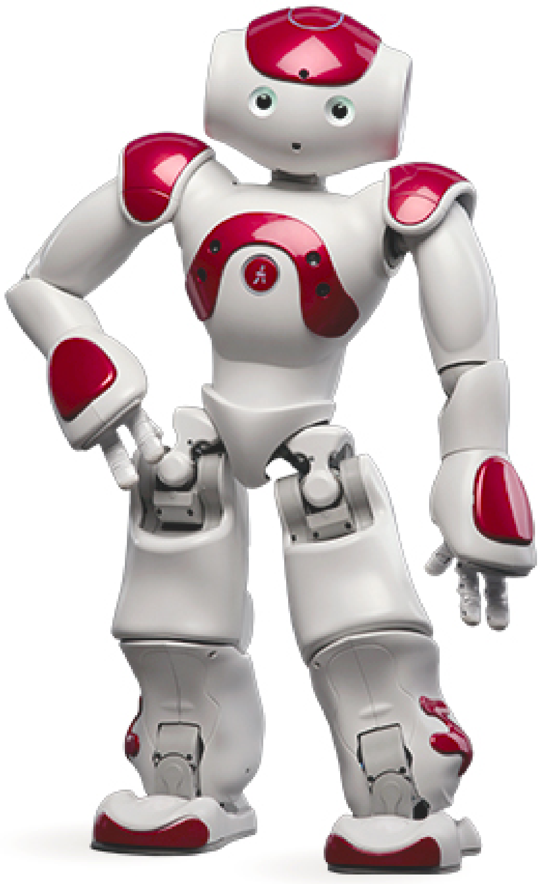
University of Amsterdam

BSc Artificial Intelligence Thesis

Ball Localization in Humanoid Soccer using Machine Learning

February 1, 2019



Ball Localization in Humanoid Soccer using Machine Learning

Jimmy Badrie
10465553

Bachelor thesis
Credits: 18 EC

Bachelor Opleiding Kunstmatige Intelligentie

University of Amsterdam
Faculty of Science
Science Park 904
1098 XH Amsterdam

Supervisor
dr. A. Visser

Informatics Institute
Faculty of Science
University of Amsterdam
Science Park 904
1098 XH Amsterdam

February 1st, 2019

Abstract

In pursuit of the ideal robotic soccer player it is imperative that it has the ability to accurately detect key objects related to the game of soccer. One of the most significant objects in the game is the soccer ball. Contemporary humanoid soccer, however, still rely on strict specifications regarding their environment, such stable lightning conditions and field borders. The RoboCup Standard Platform League, an international league for robotic soccer teams, challenges these limitations every year. In this thesis, a machine learning approach is presented using Local Binary Patterns and Haar-like Features in order to evaluate their use of ball detection in real-time robotic soccer. Both Algorithms are integrated into a cascade classification system, which is trained and tested on data involving the RoboCup Standard Platform League setting. Research have been conducted into the comparability of the two algorithms within the field of computer vision, supporting the obtained results in concluding the efficiency time-wise of training a classification system using Local binary patterns over Haar-like Features. However, accuracy results do recommend the use of Haar-like Features in professional matches. Nonetheless, the use of Local Binary Patterns does show significant potential for ball localization compared to a more renowned algorithm such as Haar-Like Features in the field of computer vision.

Contents

1 Introduction

1.1 Background

The field of artificial intelligence has given rise to various new applications in the past years. One of these fields is the area of sports. A computer called Deep Blue was developed by International Business Machines Corporation (IBM), which later came to be known for its victory over the reigning chess champion of 1997. This performance was a precedence unknown in the artificial intelligence community and set the stage for new research challenges in this field, which became known as a major turning point for the RoboCup [1].

The RoboCup Standard Platform League (SPL) is an international competition in which teams of robotic soccer players compete against one another. The RoboCup began as a project by researchers with the following goal:

> *"By the middle of the 21st 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 Federation[1],

The robots used in the RoboCup SPL are humanoid robots called Nao, which is seen in figure 1. This figure[2] displays the Australian robotic soccer team, USNW Sydney. Robotic soccer players in the Standard Platform League are currently dependant on a stable environment for object detection in order to play the game. Unlike humans who can play the game in various settings, like different lightning conditions, locations and even using different objects as a ball and goal. In order to move the boundaries of contemporary robotic soccer towards the initial RoboCup goal, the RoboCup Federation addressed the issue by removing artificial field elements such as borders and additional beacons in past editions of the RoboCup. There has been much research done for ball detection in RoboCup setting, including detection based on shape [2] and color [3]. Since the year 2014, the Standard Platform League introduced the replacement of a regular black/white ball, whereas before a single color-coded ball was used. This change in ball, made color segmented approached towards ball detection rather obsolete. As such, a change in approach was needed.

1.2 Research Focus

The main focus of this thesis will be on the analysis on whether the use of Local Binary Patterns is viable for ball localization in contemporary robotic soccer. To this extend, a machine learning approach will be taken in which a cascade classification system will be implemented in order to train and test a classifier for ball detection. Haar-Like Features will be used likewise in order to act as a baseline in order to evaluate the performance of Local binary patterns. Comparative research between Local Binary Patterns and Haar-Like Features for object detection, states the benefit using Local Binary Patters in training time

[1] `https://www.robocup.org/objective`
[2] `http://bitterempire.com/robocup-robot-soccer-world-cup/`

and robustness over Haar-like Features [4]. However, Accuracy is overall better when using Haar-Like Features. Assuming these statements, the hypothesis for ball-detection would be the same in terms of training and testing these algorithm. As such, it will be a matter of how much a difference there will be for it to be feasible in real-time use of a soccer match.

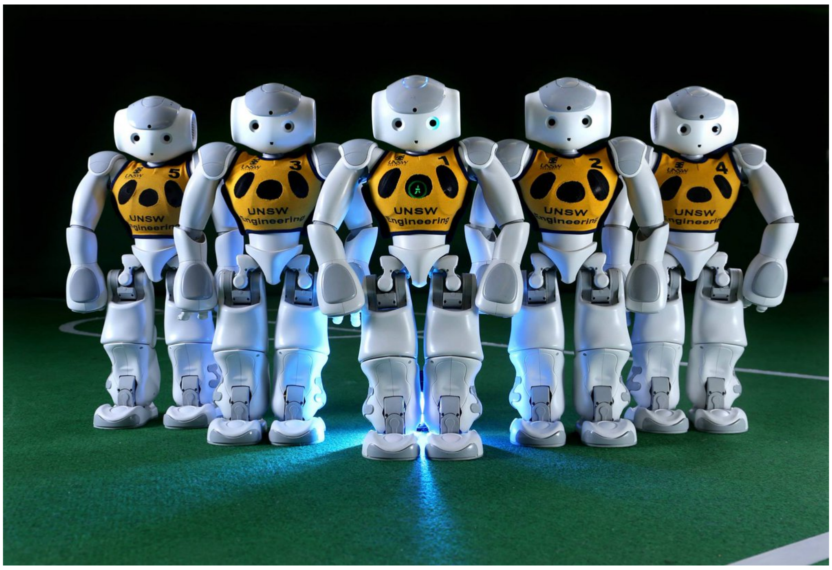


Figure 1: Robotic soccer team

1.3 Thesis Overview

Firstly, an overview of the used theory will be given. The main focus will be on the workings of Local Binary Patterns. Afterwards, The method for constructing, training and testing the classification system will be outlined. Followed by the results in the form of images and graphs, depicting localization capability and efficiency. A section for the conclusion is taken and discussed, following possible future research topics.

2 Theory

In this section, a theoretical overview is given of the used algorithms in question. The application of Local Binary Patterns (LBP) as well as the workings of the Haar-Like Features are explained which eventually come together in a cascade classification system.

2.1 Local Binary Patterns

The method of using Local Binary Patterns as a descriptor for object detection in computer vision was a novel approach, which was introduced by Ojala, Pietikainen and Harwood in 1994 [5]. This approach was eventually extended by using a circular neighborhood in 2002 [6]. Further improvement by Liao et al. in 2007 [7] utilized the algorithm as a series of descriptors which came to known as a Multi-Block Local Binary Pattern.

2.1.1 Basic LBP

The basic method of LBP is used on a 2-dimensional surface texture in which a 3x3 window of pixels is examined one at the time. For every observed cell of pixels, there is a numerical feature determined called the LBP-Code. This LBP-code is calculated by means of its surrounding neighborhood as seen in figure 2. This figure is taken from the book *Local Binary Patterns: New Variants and Applications* [8].

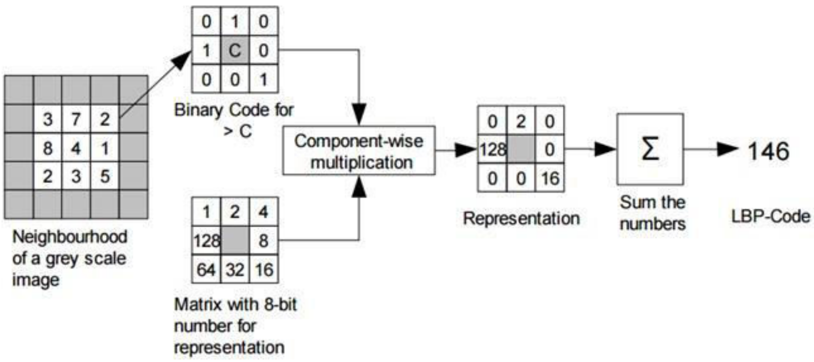


Figure 2: LBP feature extraction

The central pixel C acts as a threshold for its neighbors in order to calculate a binary code. In figure 2, the threshold is set at $C = 4$. Thus enabling its surrounding 8 cells to be labeled in a binary matter. Equation 1 outputs label $s(x)$ by evaluating pixel g_p for $p = 7$ against C. After the corresponding labeled window is computed, it is multiplied component-wise by a matrix with a 8-bit number representation. The following matrix is summed which converts the binary code into a decimal number representation. It is this conversion that computes the final LBP-Code for the corresponding window of pixels. (eq. 2)

$$s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \tag{1}$$

$$LBP = \sum_{p=0}^{7} s(g_p - C)2^p \tag{2}$$

Having a descriptive representation of every block in an image, it is possible to derive a texture descriptor. LBP does this in the representation of a feature histogram.

$$HI = \sum_{p=0}^{P-1} s(g_p - g_c)2^p \tag{3}$$

The benefit of expressing a texture descriptor using local binary patterns is that it is illumination invariant. As the lightning of the environment changes, the pixel values will change. However, the relative difference between the pixels will remain the same irrespective of global illumination variation.

2.1.2 Extended LBP

The approach of LBP was eventually extended by Ojala et al.in 2002 [6], using a circular neighborhood that could be modified. This offers a greater flexibility over the fixed quadrilateral. Various window sizes are displayed in figure 3, this image is taken from [9]. The computation of LBP-codes for every window concerning circular shaped windows is given in equation 4 in which P concern the amount of neighborhoods and R the radius of the window.

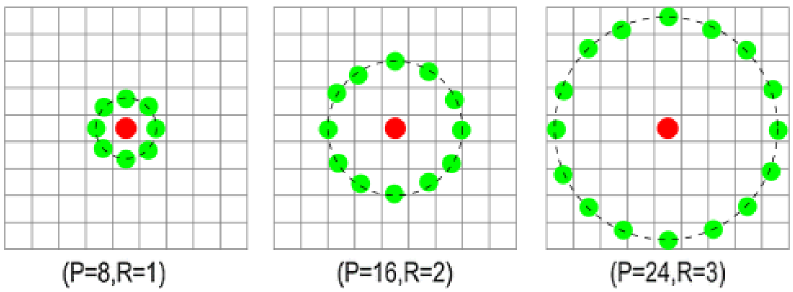


Figure 3: Region extension in LBP

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p \tag{4}$$

2.1.3 Bilinear interpolation

To counter unknown pixel values in neighbors when using circular windows, the use of bilinear interpolation is implemented beforehand in order to get an estimation of the corresponding pixel values. An implementation is given in the book *Numerical Recipes in C: The Art of Scientific Computing* [10]. This will be briefly explained.

Let $A = \{f(x,y), f(x_0,y), f(x_1,y), f(x,y_0), f(x,y_1)\}$ and $B = \{f_{00}, f_{01}, f_{10}, f_{11}\}$ be sets of sample points in a 2-dimensional grid, where the points in A are unknown and in B are known. The overview of these points are given on the right-hand side in figure 4. This image is taken from [10].

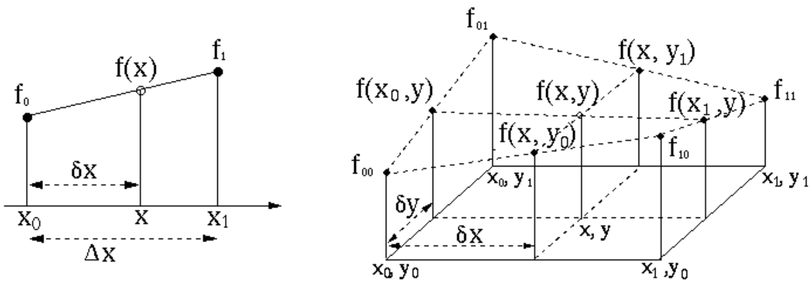


Figure 4: 1-D Interpolation and 2-D Interpolation

By performing linear interpolation in the x dimension, an estimation can be calculated by means of curve fitting as seen in equation 5 for the point $f(x, y_0)$ and $f(x, y_1)$. Assume $\delta i = i - i_0$ and $\Delta i = i_1 - i_0$ for $i = \{x, y\}$.

$$\begin{aligned} f(x,y_0) &\approx \frac{\delta x}{\Delta x}(f_{10} - f_{00}) + f_{00} \\ f(x,y_1) &\approx \frac{\delta x}{\Delta x}(f_{11} - f_{01}) + f_{01} \end{aligned} \tag{5}$$

Afterwards, linear interpolation in the y dimension is done following the equation 6. This results in a bilinear interpolating process as whole, win which the order of interpolation is irrelevant as the obtained expression is symmetric for both dimensions.

$$\begin{aligned} f(x,y) &\approx \frac{dy}{\Delta y}[f(x,y_1) - f(x,y_0)] + f(x,y_0) \\ &\approx \frac{dy}{\Delta y}\Big[\frac{\delta x}{\Delta x}(f_{11} - f_{01}) + f_{01} - \frac{\delta x}{\Delta x} \\ &\qquad (f_{10} - f_{00}) - f_{00}\Big] + \frac{\delta x}{\Delta x}(f_{10} - f_{00}) + f_{00} \\ &\approx \frac{\delta x}{\Delta x}\frac{dy}{\Delta y}f_{11} + \frac{dy}{\Delta y}\Big(1 - \frac{\delta x}{\Delta x}\Big)f_{01} + \frac{\delta x}{\Delta x}\Big(1 - \\ &\qquad \frac{dy}{\Delta y}\Big)f_{10} + \Big(1 - \frac{\delta x}{\Delta x} - \frac{dy}{\Delta y} + \frac{\delta x}{\Delta x}\frac{\delta y}{\Delta y}\Big)f_{00} \end{aligned} \tag{6}$$

2.1.4 Multi-Block Local Binary Patterns

Liao et al. improved the LBP algorithm of Ojala et al. [7]. Whereas the original LBP scanned a specified region throughout a whole image and assigning LBP-codes to each individual pixel, improvement was made by assigning not pixel-wise, but region-wise. Hence the name Multi-block LBP. See [7] for more information.

2.2 Haar-Like Features

Haar-Like Features are used in order to detect object in computer vision using a conversion of edges into features [11]. These features to detect edges are called Haar-Like Features and can be seen in figure 5. which is taken from [11]. Such basic elements are turned into a feature descriptor for object detection as described in the paper of Viola and Jones.

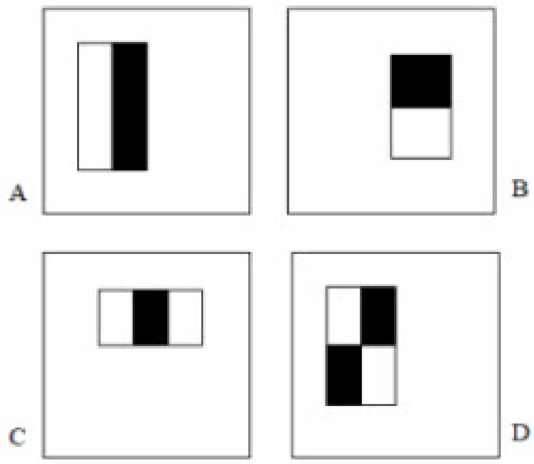


Figure 5: Haar-Like Features

2.3 Cascade Classifier

A cascade classifier is a type of machine learning classifier. In this classification structure, the overall structure is build upon a series of weak classifiers, designed to form one strong classifier. See [11] for more information.

3 Method

In this section, the general approach is outlined in which this experiment is build on. This involves acquisition and processing of experimental data, the layout of the pipeline for the classification system and the implementation of the ball detector.

3.1 Robot

The robot used in official RoboCup Standard Platform League matches are Nao robots. These robots were developed by Aldebaran Robotics with a humanoid design [12]. As it appears to have two eyes, the fact of the matter is that these are not used as a visual perceptor like humans do. The Nao robot uses two video cameras that provide a up to 1280x960 resolution at 30 frames per second. One is located on the forehead, whereas the other is located at a slightly lower position as seen in figure. The position alongside the angle of these cameras are given in figures 6 and 7 which is taken from the Aldebaran documentation[3]. It is from these two sensors that the robot obtains its visual data. Data originating from this source will be used to for image processing in order to detect balls.

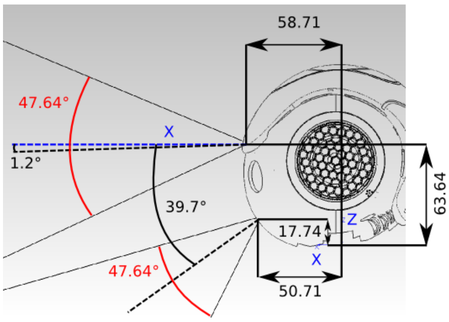


Figure 6: Side view camera

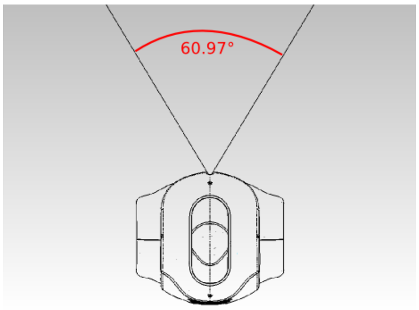


Figure 7: Top view camera

3.2 Data

The data used to train and assess the cascade classifier comes from 2 professional robotic soccer teams: The Dutch Nao team [13] and the SPQR team [14]. These are images of JPG and PNG format. A total of 5706 images were initially obtained and a selection is processed into 2726 positive and 1450 negative samples to serve as training data, see also figures 8 and 9. Positive samples are cropped images of the object of interest only. In this particular case it concerns a soft foam ball with a black and white soccer ball print. This is the official ball used in robotic soccer matches as stated in the official RoboCup Standard Platform League Rule Book[4]. In the set of positive samples, there are a variety of images selected in order to train the classifier more efficiently. These include

[3] `http://doc.aldebaran.com/2-1/family/robots/video_robot.html`

[4] `https://spl.robocup.org/wp-content/uploads/downloads/Rules2018.pdf`

convoluted images, blurred object within images due to movement of the corresponding objects and image taken from different perspectives. A critical note in the preparation of positive samples is the size of the image. These are all of a squared composition, thus having a x by x pixel format. The negative samples are images in which the object of interest is not present. This set also varies in images likewise in the set of positive samples. However, the focus is mostly set on the environment within and around a robotic soccer field. Including objects as robot soccer players, the goal and different backgrounds.

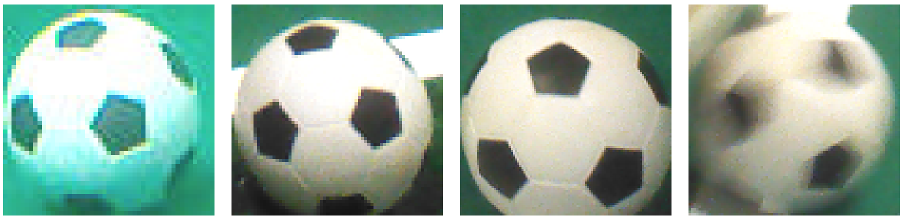


Figure 8: Positive samples

Figure 9: Negative samples

3.3 Classification Pipeline

Once the experimental data consisting of positive samples and negative samples is obtained, further preprocessing steps are needed in order to input those to the cascade classification system for training. An overview of these steps are given in a pipeline structure in figure 10 which is derived from [15]. The steps needed to process the negative samples are evident, as compared to the steps needed to taken for the positive samples. The only step needed for the negative samples is a listing in TXT format. After the cropping step which is earlier described, the set of positive samples need to be listed in a similar manner in TXT format, however this listing is only a stepping stone in order to convert the positives samples into a binary image file of VEC format. Thus having a representative file for each of the positve and the negative sample set, one in TXT format and the other in VEC format. Using OpenCV library, more positive samples are generated with the *opencv_createsamples* utility. The created samples include distorted samples by means of intensity tweaking and projectile translations. Furthermore the use of negative samples are used in order to act as backgrounds for the newly generated positive samples. Note that this feature is optional, but recommended as it would make the classifier more robust.

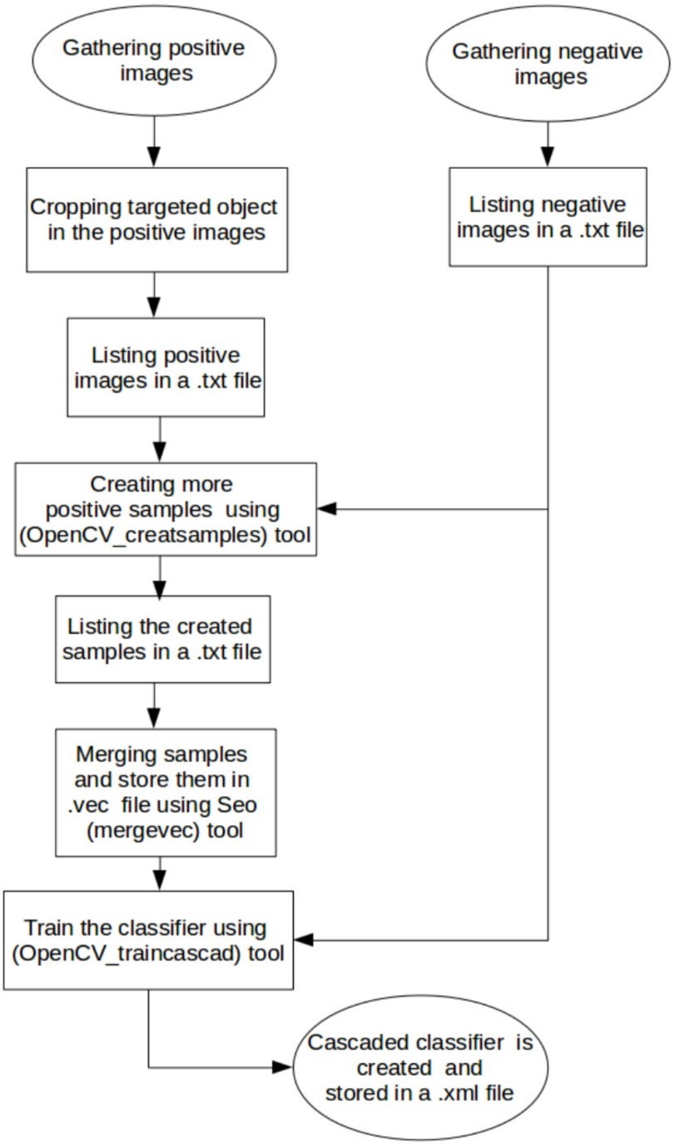


Figure 10: Pipeline for the classification system

3.4 Implementation

OpenCV library utility *opencv_traincascade* is used to train the cascade classifier. See table 1 for the used parameters. Note that the parameter *-w* and *-h* should have the exact same values as the image sizes of the set of positive samples. As seen in the pipeline structure, a cascaded classifier takes the format of a XML file. Using the SPQR team's framework[5] for ball detection and a technical report concerning the application of a machine learning classification system, an implementation of ball detection with the cascade classifier could be constructed. Note that the number of positive samples in the set has to be greater than the specified parameter when training the cascade classifier. This is because the classifier can incorrectly classify positive samples that are used in every stage of training. Thus rendering these pictures useless from the set. That is why it is important to have an overflow of positive samples in this manner. Used overflow in for this experiments is set on 100 images which proved to be sufficient for every trained stage.

Training parameter	Description	LBP	Haar
-data	Folder containing classifier	n/a	n/a
-vec	Positive samples in VEC format	n/a	n/a
-bg	Negative samples in TXT format	n/a	n/a
-numStages	amount of stages	1-20	1-20
-minHitRate	Minimum hit rate	0.999	0.999
-maxFalseAlarmRate	False Alarm Rate	0.5	0.5
-featureType	Type of algorithm	LBP	Haar
-numPos	Amount of positive samples	2626	2626
-numNeg	Amount of negative samples	1450	1450
-w	Positive sample width	16	16
-h	Positive sample height	16	16
-precalcIdxBufSize	Allocated memory: feature indices	512	512
-precalcValBufSize	Allocated memory: feature values	512	512

Table 1: Parameter settings

[5]`https://github.com/SPQRTeam/SPQRBallPerceptor`

4 Results

In this section, an overview of the results are given. Firstly, examples of positive detection on the ball are given in figure 11. It can be observed that the trained classification system draws a pink circular line on test images around the object it classifies as a ball. The ball is correctly detected in the test set involving a plain green soccer field, background of a Nao robotic soccer player, a blurred image of the ball and a situation of having a white background.

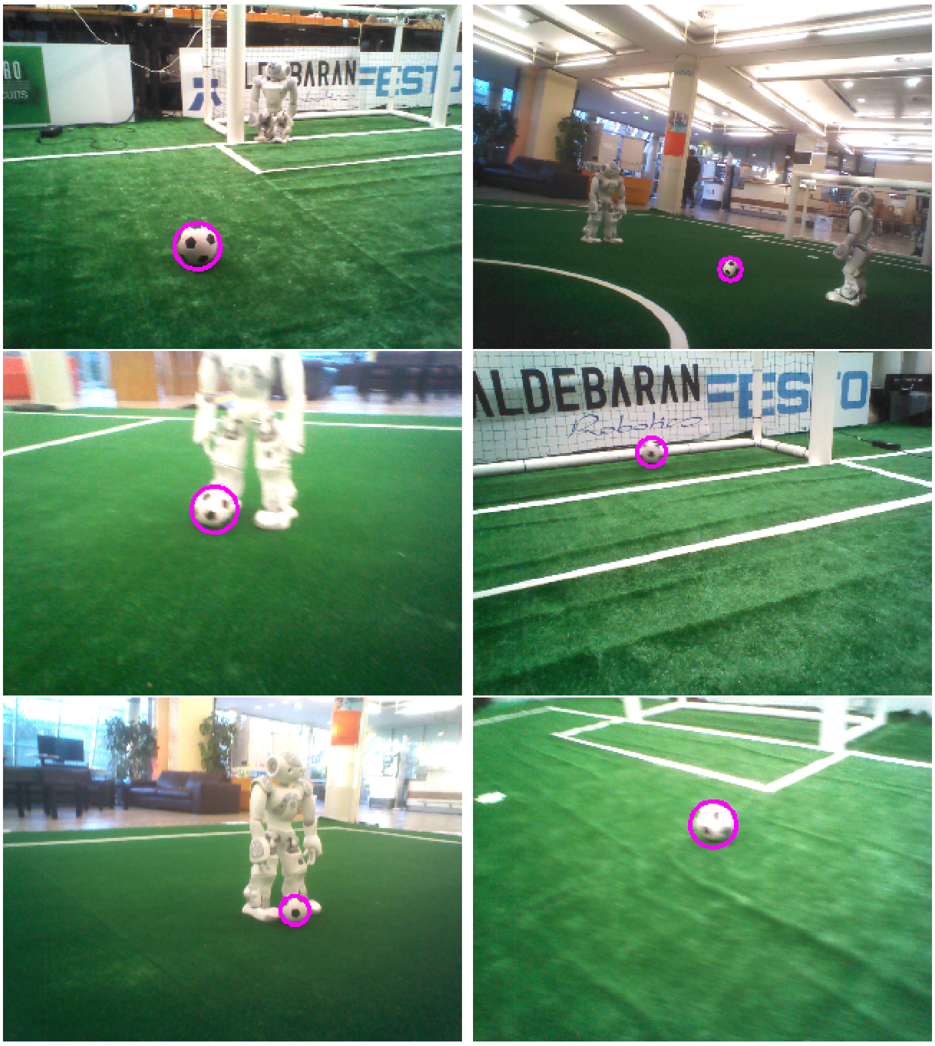


Figure 11: Accurate ball detection

Figure 12 shows examples of object detection involving false positives and the bottom 2 images depicting ball localization examples, unique to Haar-Like features. In regard to the false positives, it is observed that some images contain more false positives compared to others.In the middle row of figure 12, the image on the right-hand side is tested with the classifier using LBP till stage 10, whereas the image on the left-hand side is tested with the classifier using LBP till stage 17. Unique detection using Haar-Like Features in figure 12 are ball localizations that are either at close-quarter distance or at significant distance away. See figure 13 and 14 for the accuracy and training time results.

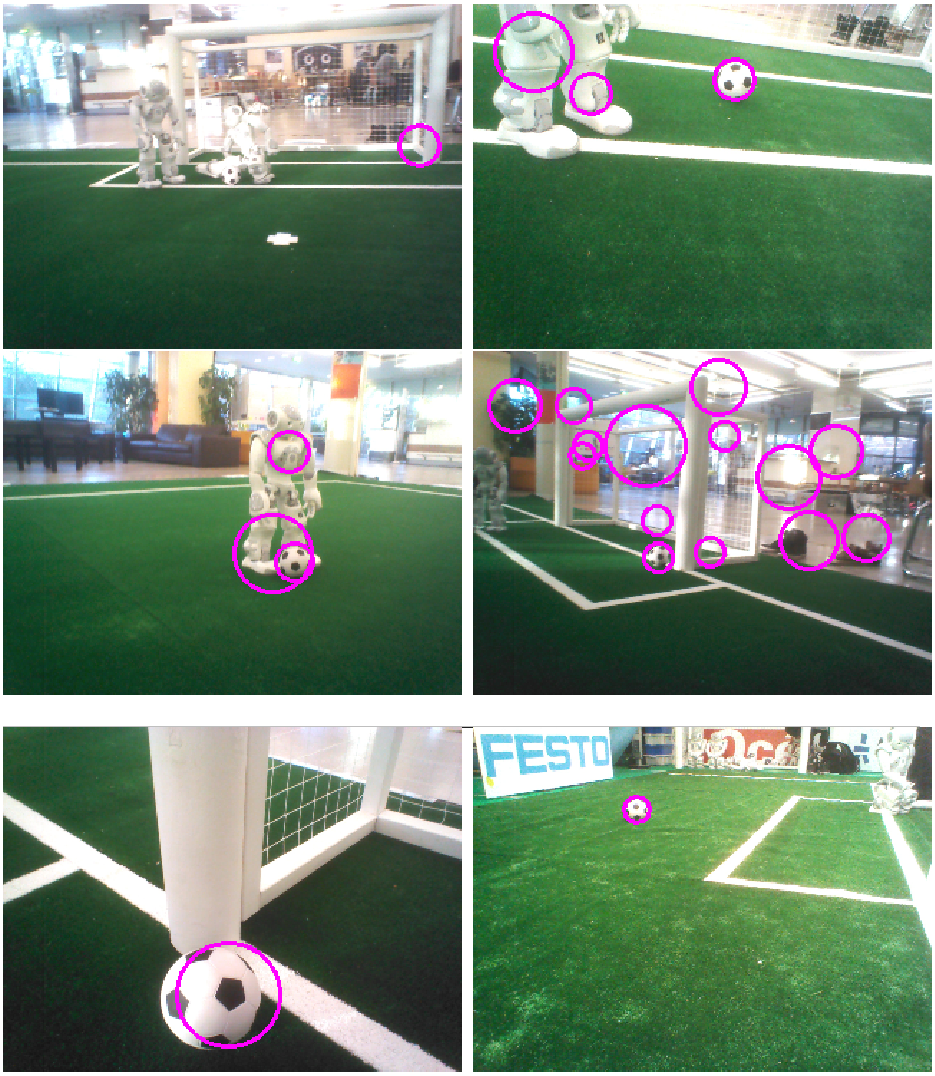


Figure 12: False positives + unique localization using Haar-Like Features

Figure 13: Algorithm Training time

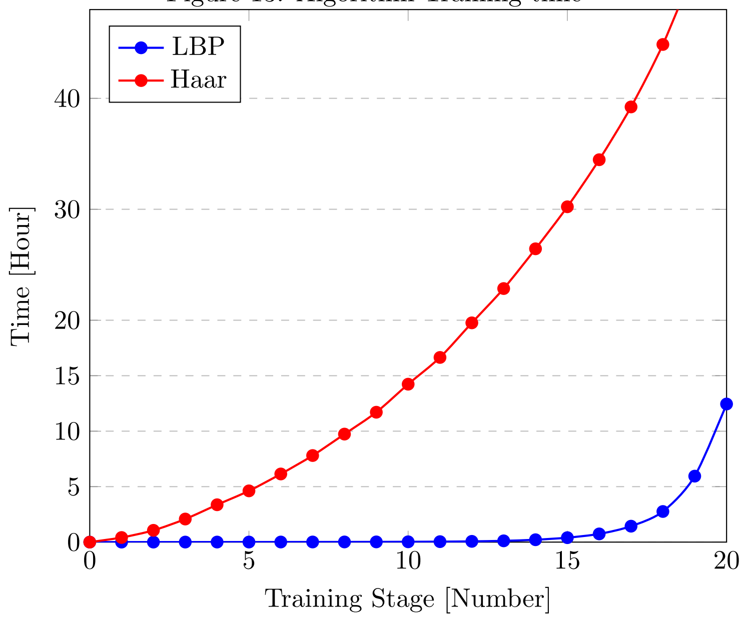
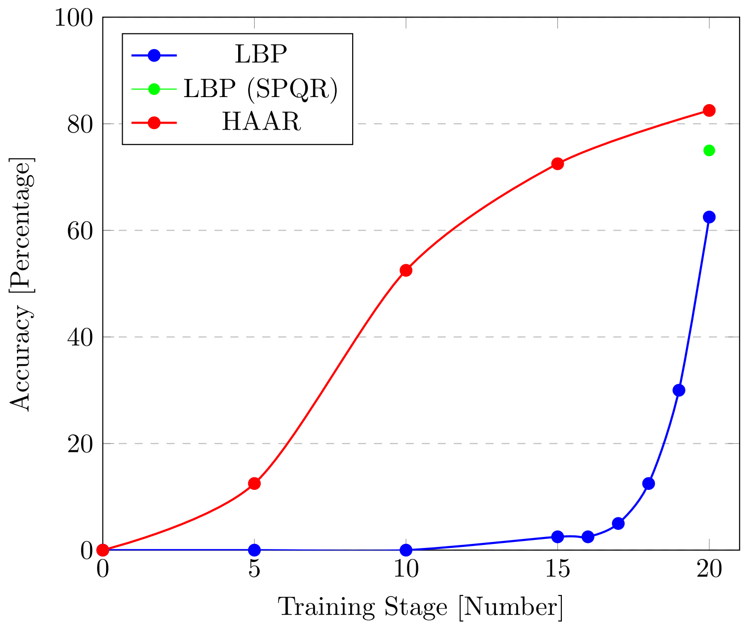


Figure 14: Algorithm Accuracy



5 Conclusion

After training and testing multiple classifiers using Local Binary Patterns and Haar-Like Features in a similar settings, it can be observed from the results that the Haar-Like Features take significant longer time to train. The Local Binary Patterns however, is more efficient when looking at it time-wise. Accuracy-wise, it can be observed from the results that the LBP is less accurate compared to Haar-Like Features. Concluding that the utilization of Local Binary patterns in professional robotic soccer matches is not up to the task by currents standards, as training can be done outside matches whereas accuracy has a direct effect within matches.

6 Discussion

As the feasibility is deemed not up to the task within this experiment, it displays significant potential compared to the more renowned Haar-Like Features in object detection. For one not familiar with the field of artificial intelligence, the result shown by the use the cascade classification system and LBP could be up to standard for ball detection. However, as the RoboCup goal pushes the boundary of robot capabilities concerning robotic soccer to its limit, it is not sufficient to this extent.

7 Future Research

Further training and testing the system using the algorithms could bring more insight in evaluating the feasibility of the use of Local Binary Patterns in robot soccer matches. Only the top camera of the nao robot has been trained in this experiment, which clearly put a disadvantage when at testing the system with Local Binary Patterns from up close. Haar-Like Features however, could detect instances of the ball at close distances. Thus constructing a framework in which both the top and bottom camera are integrated for training and testing could improve the use of Local Binary Patters relatively more, because the relative gain for Haar-Like features would be significantly less.

References

[1] S. Hedberg, "Robots playing soccer? robocup poses a new set of ai research challenges," *IEEE Expert*, vol. 12, no. 5, pp. 5–9, 1997.

[2] G. E. Ras, "Cognitive image processing for humanoid soccer in dynamic environments," *Maastricht University*, 2015.

[3] A. N. Fitriana, K. Mutijarsa, and W. Adiprawita, "Color-based segmentation and feature detection for ball and goal post on mobile soccer robot game field," *International Conference on Information Technology Systems and Innovation (ICITSI)*, pp. 1–4, 2016.

[4] S. Guennouni, A. Ahaitouf, and A. Mansouri, "A comparative study of multiple object detection using haar-like feature selection and local binary patterns in several platforms," *Modelling and Simulation in Engineering Journal*, pp. 365–374, 2015.

[5] T. Ojala, M. Pietikainen, and D. Harwood, "Performance evaluation of texture measures with classification based on kullback discrimination of distributions," *Proceedings of 12th International Conference on Pattern Recognition*, vol. 1, no. 1, pp. 582–585, 1994.

[6] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[7] S. Liao, X. Zhu, Z. Lei, L. Zhang, and S. Z. Li, "Learning multi-scale block local binary patterns for face recognition," *Lecture Notes in Computer Science*, vol. 4642, no. 1, pp. 828–837, 2007.

[8] S. Brahnam, L. C. Jain, L. Nanni, and A. Lumini, "Local binary patterns: New variants and applications," *Springer*, pp. 149–175, 2014.

[9] H. R. E. Doost and M. C. Amirani, "Texture classification with local binary pattern based on continues wavelet transformation," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, vol. 2, no. 10, 2013.

[10] B. P. Flannery, S. Teukolsky, W. H. Press, and W. T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, 2 ed., 1988.

[11] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, 2003.

[12] D. Gouaillier, V. Hugel, P. B. C. Kilner, J. Monceaux, P. Lafourcade, B. Marnier, J. Serre, and B. Maisonnier, "Mechatronic design of nao humanoid," *IEEE International Conference on Robotics and Automation*, vol. 1, no. 1, pp. 769–774, 2009.

[13] H. Lekanne gezegd Deprez, D. van der Wal, P. Kronemeijer, M. van der Meer, L. Petrini, T. Groot, J. Nzuanzu, and C. Lagrand, “Dutch nao team - technical report,” tech. rep., University of Amsterdam, December 2018.

[14] V. Suriani, F. Riccio, L. Iocchi, and D. Nardi, “Spqr robocup 2016 standard platform league qualification report,” December 2015.

[15] I. A. Alsayyari, “Real-time object detection on humanoid robots for the robocup soccer spl using cascaded classifiers,” Master’s thesis, University of Miami, 2017.