Autonomous Robots (2021) 45:335–350
https://doi.org/10.1007/s10514-021-09966-9

Deep learning applied to humanoid soccer robotics: playing without using any color information

Nicolás Cruz[1] · Francisco Leiva[2] · Javier Ruiz-del-Solar[2]

Received: 13 February 2020 / Accepted: 5 January 2021 / Published online: 30 January 2021

Abstract
The goal of this paper is to describe a vision system for humanoid robot soccer players that does not use any color information, and whose object detectors are based on the use of convolutional neural networks. The main features of this system are the following: (i) real-time operation in computationally constrained humanoid robots, and (ii) the ability to detect the ball, the pose of the robot players, as well as the goals, lines and other key field features robustly. The proposed vision system is validated in the RoboCup Standard Platform League, where humanoid NAO robots are used. Tests are carried out under realistic and highly demanding game conditions, where very high performance is obtained: a robot detection accuracy of 94.90%, a ball detection accuracy of 97.10%, and a correct determination of the robot orientation 99.88% of the times when the observed robot is static, and 95.52% when the robot is moving.

Keywords Soccer robotics · Deep learning · Convolutional neural networks · Robot detection · Ball detection · Robot orientation determination

1 Introduction

Robotic soccer promotes robotics and artificial intelligence research by offering a formidable challenge: "By the middle of the twenty-first century, a team of fully autonomous humanoid robot soccer players shall win a soccer game, complying with the official rules of FIFA, against the winner of the most recent World Cup" (RoboCup 2020a). Soccer is a real-time, distributed decision-making problem, where players need to perceive and understand the environment, make collective decisions, and execute these decisions with the final objective of winning the match; i.e. scoring goals against the opponent team and avoiding goals from it.

The perception of the environment is one of the key abilities for playing soccer; without an adequate vision system it is not possible to determine robustly the position of the ball and the pose of the other players, to identify key field features (e.g. goals and field lines) and to self-localize, which are essential abilities to play properly. Given that the soccer environment has a predefined physical setup, and that robots used in RoboCup soccer leagues normally have limited processing capabilities, most of the current vision systems used in soccer robotics are based on the use of color information. However, the use of color information has some drawbacks such as (i) the need for calibration of the camera and tuning of the color-segmentation' parameters to achieve a properly color segmented image and/or the calibration of perception algorithms employed due to the fact that color perception depends on the environmental illumination, and (ii) the need of a soccer field with predefined colors (e.g. lines need to be white, field/carpet needs to be green).

Currently, there are different robot soccer leagues, which use different kinds of real or simulated mobile robots (RoboCup 2020a). In this work we are interested in playing soccer with real humanoid robots. We choose to work in the RoboCup Standard Platform League (SPL) given that it uses a standard platform, the NAO humanoid robot (RoboCup 2020b), which allows to compare and share developments with other teams, and to focus on the cognitive aspects of the problem.

The RoboCup SPL started in 2008, and the first vision systems used by the competing teams were based on those developed in the former Four-Legged League, which used

✉ Javier Ruiz-del-Solar
jruizd@ing.uchile.cl

[1] Department of Electrical Engineering, Universidad de Chile, Santiago, Chile

[2] Department of Electrical Engineering, Advanced Mining Technology Center, Universidad de Chile, Santiago, Chile

SONY-AIBO four-legged robots as its standard platform. In both leagues, the first generation of vision systems was based on the segmentation, detection, and analysis of colored objects of interest: the ball, lines, beacons, goals, players and the field/carpet. Year by year, the restriction of having colored objects in the field was relaxed: (i) the number of colored beacons (used for the robot's self-localization) was first reduced from six to four, then to two, and then beacons were removed in 2008, (ii) the goals were first colored and solid, then composed by three colored cylinders (goalposts and crossbar) and a white net, and finally composed by three white cylinders (goalposts and crossbar) and a white, gray or black net (since 2015), (iii) the ball used to be orange, and since 2016, black and white. However, still most of the teams use color information to detect some field features (the lines, goal posts and penalty marks), the players, and the ball.

Recently, Convolutional Neural Networks (CNNs) have been used for detecting the robots and/or the ball (Albani et al. 2017; Speck et al. 2017, 2019; Cruz et al. 2018; Javadi et al. 2018; Menashe et al. 2018; Gabel et al. 2019; Felbinger et al. 2019; Kukleva et al. 2019; Poppinga and Laue 2019; Teimouri et al. 2019). Most of these CNN-based detectors require object proposals, which are currently obtained using color information of the field/carpet (green) and the lines (white). There have also been efforts to use end-to-end trained CNNs to detect all field's objects without relying on object proposals (Szemenyei and Estivill-Castro 2019a, b). However, considering the limited processing capabilities of the NAO's CPUs (the NAO v4 and v5 models are powered with an ATOM Z530 1.6 GHz CPU), these vision systems are still unable to run in real-time while playing soccer, and, at the same time, to obtain the required performance for highly competitive matches.

Therefore, to the best of our knowledge, color-free vision systems have not been used in real robot soccer games, at least not in the SPL. Some of the main reasons underlying this are the following: (i) the challenge of achieving real-time operation when using limited computational resources, (ii) the problem of training deep detectors without having very large databases, which are difficult to create when real-world soccer conditions are taken into account, and (iii) the challenge of developing efficient and reliable color-free object proposal generators.

We believe that using color-free vision systems in soccer robotics is relevant, because this eliminates the constraint of having objects on the field with specific colors (e.g. the lines), and because it eliminates the need for calibration of the vision systems (before and/or during the games), making it possible to play soccer under variable lightning conditions (e.g. indoors near big windows or outdoors).

The goal of this paper is to propose a color-free vision system for humanoid soccer robotics, which will be validated in the SPL. The main features of this system are (i) real-time operation in humanoid robots (specifically in the NAO v5 robots that are part of the official platform for the SPL), and (ii) the ability to detect the ball position, the robots' pose, the lines, and key field features very robustly. In fact, as it will be shown in Sect. 5, the proposed ball, robots and robots' orientation detectors are highly performant; they achieve very high detection rates, measured under realistic RoboCup SPL game conditions.

To the best of our knowledge, the proposed system is the first color-free vision system for humanoid soccer robotics that is able to run in real-time, with a performance that allows its use in robotics world championships. It is very important to stress this point, because in images acquired under real-world conditions, the objects are much difficult to detect than in standard databases. For instance, ball perception is prone to have image blurring produced by the fast movement of the ball and the unstable walking of the robots. The proposed vision system was used by our team, UChileRT, in the RoboCup 2018 Word Competition, and its robot detector in the RoboCup 2017 Word Competition.

The main technical contributions of this paper are the following: (i) the proposal of a vision framework that combines concepts of deep learning and cascade classification to obtain, at the same time, high detection rates and fast processing, (ii) the use of a training methodology based on bootstrap and active learning, and (iii) the proposal of a method that is able to accurately determine the orientation of an opponent humanoid robot player by using a combination of heuristics and CNNs.

A preliminary version of this work was presented in Leiva et al. (2019). In this extended version a much deeper explanation of the proposed color-free vision system and its main modules is provided, as well as a better description of the design and training of the CNN-based detectors. The structure of all of the proposed CNN detectors is explicitly described, as well as new detection results in real soccer fields. In addition, in this extended version the proposed framework is also validated in a different domain (detection of human soccer players in thermal images) to show its applicability beyond soccer robotics.

The paper is organized as follows: related work is presented in Sect. 2; the proposed color-free vision system is described in Sect. 3. Section 4 describes the design and training of the proposed CNN-based detectors. The experimental validation and results are shown in Sect. 5; and finally, conclusions and suggestions for future work are presented in Sect. 6.

2 Related work

Since 2016, CNNs have been used for detecting the robots and/or the ball in the SPL and humanoid RoboCup leagues

(Albani et al. 2017; Speck et al. 2017, 2019; Cruz et al. 2018; Javadi et al. 2018; Menashe et al. 2018; Gabel et al. 2019; Felbinger et al. 2019; Kukleva et al. 2019; Poppinga and Laue 2019; Teimouri et al. 2019).

In Albani et al. (2017), the first CNN-based robot detector for the SPL league was proposed. In this system, robot proposals are first computed by using color-segmentation based techniques, and then, a CNN is used for validating the robot detections. Different architectures with three, four, and five layers are explored. In the reported experiments, the 5-layer architecture was able to obtain 100% accuracy in the SPQR NAO image data set, also proposed in Albani et al. (2017). However, evaluating a detector using this dataset is different from evaluating it in real game conditions, which have much harder requirements. The detector was able to run at 11–19 fps on a NAO robot when all non-related processes (such as self-localization, decision-making, and body control) were disabled. Because of the latter, this detector could not be used to play soccer in real soccer games.

In Javadi et al. (2018), the performance of three well-known CNN architectures (namely LeNet, GoogLeNet, and SqueezeNet) was analyzed in the task of detecting humanoid robots. In this study, however, no real-world deployment was presented.

In Poppinga and Laue (2019), a proposal-free robot detector based on CNNs was presented. The proposed network has an adaptable architecture, it is multi-scale, and uses separable convolutional blocks (Howard et al. 2017). The authors also proposed a novel training procedure inspired in the generator-discriminator adversarial learning paradigm, which allowed training the networks using real and simulated images at the same time. The trained detectors were able to detect robots under realistic conditions, and the obtained detection time was 9.0 ms for a single image. In case that a similar approach is used to detect other objects (e.g. the ball), it is not clear that those detectors would be able to run simultaneously in real-time.

In Cruz et al. (2018), we presented a CNN-based robot detector, capable of operating in real-time. The system was based on the classification of color-based robot proposals generated by B-Human's robot perceptor (Röfer et al. 2017). This was modeled as a binary classification problem, where proposals could be labeled as robots or non-robots. The system processed robot proposals in $\sim$1 ms while playing soccer, with an average accuracy of $\sim$97%. Although this detector achieved a very high performance, it possessed some drawbacks. While the CNN classifier was robust to noise and variations of the illumination, the same did not apply to the color-based robot proposal generator. Adverse environmental conditions could lead the algorithm to produce an excessive amount of object hypotheses, or none at all. The second drawback derived from the CNN $\sim$1 ms inference time. While such a network is deployable on a NAO robot, it is much slower than alternative algorithms based on heuristics or shallow classifiers, and can be prohibitively slow when too many robot proposals are generated. In this paper we address both problems by changing the robot proposals generation approach, and by further reducing the inference times while maintaining the detection accuracy.

In Speck et al. (2017), the first CNN-based ball detector for the RoboCup humanoid league was proposed. The detector used two CNNs, which were able to obtain a localization probability distribution for the ball over the horizontal and vertical image axes, respectively. Several non-linearities were tested, with the soft-sign activation function generating the best results. Processing times in the robot platforms were not reported in that work, and the obtained accuracy was about 80%.

In Teimouri et al. (2019), a CNN-based ball detector for the humanoid league was presented. The proposed architecture is multi-scale and uses separable convolutional blocks (Howard et al. 2017). The detector is not color-free, because the ball proposals are generated considering the white and green patterns of the soccer field. The obtained performance of the detector is 70.9%, and it decreases with variable lighting conditions and blurred images.

In Menashe et al. (2018), ball detection using different machine learning methods is analyzed. The system considers several heuristic stages used for generating the ball proposals, and a final classification stage implemented using either a SVM or a CNN based classifier. The performance of both systems is analyzed, but the analysis was focused on the transferability between different soccer environments.

In Felbinger et al. (2019), a genetic design approach for optimizing the hyper-parameters of a CNN designed to detect the ball is presented. The focus is not on the real-world deployment, but on the genetic based design of the network. Nevertheless, an average runtime of 8 ms was obtained in the NAO robots.

Some other authors have proposed CNN based ball detectors that requires a GPU for running in real-time (Gabel et al. 2019; Speck et al. 2019; Kukleva et al. 2019). Obviously, these detectors cannot be used in robots that just rely on CPU-based processing (such as the NAO robots).

In a different research line, some authors have proposed end-to-end trained CNNs to detect all field's objects without using object proposals (Szemenyei and Estivill-Castro 2019a, b). In Szemenyei and Estivill-Castro (2019a), the use of two networks, one to perform semantic segmentation of the images, and a second one to propagate class labels between consecutive frames, is proposed. Authors reported that the fully neural vision pipeline runs at 6 frames per second, which from our point of view is not enough for playing soccer at a competitive level. In Szemenyei and Estivill-Castro (2019b), ROBO, a new CNN model inspired in the popular Tiny YOLO (Redmon and Farhadi 2017), is proposed. ROBO

is able to detect all relevant objects in the soccer field. The processing time of the different versions of the CNN, which consider different levels of pruning, range from 2.3 frames per second to 13 frames per second, which obtains a Mean Average Precision (mAP) of about 83%.

We believe that the limited processing capabilities of humanoid robots currently used in robotic soccer, are not sufficient to use end-to-end trained CNNs to reliably detect all field objects in real time while playing soccer.

3 Playing soccer without color information

In this section we describe the proposed vision system. Section 3.1 broadly explains the general characteristics and functioning of the vision framework, while Sects. 3.2–3.9 describe the operation of each of its main modules.

3.1 The general framework

As already mentioned, the main feature of the proposed vision system is that it manages to detect the ball, the robot players, their orientations, and key features of the field without using any color information, i.e. the whole processing is performed using grayscale images rather than on a color segmented image. Removing the color segmentation step from the pipeline offers several advantages such as reduced operation times, reduced points of failure for the vision modules, easier pre-game calibration, and larger range of valid camera parameters since our object and field feature detectors are more resilient to changes in illumination than color-based approaches.

The key design components that allow the robust detection of all these objects in real-time (using robots with processing limitations) are the following: (i) custom-made object proposals generators for each kind of object, which are based on the characteristics of the soccer problem, (ii) CNN-based object detectors using a light CNN architecture specially designed for this application (Cruz et al. 2018), (iii) a cascade classification methodology inspired in Viola and Jones (2001), which implements the detection of some objects (e.g., the ball) using a two-stage classification cascade of CNN-based detectors, where the first stage discards, very quickly, non-objects that are very different from the objects being detected, and the second stage performs the final classification, and (iv) the use of the detection results of some object detectors for constraining the search of the others. In summary, we follow a pragmatic approach that combines classical algorithms widely used in robot vision with modern CNN-based classifiers.

The proposed vision framework is illustrated in Fig. 1. While the detection of lines and field features is done by using a set of rules and heuristics commonly employed in the SPL community (modules in yellow), the detection of the ball, the robot players and their orientation is done by means of object proposals (modules in green) and their subsequent classification using CNNs (modules in blue). The ball and robot orientation detectors are implemented as a two-stage cascade of classifiers.

3.2 High contrast regions detection

Given the environmental conditions in which RoboCup soccer matches take place (soccer field and players' characteristics), an appropriate heuristic to speed up the process of finding the soccer ball and other players is to search for them in high contrast regions of the images. Accordingly, the grayscale input images are scanned using 16x16 pixels windows to find those regions. Any window laying outside the field boundary (determined using a priori knowledge of the field dimensions and the pose of the robot's camera) is automatically discarded. Windows containing body-parts of the observer robot are also discarded. Over each of the remaining windows, a threshold for binarization is estimated using Otsu's method (Otsu 1979). Only windows with a corresponding binarization threshold that is greater than a predefined value are considered to have high-contrast properties. Since the value utilized to select those windows may leave out image regions containing objects of interest, a dilation operation is applied on the selected windows. That is, all adjacent windows to any window considered to have high contrast properties, according to its binarization threshold, is also considered to have high contrast.

3.3 Robot proposals generator

The robot proposal generation applies vertical scan lines (y direction) over all the image's x-coordinates where high contrast regions were detected. The scan lines search for vertical abrupt contrast changes. Depending on the y coordinate of contrast changes found by the scan lines, a check is performed to see if enough of these detections have roughly the same y coordinate across the x-direction. If this is the case, the midpoints (in image coordinates) of all the sets of detections that fulfill this condition are considered to be the midpoints of the bottom segments of the bounding boxes containing the observed robot players. Then, by performing geometric sanity checks using a priori information of the other robot players (such as their height), the proposal generator provides a set of bounding boxes which may contain other robots' bodies. These sanity checks are similar to some of the rules used in Röfer et al. (2017), but adapted to be applied on a grayscale image. Moreover, all the rules that only rely on color information (such as checking for a player's jersey color, or counting colored pixels to get specific features) are not utilized. This approach is more robust

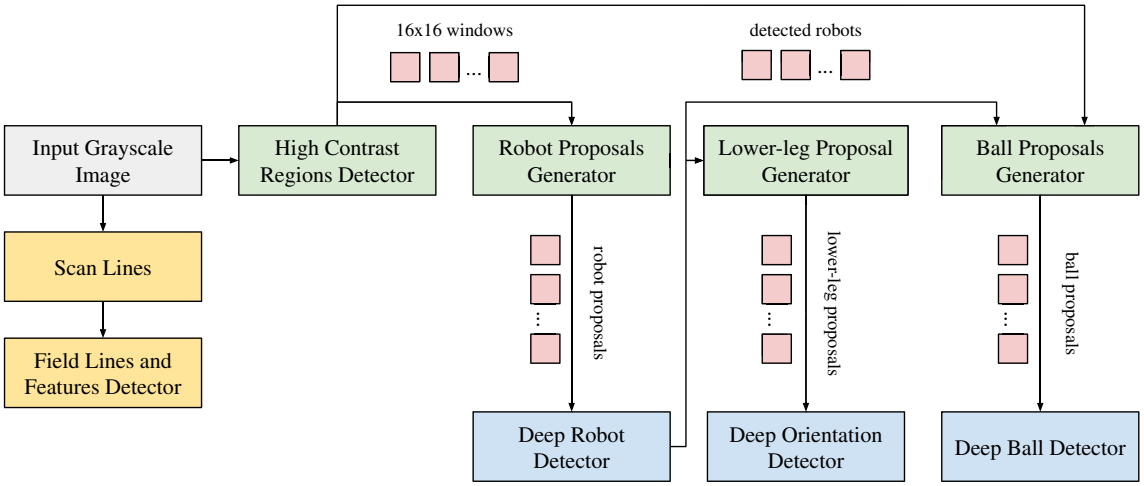


Fig. 1 Block diagram of the proposed vision system

to changes in lighting conditions, since it relies on local contrast information rather than on heuristic color segmentation. However, it may produce a much larger set of proposals since it has less filtering steps than the original pipeline proposed in Röfer et al. (2017).

3.4 Deep robot detector

The obtained robot proposals are then fed to a CNN that classifies the proposals as robots or non-robots. This CNN is based on the architecture proposed in Cruz et al. (2018), which will be described in Sect. 4.1. Using grayscale image regions allows the network to process in real-time a large number of robot proposals, since the reduction of input channels from 3 (color space) to 1 (grayscale) greatly reduces the CNN's inference time. The trained robot detection CNN will be called *RobotNet* in the experiments reported in Sect. 5. The team of each robot in the image is determined by analysing the region corresponding to the robot's shirt, which can be estimated given that the robot's position in the image is known, and using bounds over the standard RGB image to determine the color of the shirt. This approach works well since the shirts have a very high color saturation following the official rules of the SPL. It is important to note that this analysis does not require the color segmented image and its computational cost is very small given that only a small region of the image is analysed. The robot detection pipeline provides the estimation of the observed robots' positions and the team to which they belong. This information is latter combined, via wireless network, with the information gathered by the other teammates. This allows for an accurate estimation of the players' positions in the field, while also accounting for misdetections that single observers may be prone to commit.

Fig. 2 Major and minor lines depiction

3.5 Lower-leg region proposals generator

Inspired on the work presented in Mühlenbrock and Laue (2018), we propose an improved orientation determination method, which makes use of CNNs in order to achieve much better prediction accuracy than the original approach. The proposed method uses the bounding boxes of the detected robots as inputs, finds the regions that contain the lower-legs of each detected robot, and determines the body orientation of each robot by analyzing each lower-leg region. The lower-leg of each robot is characterized by two lines: the so-called *major* and *minor* lines. "The major line is defined from toe to toe and from heel to heel, while the minor line is defined as a side line of a foot" (Mühlenbrock and Laue 2018). Examples of both lines in different robot poses are shown in Fig. 2.

As a first step, the set of points that compose the robots' lower silhouette is calculated (Mühlenbrock and Laue 2018). Then, a region corresponding to the robot's feet is extracted and its Contrast-Normalized Sobel (CNS) image (Müller et al. 2012) is analyzed by using vertical scan lines. Over each scan line pixel, a horizontal median filter is applied and its response is compared to a threshold. Pixels with a filter response below the threshold are considered as part of the lower silhouette. Then, by iterating for each scan line, the subset of points that make up a closed convex region can be

Fig. 3 Lower-leg proposals and labels depiction

obtained by using Andrew's convex hulls algorithm (Andrew 1979). For each consecutive pair of points of the convex set, a line model in field coordinates is calculated. Each line model is then validated with the set of points of the lower silhouette, by using a voting methodology akin to the RANSAC algorithm (Fischler and Bolles 1981). The line with the highest number of votes is selected as the major line. Once the linear model has been chosen, the minor line may be generated by iterating over the remaining pairs of convex points. This line must fulfill a series of conditions such as a minimum and maximum length, and to be approximately orthogonal to the major line in order to be accepted as valid. Finally, the so-called "lower-leg" proposal is built based on the major and minor lines.

3.6 Deep robot orientation detector

While the major and minor lines can be used to calculate a rotation, the uncertainty on the direction of the robot means that there could be an error of 180° in the orientation estimation. Indeed, a major or minor line can correspond to both the front or the back of the robot. To solve this problem, the robot orientation is determined using a two-stage classification cascade of CNNs, where the first CNN discards low-quality lower-leg regions, and the second CNN determines the robot orientation.

Thus, for each lower-leg proposal, a CNN that measures its quality, *OriBoostNet*, is first applied. Proposals with too much motion blur or that do not correspond to the robots' feet are discarded. This results in a reduction on the number of wrong orientation estimations, since outliers' region proposals are discarded.

If a proposal is not discarded in the first stage of the cascade, it is then analyzed by a second CNN, *OriNet*, which classifies the lower-leg proposal as a *side*, *front* or *back* region. Examples of the proposed regions and their labels are shown in Fig. 3.

After the lower-leg proposals are classified, a consistency check is carried out by imposing that no more than one region of each class must exist for any given robot. This further reduces the number of incorrect orientation estimations. The rotation determination is performed by applying the inverse tangent from two points belonging to the major or minor lines. Then, by using the classes (side, front, back) assigned to each line, the direction of the line can be determined in order to tackle the symmetry problem and to estimate the correct robot orientation.

Finally, the temporal consistency of the orientation estimation is verified; the resulting orientation is added to a buffer that stores the last 11 estimations, and a circular median filter is applied over it. Moreover, in order to avoid invalid results, we consider that the orientation estimation as valid only for a small period of time if no new samples are added to the buffer.

3.7 Ball proposals generator

Our ball proposal generator is inspired on the hypotheses generator proposed in HTWK (2018). The main differences between both approaches are the following: (i) we only use grayscale images, (ii) we use a different method to estimate high contrast regions (see Sect. 3.2), and (iii) we use the robot detections to improve the generation of proposals.

The proposal generator uses both the detected high contrast regions and the detected robots' bounding boxes to generate ball hypotheses. The high contrast image regions are utilized because of the soccer ball's high contrast properties (black and white pattern), whilst the robot detections are utilized to discard some of the regions in which a ball detection would be highly unlikely. This way, the detected robot's bounding boxes are used to filter out any high contrast region that would lie on a detected robot's body, keeping those regions lying on the robot's feet.

The filtered image regions are then scanned in a pixel-wise fashion, and the radius that the soccer ball would have for all of the traversed image coordinates is calculated (considering prior knowledge of the field, the ball size, and the robot's camera pose). These radii are used to set the support region of *Difference of Gaussians* (DoG) filters, which are constructed and applied for every image coordinate where a ball radius was calculated. Only the highest filter responses are considered as a ball proposal. This process follows the same principles that the blob search performed to find keypoints in the SIFT algorithm (Lowe 2004).

3.8 Deep ball detector

The ball detection is carried out using a two-stage cascade of CNNs-based classifiers, where the first CNN discards region containing objects that are very different from balls, and the second CNN takes the final decision.

In order to speed up the detection process, the number of detected balls by the first CNN, *BoostBallNet*, is limited to a maximum of five, and then, they are sorted based on their confidence. Then, the second CNN, *BallNet*, analyzes

the sorted ball hypotheses to detect the ball. Once the second CNN detects a ball, the remaining hypotheses are discarded.

3.9 Field lines and special features detection

The field lines and features detection is based on the algorithm proposed in Röfer et al. (2017). The main difference with respect to the original approach, is that in the proposed framework no color information is used. To do this, a set of vertical and horizontal scan lines are used, which save transitions from high-to-low and low-to-high pixel's values. This allows the detection of a set of points which are then fed to the algorithm described in Röfer et al. (2017), in order to associate them with lines and other field features, such as the middle circle, the corners, and line intersections. More details about the algorithm can be found in Röfer et al. (2017).

4 Design and training of the CNN-based detectors

In this section we describe the design and training methodologies used to obtain the CNN based classifiers used in the proposed vision framework. Section 4.1 presents the network architecture of the classifiers, and Sect. 4.2 describes the active learning procedure used to train them.

4.1 Base CNN

The proposed vision system is composed of several statistical classifiers. Each of these classifiers, *RobotNet*, the robot detector, *BoostBallNet* and *BallNet*, the two cascade-stages of the ball detector, and *OriBoostNet* and *OriNet*, the two cascade-stages of the robot orientation estimation network– uses as base the same CNN architecture. The preliminary version of this architecture (*miniSqueezeNet*) was described in Cruz et al. (2018), while in this work slight variations are incorporated to achieve higher processing speeds, while maintaining accuracy.

The main component of *miniSqueezeNet* is the *extended Fire module*, which was proposed in Cruz et al. (2018), inspired by the original *Fire module* (Iandola et al. 2016) and on GoogleNet's *inception module*. This module uses a 1×1 filter placed at the beginning of each extended Fire module to compress the size of the representation into a feature tensor with less channels. This compressed representation is then fed to filters of different sizes; small filters are used to extract spatially local information, while bigger filters obtain global information which is more spatially spread out. The features obtained from these filters are then combined into a single tensor by means of channel wise concatenation and then fed to the next layer. Following this approach allows the

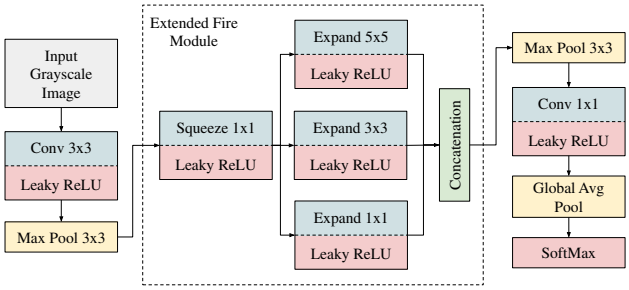


Fig. 4 Modified MiniSqueezeNet network structure

training of performant models whose use is computationally inexpensive.

In Cruz et al. (2018) guidelines for designing CNN architectures to be used in embedded systems with low processing capabilities are proposed. The main design variables are the depth of the network and the number of filters in each layer. In addition, it is proposed to use max-pooling operations implemented using non-overlapping windows to reduce the inference time. Following these guidelines and using the *extended Fire module*, the so-called *miniSqueezeNet* was designed for the detection of robots in real-time while playing soccer (Cruz et al. 2018).

In this work the *miniSqueezeNet* is further improved. First, grayscale images instead of color images are used as inputs, which reduces the number of input channels from three to one, and modifies the whole structure of the network. Second, leaky ReLU (Maas et al. 2013) instead of ReLU is used as activation function. Previously, we used ReLU in most layers, however, this sometimes resulted in the "dying ReLU" problem while training (no gradients flow backward through the neurons). The use of leaky ReLU solves this, while incurring in no accuracy losses. Further fine-tuning was performed on the networks' structure in order to estimate the correct input size and the required number of parameters. This was done by manually modifying the number of filters in accordance with the requirements of the problem.

A diagram of the new base CNN structure is presented in Fig. 4. All CNN based classifiers were developed using the Darknet library (Redmon 2013), and trained according to the methodology described in the next section. Taking into account the specific needs of the problem, variations on the number of convolutional filters were used for each of the CNN classifiers. The exact parameters used for each convolutional and maxpooling layer of the trained CNNs can be found in Table 1. Each one of this layers is then followed by batch-normalization and a leaky ReLu activation function.

4.2 Active learning training methodology

The use of an appropriate methodology for the training of the classifiers, which considers realistic game conditions, is cru-

Table 1 Structure of the trained CNNs

Layer	RobotNet	BoostBallNet	BallNet	OriBoostNet	OriNet
Conv 3 × 3					
Size	3 × 3	3 × 3	3 × 3	3 × 3	3 × 3
Filters	12	4	10	4	12
Stride	2	2	2	2	2
Max Pool					
Size	3 × 3	3 × 3	3 × 3	3 × 3	3 × 3
Stride	2	2	2	2	2
Squeeze 1 × 1					
Size	1 × 1	1 × 1	1 × 1	1 × 1	1 × 1
Filters	6	2	4	4	6
Stride	1	1	1	1	1
Expand 1 × 1					
Size	1 × 1	1 × 1	1 × 1	1 × 1	1 × 1
Filters	6	2	4	4	6
Stride	1	1	1	1	1
Expand 3 × 3					
Size	3 × 3	3 × 3	3 × 3	3 × 3	3 × 3
Filters	3	3	2	4	3
Stride	1	1	1	1	1
Expand 5 × 5					
Size	5 × 5	–	5 × 5	–	5 × 5
Filters	3	–	2	–	3
Stride	1	–	1	–	1
Max Pool					
Size	3 × 3	3 × 3	3 × 3	3 × 3	3 × 3
Stride	2	2	2	2	2
Conv 1 × 1					
Size	1 × 1	1 × 1	1 × 1	1 × 1	1 × 1
Filters	2	2	2	2	3
Stride	1	1	1	1	1
Avg Pool					
Size	Global	Global	Global	Global	Global

cial to obtain high performant classifiers. We implemented an active learning procedure that selects and annotates unlabeled data obtained under realistic conditions. The training process has several stages which are described in the following paragraphs.

As a first step, the different CNNs are trained using publicly available soccer-robotics datasets, e.g., the SPQR dataset (Albani et al. 2017). However, when the trained CNNs are used for processing images obtained under realistic soccer conditions, the classifiers will likely behave poorly because there is a distribution mismatch between this kind of images and the samples present in the public datasets.

To address this problem, the classifiers must be fine-tuned using the same kind of samples that would actually reach the networks during games. Examples of such images are shown in Fig. 5. To accomplish this, the vision system is deployed on the NAO robot and data is collected using the objects proposal algorithms. Each obtained proposal is classified and stored in the robot's memory with its corresponding label. To get uncorrelated data, we set a constraint for the object's hypotheses to be saved: for the robot proposals and lower-leg proposals for orientation determination, data is acquired periodically in accordance to a predefined time step; for the ball proposals, samples can only be saved if no other proposals with the same position and estimated radius were previously collected. The next stage consists of actively checking the data saved by the observer robot, and manually annotating only the samples that were incorrectly labeled. We then aggregate this data to the original data set and re-train the models.

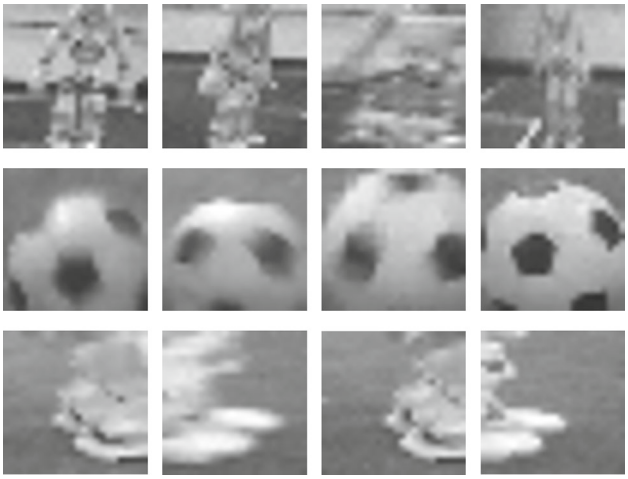


Fig. 5 First row: robot samples, second row: ball samples, third row: feet side samples

The above process is repeated until each CNN reaches a high performance. By doing this, we are progressively aggregating correctly labeled samples to acquire enough training data for robust feature learning, but we are also aggregating samples which the models fail to correctly infer, to encourage changes in the decision boundaries of the classifiers.

After we obtain proficient models by following the described methodology, we further enhance them by switching to a bootstrap procedure. To do this, we add confidence-based constrains to collect new training data in environments where the objects we want to detect are absent. For instance, if we are getting false positives from the ball detector, we would set the NAO robot to collect data in environments were no balls are present, and we would store every high confidence detection, relabelling them afterwards as non-balls. The samples collected would then be used to re-train the ball classifiers. Likewise, if the orientation detector is labeling a front region as a back region, generating a false positive, we would set the NAO robot to collect data in an environment where only back lower-leg regions are visible to re-train the classifier. Notice that the fine tuning procedure is applied over the detector, which means that when a cascade of CNNs is utilized, a sample is stored based on the compound performance of the CNNs, being the confidence constrain only considered for the last network involved in the classification. This active learning-bootstrap procedure results in a dramatic improvement in the performance of the classifiers after only a few iterations, and also allows the fine tuning of the CNN parameters by means of using data aggregation when an abrupt domain change occurs. Since the inputs to our models have relatively low dimensionality, the space used in the NAO memory during the data collection process is very small, for instance, 1000 robot proposal samples weight about 3 MB. This procedure, combined with the semi-supervised selection and labeling of the new samples, makes the training process extremely time-wise efficient.

5 Results

5.1 CNN classification

All classifiers were trained using the methodology described in the previous section. Table 2 shows the obtained model complexity (number of CNN parameters), average inference time (on the NAO robot), and accuracy calculated over a balanced database with a 50% of positive and 50% negative samples for each developed CNN.

Results show that the classifiers achieve very high performance while maintaining low inference times, which proves that their use is suitable for real time applications, such as robotic soccer. This also validates the effectiveness of the proposed methodology for the design and training of the classifiers. Finally, this also shows that the use of color information is not necessary to detect robots or balls when using classifiers such as CNNs. In fact, the CNN used in the robot detector achieves a similar accuracy rate that the model proposed in Cruz et al. (2018), while being approximately 2.75 times faster.

5.2 Robots, ball and field features detection systems

In order to evaluate the designed robot/ball proposal generators and classifiers, we acquire about 600 frames by a humanoid robot player under typical and challenging game conditions. Several lighting conditions were imposed while collecting these frames in order to test the robustness and reliability of our modules. Some examples of the cropped samples obtained from these frames can be found in Fig. 6. The testing database can be found at https://drive.google.com/file/d/1qAoQVU3H7JUuzAeNob2Sa6Qy7x62xZ7o/view?usp=sharing. The analysis of these frames allowed the extraction of empirical results in relation to the performance of the proposals generators and the CNN based classifiers, which are shown in Table 3. Examples of robot and ball detections can be found in Figs. 7 and 8.

Results show that the robots and ball proposals generators achieve high recall rates, while producing an average number of proposals per frame that can be processed in real time by the subsequent classifiers. Given the recall rate of the ball proposals module and the percentage of true positives of the boosting stage, the overall detection module has a very high detection rate. In fact, our ball detector outperforms B-Human's implementation proposed in Röfer et al. (2017), which achieves an overall accuracy rate of 0.697 when testing it under the same conditions. One of the main advantages of our ball detector is that it can identify balls at large dis-

Table 2 Performance of the developed CNNs (Leiva et al. 2019)

Model	RobotNet	BoostBallNet	BallNet	OriBoostNet	OriNet
Input size	24 × 24 × 1	12 × 12 × 1	26 × 26 × 1	12 × 12 × 1	24 × 24 × 1
No. of parameters	884	125	444	246	657
Inference time (ms)	0.382	0.043	0.343	0.059	0.329
Accuracy	0.969	0.965	0.984	0.962	0.984

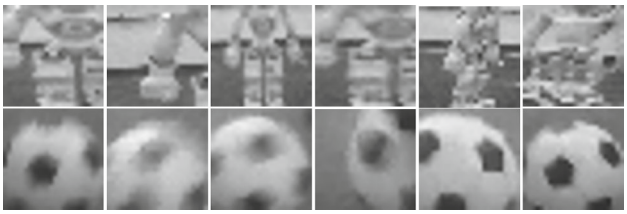


Fig. 6 Example images from our dataset. First row: robot samples, second row: ball samples

Table 3 Performance of the robots and ball detection systems (Leiva et al. 2019)

Module	Robot detector	Ball detector
Proposals per frame	3.05	10.3
Proposals recall	0.972	0.993
Overall accuracy	0.949	0.971

tances since it does not rely on low resolution scanlines. The proposed approach generates consistent detections with an accuracy rate above 0.9 within a 4.5 m range which is enough to generate a reliable model of the ball. From this point detections decay rapidly, with some detections still possible at a range of 6 meters to the ball. This is useful since even sporadic detections are enough to give an approximate location of the ball.

Similarly, the robot detector achieves high recall for the proposal generation and an overall very high accuracy.

Finally, the field lines and features detector was tested by comparing the difference between the real and the estimated robot pose. The estimation was obtained by using the field lines and features detected by our module. By using this approach we calculated a mean squared error of 40.07 mm, which indicates a suitable accuracy and reliability.

5.3 Robot orientation determination

The proposed robot orientation determination system is compared with the one proposed in Mühlenbrock and Laue (2018) (BH: B-Human), which is the only orientation determination system for NAO robots reported in the literature. We analyzed two flavors of our system: the proposed base orientation determination system (UCh), and its output after applying a circular median filtering (UChF). Some examples

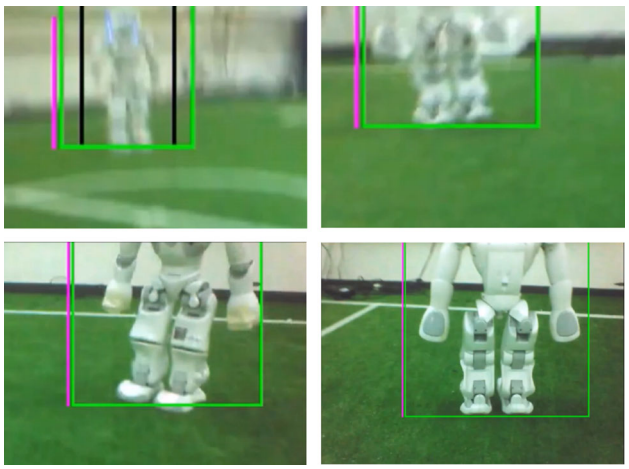


Fig. 7 Examples of robot detections, showing robots' bounding boxes

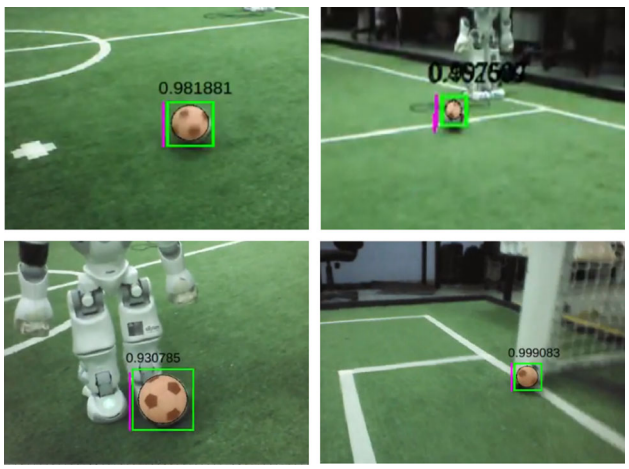


Fig. 8 Examples of ball detections, showing ball bounding boxes and confidence estimations

of the detected rotations as well as the corresponding major and minor lines are shown in Fig. 9.

In the first experiment (static robot), whose results are shown in Fig. 10, the observer and the observed robot are static and placed at a distance of 120 cm from each other. For each measurement the observed robot was rotated 22.5° around its axis. As in Mühlenbrock and Laue (2018), we define a *false positive* as any estimation that deviates more than a tolerance angle of 11.25° from the ground-truth. The orientation is classified as *semi perceived* when the rotation

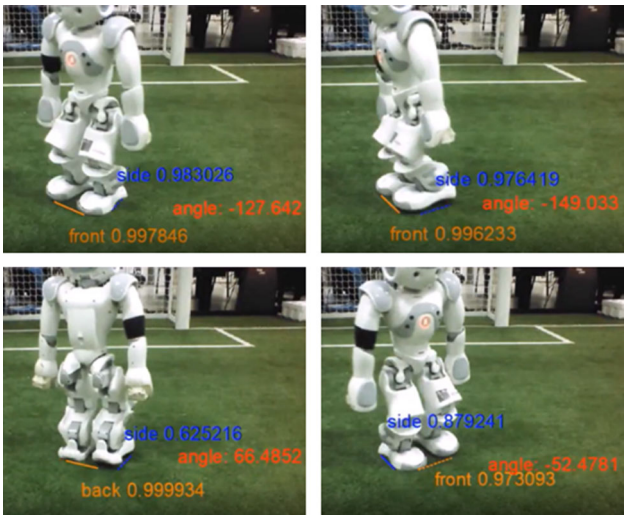


Fig. 9 Top number: confidence minor line. Middle number: estimated rotation. Bottom number: confidence mayor line

can be determined but the facing direction is unknown. The class *not perceived* corresponds to any frame where the orientation could not be calculated, while an orientation estimation is *perceived* if it does not deviate more than a tolerance angle of 11.25° from the ground-truth orientation.

In the second experiment (moving robot), whose results are shown in Fig. 11, the observed robot is moving at a speed of 12.0 cm/s, while the observer remains static. The observed robot is rotated in 45° around its axis for each measurement. We define the same classes for the orientation estimations as in the static experiment, but using a tolerance angle of 22.5°.

As shown in Figs. 10 and in 11, the proposed method outperforms the baseline system (Mühlenbrock and Laue 2018). The orientation estimation is completely perceived 99.88% of the time in static conditions, and 95.52% of the time in the dynamic experiment. It is clear that the algorithm proposed is better at determining the facing direction of the observed robots. This results in an increased number of completely perceived orientations while sharply decreasing the number of semi perceived orientations. Also, noise filtering techniques such as the median filter and RANSAC algorithm, combined with the utilization of a CNN contribute to lowering the number of false positive estimations. Finally, the integration of the circular median filter further reduces the number of false positives.

5.4 Profiling

Table 4 shows the maximum and average execution times for the different modules of the proposed vision framework when

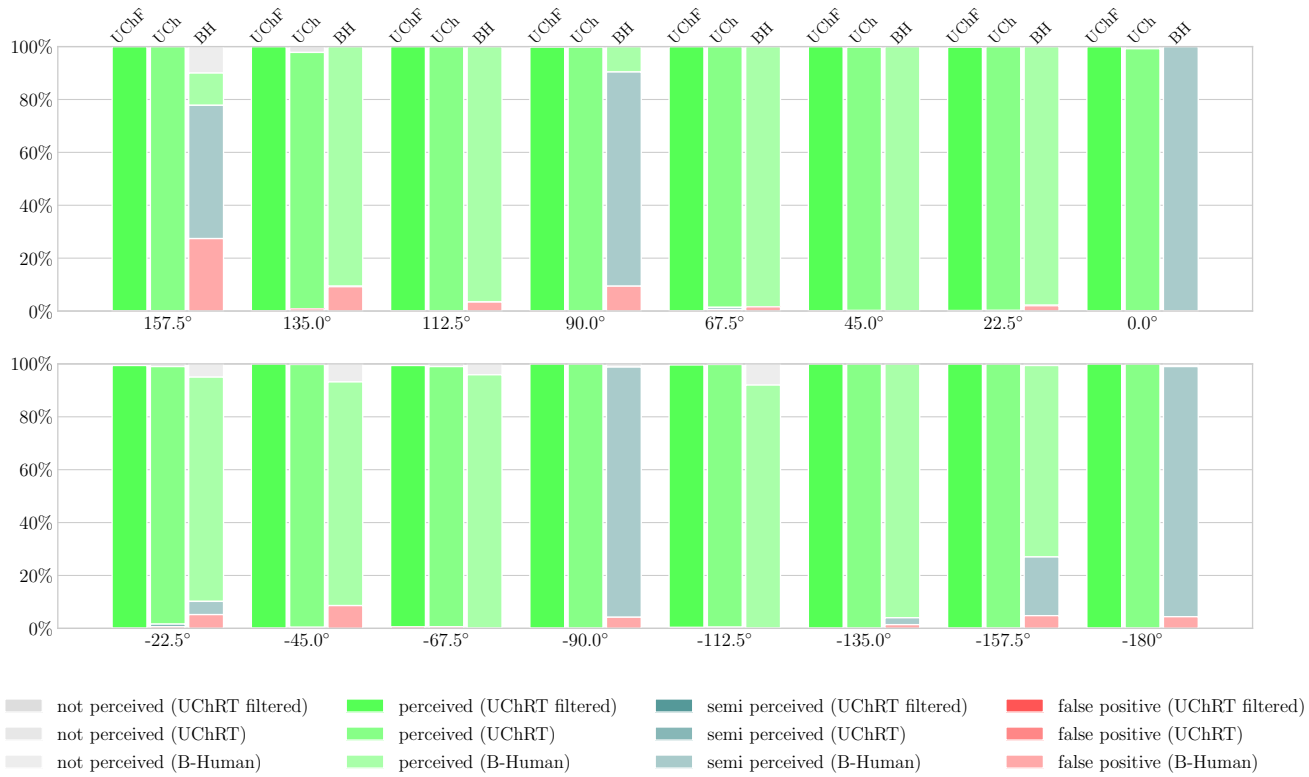


Fig. 10 Results obtained for the first experiment. Graph shows a performance comparison between raw (UCh) and filtered (UChF) estimations for our orientation detector and a B-Human system replication (BH) (Leiva et al. 2019)

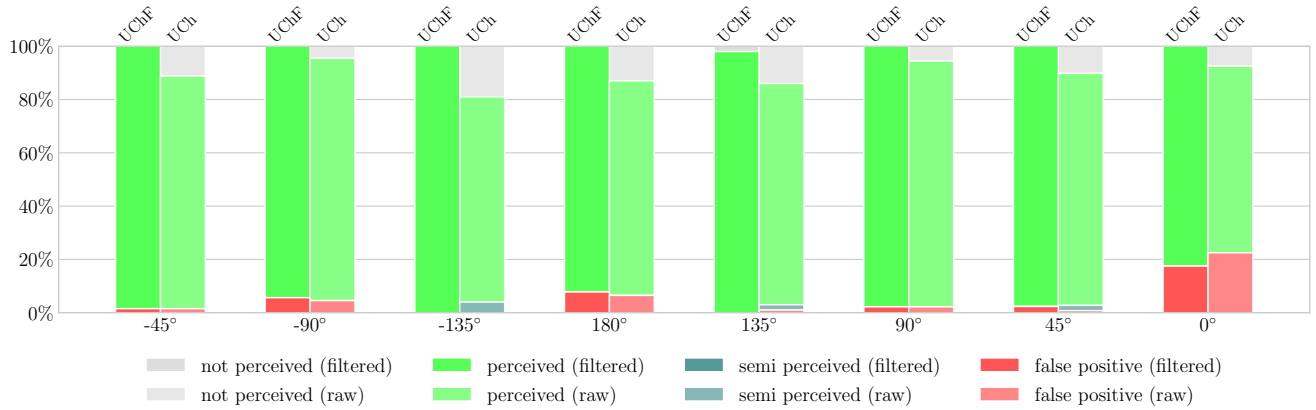


Fig. 11 Dynamic experiment results. Graph shows a performance comparison between raw (UCh) and filtered (UChF) estimations for our orientation detector (Leiva et al. 2019)

Table 4 Vision framework profiling. Maximum (Max.) and Average (Avg.) processing time in a NAO v5 platform (Leiva et al. 2019)

Module	Max. (ms)	Avg. (ms)
High contrast regions detector	2.755	1.478
Field lines and features detector	2.909	1.300
Robot proposals generator	2.692	1.083
Robot detector	2.417	0.939
Robot orientation detector	4.537	1.366
Ball proposals generator	2.506	1.132
Ball detector	6.959	2.452

deployed on the NAO v5 platform. The results obtained show that the proposed color-free vision system is deployable on platforms with limited processing capacity (such as the NAO robot). In addition, they prove the importance of the dimensionality reduction of CNN-based classifier inputs, and how this design decision impacts the performance of the system from a time-efficiency point of view.

5.5 Approach comparison

To further validate our approach, we compare the proposed detectors to those included in the latest release of the B-Human soccer code (Röfer et al. 2019). The comparison is performed on a realistic simulator (Cruz and Ruiz-del Solar 2020) able to produce images that closely match reality by using a generative model, trained with images collected from real soccer environments. Samples from the realistic simulator are show in Fig. 12. The simulator is able to randomly shift the pose of the robot in the field as well as the pose of all the other objects in the scene (opponent robots and ball). Moreover, the simulator is also able to provide ground truth information to calculate precise statistics.

We use this simulator to evaluate the performance of the proposed ball and robot detector systems as well as the performance of the robot (Javadi et al. 2018) and ball detectors systems of the B-Human team. Given that teams use different pipelines to detect objects, we define a detection system as the combination of modules that are used to detect an object in the scene. For our framework this means a combination of the region proposal extractor and the region classifier. Other teams use different approaches such as B-Human's robot detector system which is composed of an end-to-end detection model.

All four systems were trained using a data set composed of real samples. In the case of our ball and robot detectors we use the exact same models that were used to achieve the results presented in Table 2. Table 5 presents the metrics collected for both robot detectors systems, while Table 6 presents the results of the ball detector systems. The reported recall and precision metrics correspond to the detection system as a whole, in accordance to how results are presented in Javadi et al. (2018). The module's average times are measured on a NAO v5 robot.

From Table 5 it can be seen that our method offers better recall than its B-Human counterpart. We found that this difference is the result of a better detection rate of robots facing sideways to the camera. Furthermore, our proposed methodology is less computationally expensive when tested on a NAO v5 robot, running at more than double the average speed when compared to the B-Human approach, as reported in Javadi et al. (2018). We attribute this to the simplicity of our CNN model, which achieves state of the art performance with fewer computations.

Table 6 shows the corresponding metrics for the ball detector systems. Both approaches are very similar and consist on a region proposal extractor followed by a CNN classifier that takes as input the proposed region in gray scale and outputs the probability that the sample corresponds to a ball. This is

Fig. 12 Image samples generated by the realistic simulator and used to estimate performance metrics

Table 5 Performance of the robots detection systems

Robot detector system	Ours	B-Human
Detector recall	0.847	0.702
Detector precision	0.985	0.962
Average time (ms)	2.0	4.5

Table 6 Performance of the ball detection systems

Ball detector system	Ours	B-Human
Detector recall	0.806	0.831
Detector precision	0.987	0.986
Average time (ms)	3.4	–

then followed by an estimation of the ball's position in the image. Given the similarity of the approaches, it is not surprising that both methods achieve very similar performances in terms of recall and precision. We report the average time of our proposed methodology on a NAO V5, however the average time in a NAO V5 for the B-Human detector is not reported in the literature.

The above results show that the proposed vision framework is competitive with the ones of other teams that consistently reach top spots on the SPL, such as B-Human. This speaks to the overall quality of the proposed system.

5.6 Applicability in other domains

We hypothesize that the effectiveness of a vision system built using the proposed approach in a different domain, would depend on the availability of exploitable patterns and regularities in that domain. Furthermore, for such system to function in real-time, its applicability would be restricted to domains in which hand-engineering computationally inexpensive proposal generators is feasible.

Such domains correspond, for instance, to structured environments in which the lighting conditions and the overall geometrical layout of the scene are stable over time. This kind of environments usually corresponds to some indoor spaces, such as industrial plants, and warehouses. Stores and hotels are also viable candidates for this kind of approach. In recent years, robots have begun to become more ubiquitous in this kind of working environments to offload some work from human operators by performing tasks such as greeting costumers, and delivering room service. Since these kind of robots are often low cost, they usually have low computational capacity, which renders them an ideal target to implement vision systems similar to those proposed on this paper.

Unstructured environments may also be approachable when using images that contain information regarding their state that simplify their complexity. For instance, the problem of generating proposals for object detection in a complex indoor scene may be simplified if depth or thermal information is available, and the target objects have a known shape and size.

As a proof of concept of these ideas, we implemented a detector for human soccer players as observed by thermal cameras (see Fig. 13). The detector consists of a proposal generator similar to that described in Sect. 3.7, and a CNN-based classifier that has the RobotNet architecture (see Table 1).

For training and evaluating this detector, the data set presented in Gade and Moeslund (2018) was utilized. This data set is constructed by stitching three simultaneously obtained thermal images from an AXIS Q1922 thermal camera, resulting in 1920×480 pixels images. These images contain between six to eight soccer players in and indoor field (Gade and Moeslund 2018).

As a proposal generator of the human players, the ball proposal generator described in Sect. 3.7 was modified so that the support region of the DoG filters applied would coarsely match the silhouette of the players. Moreover, these non-square filters were applied using two different scales. Contrary to the approach adopted for ball detection, the proposal generator this time was applied over the entire image. The produced proposals are resized to 32×32 pixels, and fed to a CNN-based classifier that was trained using labeled proposals from a fraction of the data set.

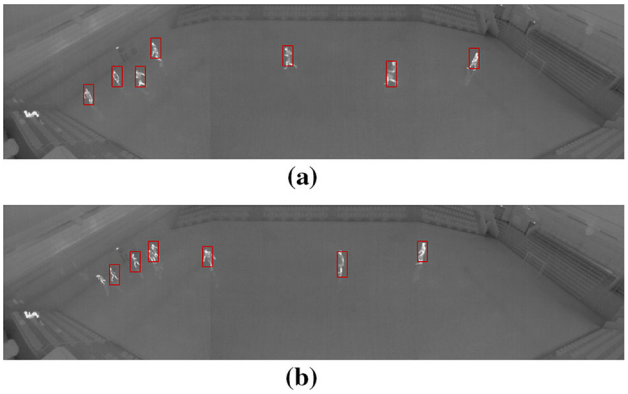


Fig. 13 Human soccer players' detection in thermal cameras. **a** Correctly detected players, **b** a false negative detection, and two players incorrectly grouped as one

Table 7 Performance of the human soccer players detector

Module	Detector
Proposals per frame	15.2
Detector system recall	0.806
Detector system precision	0.935

Figure 13 shows the results provided by the detector over two image samples. The performance metrics for the detector are displayed in Table 7. The performance of the detector could be further improved using tracking, which integrates information over time and thus reduces the number of false negatives by propagating information between consecutive frames, as proposed in Gade and Moeslund (2018). Including more heuristics to the region proposal generator could also improve the overall detector's performance. However this falls outside the scope of this paper. Overall, these results support our hypothesis regarding the applicability of the proposed approach to domains beyond the RoboCup SPL, as suitable solutions can be obtained by constructing systems based on some of the processing pipelines of our detectors and their CNN-based classifiers.

6 Conclusions

This paper describes a new vision framework that does not use any color information. This is a novel approach for vision systems designed for the RoboCup SPL, achieving very high performance while being computationally inexpensive.

The proposed vision system we present introduces four new modules: a redesigned robot detector, a visual robot orientation estimator, a brand new ball detector, and finally, a color-free field lines and features detector. All modules developed for this paper are able to run simultaneously in real-time when deployed on a NAO robot playing soccer.

Moreover, we demonstrate that CNN-based classifiers are a useful tool to solve most of the perception requirements of humanoid soccer robotics, and generally translate in an overall better performance of the corresponding modules when coupled with good region proposal algorithms, and a proper use of design and training techniques.

Furthermore, the proposed framework is successfully validated in a different domain, where human soccer players are detected using thermal images. This shows the applicability of the proposed framework beyond soccer robotics.

Acknowledgements The authors thank Kenzo Lobos-Tsunekawa for his contributions on the development of the robot detection system. We also thank Ignacio Bugueño for implementing the detection of the major and minor lines, which are used to estimate the precise rotation of the robot. Additionally, we thank him for helping to generate some of the databases used on this paper. This work was partially funded by ANID (Chile) Projects FONDECYT 1201170, PIA AFB 180004, and CONICYT-PFCHA/Magíster Nacional/2018-22182130.

References

Albani, D., Youssef, A., Suriani, V., Nardi, D., & Bloisi, D. D. (2017). A deep learning approach for object recognition with NAO soccer robots. In S. Behnke, R. Sheh, S. Sarıel, & D. D. Lee (Eds.), *RoboCup 2016: Robot World Cup XX* (pp. 392–403). Cham: Springer International Publishing.

Andrew, A. (1979). Another efficient algorithm for convex hulls in two dimensions. *Information Processing Letters*, *9*(5), 216–219. https://doi.org/10.1016/0020-0190(79)90072-3.

Cruz, N., Lobos-Tsunekawa, K., & Ruiz-del Solar, J. (2018). Using convolutional neural networks in robots with limited computational resources: Detecting NAO robots while playing soccer. In H. Akiyama, O. Obst, C. Sammut, & F. Tonidandel (Eds.), *RoboCup 2017: Robot World Cup XXI* (pp. 19–30). Cham: Springer International Publishing.

Cruz, N., & Ruiz-del Solar, J. (2020). Closing the simulation-to-reality gap using generative neural networks: Training object detectors for soccer robotics in simulation as a case study. In *The international joint conference on neural networks 2020*

Felbinger, G. C., Göttsch, P., Loth, P., Peters, L., & Wege, F. (2019). Designing convolutional neural networks using a genetic approach for ball detection. In D. Holz, K. Genter, M. Saad, & O. von Stryk (Eds.), *RoboCup 2018: Robot World Cup XXII* (pp. 150–161). Cham: Springer International Publishing.

Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, *24*(6), 381395. https://doi.org/10.1145/358669.358692.

Gabel, A., Heuer, T., Schiering, I., & Gerndt, R. (2019). Jetson, where is the ball? Using neural networks for ball detection at robocup 2017. In D. Holz, K. Genter, M. Saad, & O. von Stryk (Eds.), *RoboCup 2018: Robot World Cup XXII* (pp. 181–192). Cham: Springer International Publishing.

Gade, R., & Moeslund, T. B. (2018). Constrained multi-target tracking for team sports activities. *IPSJ Transactions on Computer Vision and Applications*, *10*(1), 2.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., et al. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. 1704.04861

HTWK, N. T. (2018). Nao-team htwk: Team research report. Retrieved from 30 January, 2020 http://www.htwk-robots.de/documents/TRR_2017.pdf?lang=en.

Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., & Keutzer, K. (2016). Squeezenet: Alexnet-level accuracy with 50x fewer parameters and 0.5mb model size. 1602.07360.

Javadi, M., Azar, S. M., Azami, S., Ghidary, S. S., Sadeghnejad, S., & Baltes, J. (2018). Humanoid robot detection using deep learning: A speed-accuracy tradeoff. In H. Akiyama, O. Obst, C. Sammut, & F. Tonidandel (Eds.), *RoboCup 2017: Robot World Cup XXI* (pp. 338–349). Cham: Springer International Publishing.

Kukleva, A., Khan, M. A., Farazi, H., & Behnke, S. (2019). Utilizing temporal information in deep convolutional network for efficient soccer ball detection and tracking. In S. Chalup, T. Niemueller, J. Suthakorn, & M. A. Williams (Eds.), *RoboCup 2019: Robot World Cup XXIII* (pp. 112–125). Cham: Springer International Publishing.

Leiva, F., Cruz, N., Bugueño, I., & Ruiz-del Solar, J. (2019). Playing soccer without colors in the spl: A convolutional neural network approach. In D. Holz, K. Genter, M. Saad, & O. von Stryk (Eds.), *RoboCup 2018: Robot World Cup XXII* (pp. 122–134). Cham: Springer International Publishing.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*(2), 91–110. https://doi.org/10.1023/B:VISI.0000029664.99615.94.

Maas, A. L., Hannun, A. Y., & Ng, A. Y. (2013). Rectifier nonlinearities improve neural network acoustic models. In *ICML workshop on deep learning for audio, speech and language processing*.

Menashe, J., Kelle, J., Genter, K., Hanna, J., Liebman, E., Narvekar, S., et al. (2018). Fast and precise black and white ball detection for robocup soccer. In H. Akiyama, O. Obst, C. Sammut, & F. Tonidandel (Eds.), *RoboCup 2017: Robot World Cup XXI* (pp. 45–58). Cham: Springer International Publishing.

Mühlenbrock, A., & Laue, T. (2018). Vision-based orientation detection of humanoid soccer robots. In H. Akiyama, O. Obst, C. Sammut, & F. Tonidandel (Eds.), *RoboCup 2017: Robot World Cup XXI* (pp. 204–215). Cham: Springer International Publishing.

Müller, J., Frese, U., & Röfer, T. (2012). Grab a mug—object detection and grasp motion planning with the nao robot. In: *2012 12th IEEE-RAS international conference on humanoid robots (Humanoids 2012)* (pp. 349–356). https://doi.org/10.1109/HUMANOIDS.2012.6651543.

Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, *9*(1), 62–66. https://doi.org/10.1109/TSMC.1979.4310076.

Poppinga, B., & Laue, T. (2019). Jet-net: Real-time object detection for mobile robots. In S. Chalup, T. Niemueller, J. Suthakorn, & M. A. Williams (Eds.), *RoboCup 2019: Robot World Cup XXIII* (pp. 227–240). Cham: Springer International Publishing.

Redmon, J. (2013–2016). Darknet: Open source neural networks in C. Retrieved from 28 January, 2021. http://pjreddie.com/darknet/.

Redmon, J., & Farhadi, A. (2017). Yolo9000: Better, faster, stronger. In *The IEEE conference on computer vision and pattern recognition (CVPR)*.

RoboCup (2020a) Robocup federation official website. Retrieved from 30 January, 2020. https://www.robocup.org/objective/.

RoboCup (2020b) Robocup standard platform league official website. Retrieved from 30 January, 2020. https://spl.robocup.org/.

Röfer, T., Laue, T., Baude, A., Blumenkamp, J., Felsch, G., Fiedler, J., et al. (2019). B-Human team report and code release 2019. http://www.b-human.de/downloads/publications/2019/CodeRelease2019.pdf.

Röfer, T., Laue, T., Bülter, Y., Krause, D., Kuball, J., Mühlenbrock, A., Poppinga, B., et al. (2017). B-Human team report and code release 2017. Retrieved from 28 January, 2021. http://www.b-human.de/downloads/publications/2017/coderelease2017.pdf

Speck, D., Barros, P., Weber, C., & Wermter, S. (2017). Ball localization for robocup soccer using convolutional neural networks. In S. Behnke, R. Sheh, S. Sarıel, & D. D. Lee (Eds.), *RoboCup 2016: Robot World Cup XX* (pp. 19–30). Cham: Springer International Publishing.

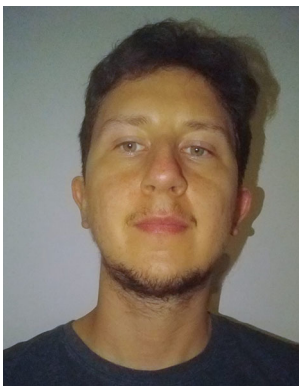
Speck, D., Bestmann, M., & Barros, P. (2019). Towards real-time ball localization using CNNs. In D. Holz, K. Genter, M. Saad, & O. von Stryk (Eds.), *RoboCup 2018: Robot World Cup XXII* (pp. 337–348). Cham: Springer International Publishing.

Szemenyei, M., & Estivill-Castro, V. (2019a). Real-time scene understanding using deep neural networks for robocup spl. In D. Holz, K. Genter, M. Saad, & O. von Stryk (Eds.), *RoboCup 2018: Robot World Cup XXII* (pp. 96–108). Cham: Springer International Publishing.

Szemenyei, M., & Estivill-Castro, V. (2019b). Robo: Robust, fully neural object detection for robot soccer. In S. Chalup, T. Niemueller, J. Suthakorn, & M. A. Williams (Eds.), *RoboCup 2019: Robot World Cup XXIII* (pp. 309–322). Cham: Springer International Publishing.

Teimouri, M., Delavaran, M. H., & Rezaei, M. (2019). A real-time ball detection approach using convolutional neural networks. In S. Chalup, T. Niemueller, J. Suthakorn, & M. A. Williams (Eds.), *RoboCup 2019: Robot World Cup XXIII* (pp. 323–336). Cham: Springer International Publishing.

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001* (vol 1, pp. I–I). https://doi.org/10.1109/CVPR.2001.990517



Nicolás Cruz studied Electrical Engineering at the University of Chile where he is part of the University's Robotics Laboratory. He is currently doing his master thesis on simulating environments using generative models.

Francisco Leiva studied Electrical Engineering at the University of Chile where he is part of the University's Robotics Laboratory. He is currently doing his master thesis on deep reinforcement learning.

Javier Ruiz-del-Solar received his degree in Electrical Engineering from the Universidad Tecnica Federico Santa Maria (Chile) in 1991, and the Doctor-Engineer degree from the Technical University of Berlin in 1997. Since 2009 he is Executive Director of the Advanced Mining Technology Center (AMTC) at the Universidad de Chile. His research interests include Mobile Robotics, Computer and Robot Vision, and Automation of Mining Equipment.