based Visual Object Recognition for the 2007 Four-Legged RoboCup League, pages 313–326. Springer Berlin Heidelberg, 2008.
18. M.D. Zeiler, M. Ranzato, R. Monga, M.Z. Mao, K. Yang, Q.V. Le, P. Nguyen, A.W. Senior, V. Vanhoucke, J. Dean, and G.E. Hinton. On rectified linear units for speech processing. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, pages 3517–3521, 2013.