Department of Mechanical Engineering
Control Systems Technology

Bachelor's Thesis

Determining the Optimal Ball-tracking Method for an Automated Recording System

---

Author:
Josja Geijsberts
0896965

Supervisor:
dr.ir. René van de Molengraft

Coordinator:
Lotte de Koning

Eindhoven, July 2017

Abstract

In this thesis, various methods for ball-tracking are investigated with the aim of determining the most robust and accurate method to be used for an autonomous recording system. This recording system will be developed to automatize the recording of the RoboCup robotic soccer matches played by the Middle Size League (MSL) team of Tech United. Tech United is a robotics team of the Eindhoven University of Technology (TU/e). The methods will be compared based on a test dataset made on the TU/e RoboCup field, using a yellow ball and with robots on the field. On top of that, various yellow non-circular objects will be included in the test set in order to challenge the color dependence of the methods. Literature research results in four methods to be investigated and compared; HSV Filters Combined with Hough Circles, Haar-Feature Based Detection, Feature Based Detection using Local Binary Patterns and Deep Convolutional Neural Networks. HSV Filters Combined with Hough Circles and Feature Based Detection using Local Binary Patterns are outperformed by Haar-Feature Based Detection. Haar-Feature Based Detection provides the best results of all four tested methods based on detection rate, number of false positives and the detection time. Therefore, it is a very suitable method for accurate ball-detection. The results of the Convolutional Neural Network described in this thesis are trained on little data compared to Neural Network standards. This leads to a relative low detection rate and high number of false positives. The detection frequency is also very low. These complications result in the convolutional neural network to not be a suitable method for the ball detection as described in this thesis. These findings lead to the conclusion that Haar-Feature Based Detection is the most suitable method for real-time yellow ball-tracking.

Contents

Chapter 1

Introduction

Tech United is a multidisciplinary team of robotics-enthusiasts that competes in two RoboCup leagues, the @Home League and the Middle Size League (MSL) [1]. The research discussed in this thesis is related to the MSL soccer robot competitions.

All matches played by the soccer robots of the MSL team of Tech United are recorded for review and media purposes. This is currently a process manually executed and very time consuming and human-error sensitive. Therefore, an automated recording system that can autonomously record these soccer matches would be beneficial. Most of the relevant action occurs around the ball, hence the recording will be based on the location of the ball. The recording system will operate independently from the match itself, since that allows the system to be operational even if the robots no longer operate to their full extent, or if other potential irregularities occur during the match. Therefore, to enable autonomous recording, a ball-tracking method using computer vision will be developed. This thesis takes the first steps into developing an automated recording system by comparing various computer vision methods and determining the most suitable method for ball-tracking.

In this thesis, various methods for ball-tracking are investigated with the aim of determining the most robust and accurate method to be used by the autonomous recording system. A number of requirements are set up in order to enable a trustworthy comparison. The comparison will be based on a test dataset made at the TU/e RoboCup field, using a yellow ball and with robots and various yellow non-ball objects on the field. Since the detection should, eventually, operate in real-time, runtime is also investigated. These requirements are thoroughly described in Section 2.1.

Based on the described goal and requirements, the following hypothesis can be formulated:
What computer vision method is most suitable for detecting a yellow soccer ball on the TU/e field in real-time?

Firstly, the experimental setup and requirements are elaborated upon in Chapter 2. After this, research will be done into the various existing methods used for object detection, this is discussed in Chapter 3. Based on this research, a hypothesis will be formulated. Then, these methods will be implemented to detect the ball at the TU/e field, which is described in Chapter 4. The result of these implementations will be compared and this comparison will lead to a conclusion on what method is most suitable.

Chapter 2

Experimental Setup

This chapter elaborates on how the requirements used for the comparison of various ball-tracking methods are established. This chapter also discusses the experimental setup used to test these methods, to allow for this research to be reproduced.

2.1 Requirements

Given the scope of this project, the number of changing parameters is limited by testing only on the Eindhoven University of Technology (TU/e) RoboCup field, as adding multiple locations will drastically increase the size of the required training dataset. For example, by adding another environment to the test dataset, the size of the training dataset has to at least be doubled to achieve the same detection rate. Another requirement is that merely a yellow ball is to be detected, since the soccer robots only play with a yellow ball. This means there is currently no use for non-yellow ball-detection for the automated recording system. The automated recording system requires the detection method to be able to detect the ball at a high frequency. Based on the maximum velocity that the ball can reach, the size of the field and the position of the camera, a minimum detection frequency is established. This minimum value is merely a guideline and is not used as a strict requirement, since this value is very dependent on computer power and optimization of the computed vision methods. Analysis of camera footage provides an estimation of the smallest distance the full width of a frame can describe, this distance is estimated to be approximately four meters as shown in Figure 2.1.

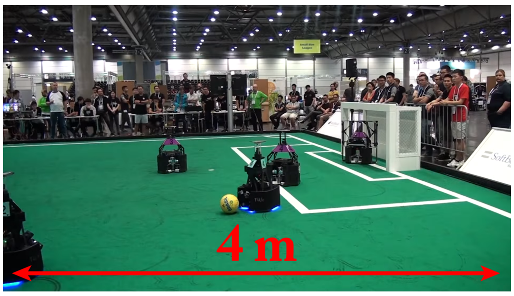


Figure 2.1: Visual explanation of the smallest distance the full width of a frame can describe.

The ball travels at a maximum speed of 12 m/s [2]. The camera will have to move fastest when the ball moves at its maximum speed across the smallest distance the full width of the frame can describe. Equation 2.1 shows the calculation of the minimum detection frequency:

$$f_{min} = \frac{v_{ball,max}}{s_{min}} = \frac{12}{4} = 3\ Hz \tag{2.1}$$

Since this value is an estimation, a safety margin of 2 Hz is applied, making the minimum detection frequency 5 Hz. For real-time object detection, a higher frequency than 5 Hz is desired, which can often be obtained using optimization methods. In order to challenge the color dependence of the methods, various yellow non-ball objects will be included in the test dataset.

2.2 Experimental Setup

All data used throughout this thesis is obtained at the TU/e RoboCup field and made through a frame by frame decomposition of videos made using a Panasonic HC-X920 video-camera. All methods are tested using a laptop of which the properties are shown in Table 2.1.

Property	**Value**
RAM	8 GB
CPU	Intel iCore i7-4700MQ
GPU	NVIDIA Quadro K610M

Table 2.1: Properties of used laptop.

A test dataset of 225 images is made with which all methods are tested and compared. These images include, as described in Section 2.1, various yellow non-ball objects and robots. Using these images, all methods are compared on a frame by frame basis. This means the methods are not dynamically tested. However, the runtime is analyzed for each frame and thereby also taken into account. Figure 2.2 shows four example images of the test dataset.

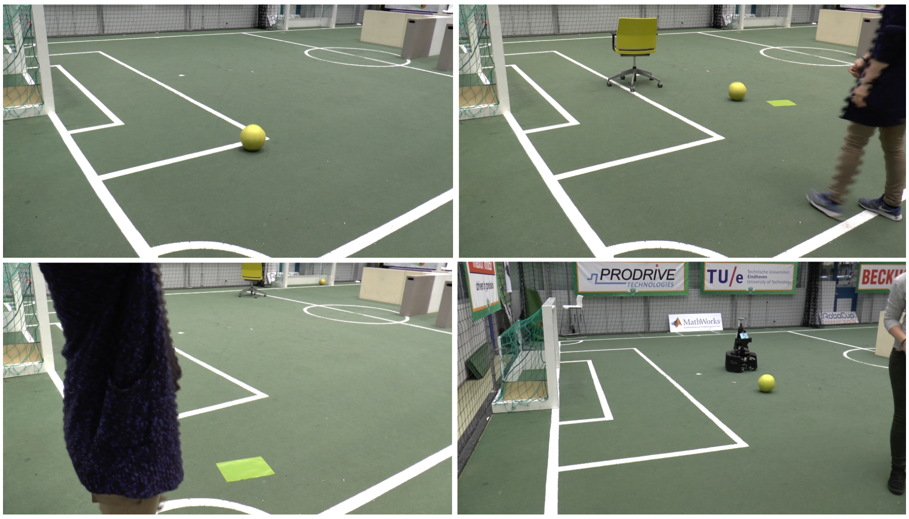


Figure 2.2: Example images of the test dataset.

Chapter 3

Related Work

There is a copious amount of literature available on object detection in computer vision. Based on this literature, the most robust and accurate methods are selected and discussed in this section.

3.1 HSV Filters Combined with Hough Circle Filter

The robots of the MSL team detect the ball based on its distinct color. A commonly used method for object detection very similar to the method used by the robots is color segmentation using a conversion of RGB space to Hue Saturation Value (HSV) space. To apply this method for round object detection, Hough Circle Transform can be applied to decrease the number of false positives [3]. The combination of a HSV filter and a Hough Circle Transform has proven very accurate when tested in the exact same environment as when calibrated [4] [5]. It is, however, very sensitive to changes, hence small variations already lead to non-detections. Since the ball will move over large distances and can move rather fast (12 m/s), the environment does change. In Chapter 4 it will be tested whether these changes are too large for this method to work properly.

3.2 Feature Based Detection using Haar-like features

One frequently used method for real-time object detection is detection based on Haar-like features. Haar-like features consider neighboring rectangular regions of an area in an image, and use the difference in pixel intensities to define that area [6]. A more extensive explanation of this method can be found in Appendix A. A commonly used toolkit for this method is the Open Source Computer Vision Library (OpenCV). OpenCV is a library containing programming functions that mainly focus on real-time computer vision [7]. The main reason for its popularity is the simplicity of its use. It should require little to no background knowledge to use OpenCV for fully functional object detection. On top of that, mainly due to its vast community, the library is updated very often, leading to large improvements. Aside from its simplicity, OpenCV has another advantage compared to alternative machine learning based detection methods. OpenCV Haar-feature-based cascade classifiers require a merely from around fifty to a few hundred positive input images [6]. The number of input images OpenCV requires depends mainly on the quality of the images, the complexity of the to be detected object and the method to generate the samples [10]. Although still a commonly used method, the use of haar feature-based classifiers has various downsides.

One of these downsides is that the computation cost of training is high; traditional techniques usually run in $O(NT\log(N))$, where N is the number of input images and T is the number of features used for classification [8]. There are various ways to overcome this high running time, but they work to the detriment of the simplicity of OpenCV and therefore discourage the use of Haar-like feature based detection. Another downside is that, although it yields a good performance in object detection, the number of false positives is high [9]. Increasing training time can decrease false positives, however, it also increases the computational cost and decreases the detection rate. Hence, when using Haar-like feature based cascaded classifiers, a balance has to be found between the number of allowed false positives and the desired detection rate. An example of the relations between false positives and detection rates can be found in Table 3.1 [11]. Since the ball has very pronounced features, this method is likely to be able to accurately detect the ball. The number of false positives, however, may be a problem. This will be investigated in Chapter 4.

	False detections							
Detector	10	31	50	65	78	95	167	422
Viola-Jones	76.1%	88.4%	91.4%	92.0%	92.1%	92.9%	93.9%	94.1%
Viola-Jones (voting)	81.1%	89.7%	92.1%	93.1%	93.1%	93.2%	93.7%	–
Rowley-Baluja-Kanade	83.2%	86.0%	–	–	–	89.2%	90.1%	89.9%

Table 3.1: Relation between false positives and detection rates (percentage) for various well-known classifiers described in literature [11].

3.3 Feature Based Detection using Local Binary Patterns

As mentioned in Section 3.2, the OpenCV community is extensive and active. In the previous section it was also mentioned that the computation cost of training a Haar-based classifier is high. Therefore, a different variation of the feature-based classifier has been developed; classification using Local Binary Patterns (LBP) [12]. The results of this method are very promising [13]. The concept of Local Binary Patterns is to summarize each pixel in an image by comparing it to its neighbors. Therefore all pixels obtain a relative value instead of an absolute value. It can be seen as texture-detection, which makes this method more robust in changing environments. The LBP method reduces the computational cost considerably compared to Haar classifiers [14] and is less sensitive to variations in environment. However, since the ball lacks a distinct texture, this method is unlikely to be sufficiently accurate for ball-detection.

3.4 Deep Convolutional Neural Networks

The concept of deep learning has become popular since it was proved that using multiple nonlinear processing layers can be used to successfully learn useful representations of features directly from data [15]. A more detailed explanation of the definition of neural networks can be found in Appendix B. When looking at computer vision, the most popular type of deep neural network is a Convolutional Neural Network (CNN). A CNN works by extracting features directly from images and the connectivity pattern between its neurons reduces the required memory during training, allowing for more layers which increases the accuracy. The complexity of a neural network also allows for a wider range of features to be learned, making it very resistant to changing environments [16]. However, there are various downsides to this method. A large, complex neural network consists of many parameters, which are what makes it accurate. A high number of parameters also leads to a high detection time which is incompatible with real-time detection. The size of the network is therefore a trade-off between accuracy and computation time. Another downside is that neural networks require a lot of data in order to obtain the high accuracy that is desired. On top of that, the training of a neural network is computationally very expensive. Nevertheless, this method is well known for its incredibly high accuracy and robustness. Therefore, if trained sufficiently, this method would be very suitable for accurate ball-detection but is likely to have an insufficiently high detection frequency.

3.5 Hypothesis

Based on the results obtained through the literature research a hypothesis can be formulated for the performance of these methods for ball-detection on the TU/e RoboCup field. This hypothesis is formulated as follows:
For detecting a yellow soccer ball on the TU/e RoboCup field, feature based detection using Haar-like features will best maintain accuracy whilst operating in real-time, and will therefore be the most suitable method.

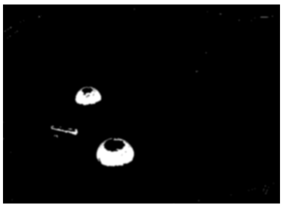
Chapter 4

Methods

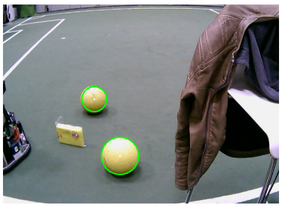
To test the hypothesis stated in Section 3.5, all methods described in Chapter 3 are implemented to detect the ball on the TU/e field. These methods can be optimized in various ways. That is, however, a very time-consuming process. Since the goal of this thesis is to determine the most suitable method, optimization is not a priority. This means that the results discussed in this chapter are not descriptions of the best capabilities of the discussed methods, but merely serve as a means for comparison. The results of these implementations are discussed in this chapter.

4.1 HSV Filters Combined with Hough Circles

For this method, the input image is separated into the three component images, i.e. hue, saturation and value. These separated images are then made into binary images. The intensity value of the binary images is then calibrated to detect the ball, after which the images are combined and only the pixels that have value 1 at all component images are given value 1. Through this method, only the areas that have the color of the ball have a value of 1. Then, a Hough Circle Transform is applies to filter out non-circular objects. The result of a calibration done on the TU/e field can be seen in Figure 4.1.



(a) Combined binary image.



(b) Final result.

Figure 4.1: Results of ball detection using HSV Filters and Hough Circle Filter.

As Figure 4.1 shows, this method can accurately detect the ball. However, once the ball starts to move and the lighting hits the ball slightly differently, the detection fails. The method also detects a high number of false positives. It does, however, operate under a very high frequency and has no problems with real-time image processing.

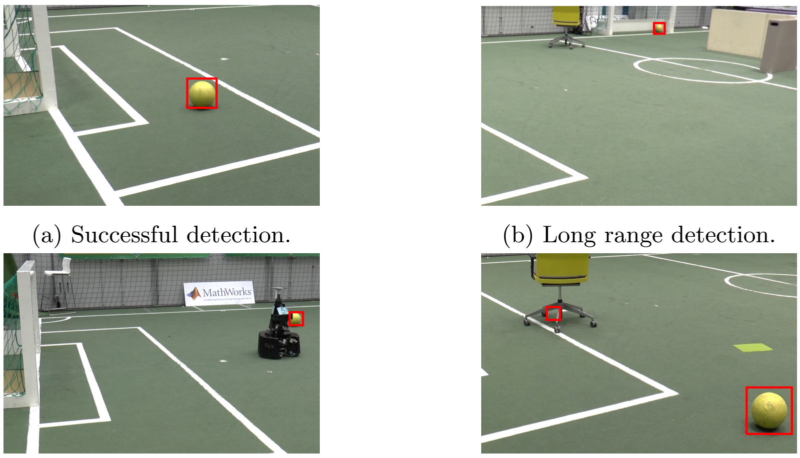
4.2 Feature Based Detection Based on Haar-like features

This method requires positive and background images. The positive images are cropped images of the to be detected object, all of similar dimensions. In this case, 182 positive images and 422 background images are used. Then, these images are used to create a larger set using a utility developed by Naotoshi Seo [17]. This utility creates multiple training samples from one image applying small distortions. The size of the dataset can thereby be increased. An example of a positive data image and the applied distortions can be found in Figure 4.2.

(a) Original positive image. (b) First created extra image. (c) Second created extra image.

Figure 4.2: Positive images, original and created through the application of small distortions.

There are many data sets available to use as background images, these sets merely have to be checked to ensure they do not include any object similar to the to be detected object. However, when using these sets, the result yields a high number of false positives. Therefore, a background data set was created at the TU/e field, this set is a set of pictures of the TU/e field in different lighting conditions and including robots and other objects that might be present on the field. Using 300 positive training images and 900 background images the result found in Figure 4.3 is obtained.



(a) Successful detection. (b) Long range detection.

(c) Partial occluded detection. (d) False detection.

Figure 4.3: Various detections using Haar-feature based classifiers.

The algorithm that uses the trained classifier for detection can be optimized in MinSize, MaxSize and MinNeigbors. MinSize and MaxSize describe respectively the minimum and maximum possible object size. MinNeighbors is a parameter that specifies how many neighboring candidate rectangles each candidate rectangle should have to retain it [18]. Giving MinNeighbors a high value leads to a decrease in false detections, but can also lead to an increase of non-detections. Since there are no non-detections, this value was given a rather high value, namely seven, to maximally decrease the number of false detections. This removed the false detection seen in Figure 4.3d and many more false detections, two of which are shown in Appendix C. To analyze the false detections, four different false detections, shown in Figure 4.4, are analyzed.

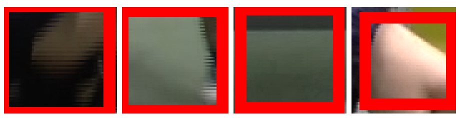


Figure 4.4: Four false detections using Haar-feature based detection.

All four false detections show a darker lower half with a lighter shape on the upper part. This contrast is a distinct feature of the ball due to the lighting on the field. This contrast is what Haar-like features are based on, so it is comprehensible why the method detects false positives on the positions shown in Figure 4.4. The algorithm runs on an average of approximately 7 Hz, which meets the requirements. The results are further discussed in Section 4.5 and 4.6.

4.3 Feature Based Detection using Local Binary Patterns

This method is very similar to the method described in Section 4.2, so the same data and functions are used. Merely the feature type is different. The results are better than expected, but there are more false detections and non detections than when using the method from Section 4.2. The runtime of the algorithm is approximately 6 Hz on average. This method meets the requirements and has sufficiently low false detections and non-detections for accurate real-time ball detection. The results are discussed in more detail in Section 4.5 and 4.6.

4.4 Deep Convolutional Neural Networks

The most crucial reason for not using this method is that it requires a lot of data. Another downside is that designing a neural network requires a lot of experience in the field of deep learning. Both of these obstacles can be solved by using a pre-trained neural network. The weights in a pre-trained network are not random, since the network has already been trained. This means that all weights are already closer to their optimal value, so less data is required to optimize the weights for a new object. An explanation of what these weights are can be found in Appendix B. The AlexNet is a network that has been trained to detect a soccer ball, so this net could be used to detect the ball. By triangulating what features are learned by this network, it is investigated whether this network can be used. The features trained after the last layer are visualized in Figure 4.5.

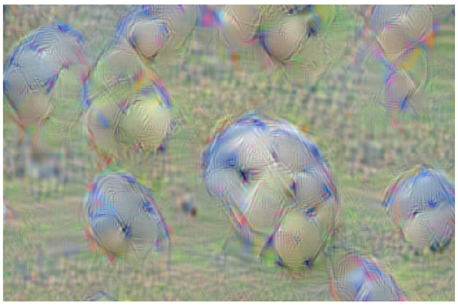
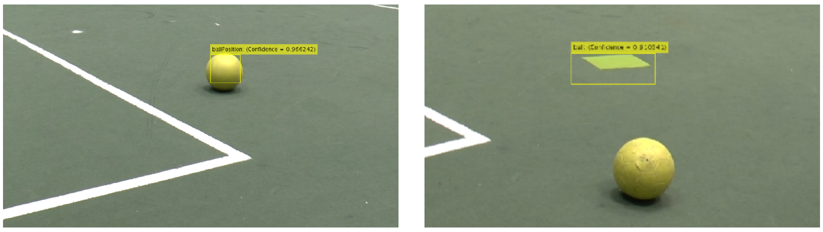


Figure 4.5: Features trained by the AlexNet visualized.

As can be seen in Figure 4.5, AlexNet is trained on a traditional soccer ball, with white and black hexa- and pentagons. Since these white and black shapes are a distinct feature used by the AlexNet network, as can be seen in Figure 4.5, this network will not properly detect a yellow soccer ball, lacking those distinct black and white shapes. Therefore a different network is used and retrained with the data of the correct ball. A fifteen-layered neural network is used that has been trained for detection using the CIFAR-10 dataset developed by Alex Krizhevsky and Geoffrey Hinton [19] [20]. A more elaborate explanation on this network can be found in Appendix B. This network is then retrained to detect merely two classes; ball objects and non-ball objects. This is done by following the Óbject Detection Using Deep Learning' example of the Matlab Computer Vision Toolbox [21]. The network is retrained using a labeled ground truth dataset. This set, initially, has 83 images. As Figure 4.6 shows that the network detects the ball, but once yellow objects are placed on the field those are also detected by the network.

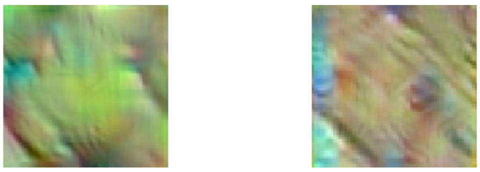


(a) Successful detection, confidence: 0.966242.

(b) False positive in a yellow object, confidence: 0.910541.

Figure 4.6: Results of ball detection using a neural network with accompanying confidence.

When investigating what the features that the network has learned look like, an explanation is found. As can be seen in Figure 4.7, the trained features do not show distinct circular objects. The yellow color is visible, but the circular shaped object, as seen in Figure 4.5, is not visible. This clarifies why the network also detects other yellow object as if they are a ball. To analyze the influence of more data and to improve the performance of the network, a larger dataset of 232 images is made and the network is retrained using the new dataset. The result of the trained features can be seen in Figure 4.7b. Figure 4.7 shows the improvement of the network upon increasing the size of the training dataset. Figure 4.7a shows the features learned by the network trained on 83 images. Those features include more shades of green than Figure 4.7b which represents the features learned by the network trained on 232 images. The decrease in shades of green seen when increasing the size of the training dataset, shows that increasing the size of the training dataset results in more representative features.



(a) Network trained on 83 positive images.

(b) Network trained on 232 positive images.

Figure 4.7: Visualization of trained features.

The network also yields a high number of non-detections, and has a high detection frequency; 0.3 Hz on average. The result of the network trained on more data is better, but there are still many non-detections and false positives. The detection time of the neural network remains the same; 0.3 Hz on average. However, both these complications were improved when increasing the size of the dataset. On top of that, neural networks provide a confidence for each detection which represents how certain the network is of its detection, as can be seen in Figure 4.6. Using this value, false detections can be filtered. By using the assumption that there can only be a single ball on the field, the detections are filtered on the highest confidence value. This leads to a significant decrease in false positives which can be seen in Section 4.5. It, however, also lowers the detection rate, since if another object has a higher confidence value, the detection of the ball is filtered out, this is shown in Figure C.2 in Appendix C. The filter does not influence runtime.

4.5 Comparison

The lack of robustness and accuracy of the method described in Section 4.1 leads to the conclusion that this method can not be used for real-time ball detection and is therefore not taken into consideration. An overview of the important resulting parameters of the remaining three methods are shown in Table 4.1.

Detector	Detection rate	False positives	Average frequency [Hz]
Haar-Feature Based Detection	94.29%	12	6.6
LBP-Feature Based Detection	93.33%	19	5.6
Neural Network trained on 83 images	53.78% / 34.32%*	217 / 42*	0.3
Neural Network trained on 232 images	64.88% / 47.26%*	164 / 36*	0.3

Table 4.1: Overview of the relevant results of the various methods. (*) represents the use of the confidence filter.

The significance of the runtime, or detection time, is debatable since this can easily be influenced with more computer power. Although this does not overcome the difference in detection time, it does decrease relevance of the outcome of this parameter. There is, however, a considerable difference in detection rate. Increasing the size of the training dataset leads to an improvement in the detection rate of the neural network, as can be seen in Table 4.1.

4.6 Discussion

HSV Filters Combined with Hough Circles is a method that can successfully detect the ball, but is highly sensitive to small changes and therefore not sufficiently robust for accurate ball-detection. The result of the implementation of Feature Based Detection using Local Binary Patterns, as shown in Section 4.3 and 4.5, shows that the method performs nearly as well as Haar-Feature Based Detection. The main advantage of using LBP is that training is significantly faster, which does not influence the quality of the classifier. And since the concepts are based on the same principle, the errors that occur, i.e., the false detections and non-detections, occur during the same scenarios. Feature Based Detection using Local Binary Patterns therefore has no significant advantages compared to Haar-Feature Based Detection.

Haar-Feature Based Detection is a very promising method, with a detection rate of 94.29 % and merely twelve false positives in a test-dataset of 225 images it is a very suitable method for accurate ball-detection. On top of that, the detection time is also meets the requirements. Haar-feature based detection can be optimized by implementing a region of interest. A region of interest is often based on the detection of previous frames. Based on these previous detections, an expectation can be formulated on where the object will be in the next frame. This expectation can be used as a region of interest where the algorithm can start its search, instead of scanning the whole frame. This form of optimization can further increase the detection rate and frequency and decrease the number of false detections.

Convolutional Neural Networks are praised in literature as the miracle method that solves all computer vision problems. It is, however, a very complex and time consuming method. Neural Networks require a lot of data to obtain the accuracy they are praised for, and the size of the network, which is also required for the accuracy, leads to a very high detection time. The results of the network described in Section 4.4 and 4.5, are trained on little data compared to Neural Network standards. This leads to a relative low detection rate and high number of false positives. The detection frequency is also very low; 0.3 Hz on average using a network trained on 232 images. The number of false positives can be drastically decreased by implementing a confidence filter, but this also decreases the already low detection rate. Similar to Haar-Feature Based Detection, Convolutional Neural Networks can be optimized. Table 4.1 in Section 4.5 shows that increasing the size of the dataset used for training, improves the results. Therefore, the performance of the network can be optimized by increasing the size of the dataset. Using transfer learning, an increase to 1000 images will significantly improve performance. Additionally, a more compact network with less layers, i.e., less than 10, could be developed to decrease detection time. However, in that case, even more data is required since a newly developed network, ergo not pre-trained, deals with random weights and demands more data to accurately determine the weights.

Chapter 5

Conclusions

The goal of this thesis is to determine the most robust and accurate method to be used by a recording system to autonomously record robot soccer matches of the MSL team of Tech United. The literature research summarized in Chapter 3, resulted in four methods to be investigated and compared; HSV Filters Combined with Hough Circles, Haar-Feature Based Detection, Feature Based Detection using Local Binary Patterns and Deep Convolutional Neural Networks. Based on this literature research, the following hypothesis was formulated:
For detecting a yellow soccer ball on the TU/e RoboCup field, feature based detection using Haar-like features will best maintain accuracy whilst operating in real-time, and will therefore be the most suitable method.
All four methods have been implemented to test this hypothesis, this process is described in Chapter 4.

HSV Filters Combined with Hough Circles is a method that is not sufficiently robust for accurate ball-detection. Feature Based Detection using Local Binary Patterns was not expected to be a successful method, as described in Section 3.3. Although it, surprisingly, performs nearly as well as Haar-Feature Based Detection, it has no significant advantages compared to Haar-Feature Based Detection and is therefore not the most suitable method for real time ball-tacking.

Haar-Feature Based Detection is the method that, according to the hypothesis, would be most suitable for accurate real-time ball-detection. The results of the implementation, as shown in Section 4.2, provide the best results of all four tested methods based on detection rate, number of false positives and the detection time.

Convolutional Neural Networks, as described in Section 3.4, are well known for their incredibly high accuracy and robustness. Therefore, based on related work, Convolutional Neural Networks were expected to be very suitable for accurate ball-detection, if trained sufficiently. However, a significant problem was also identified; the detection frequency. The detection frequency was not expected to be sufficiently high for real-time detection. The results provided a detection frequency of approximately $\frac{1}{3}$ Hz, so thereby the network proved to be too slow to meet the requirements set up in Chapter 2. Additionally, mainly due to the insufficiently large training dataset, the detection rate and number of false positives are not compatible with accurate ball-detection. These complications result in the belief that a convolutional neural network is not a suitable method for the ball detection as required for the automated recording system.

These findings lead to the conclusion that Haar-Feature Based Detection is the most suitable method for yellow ball-tracking on the TU/e RoboCup field.

5.1 Future work

The goal of this thesis, determining the most robust and accurate method for ball-detection, is set with the purpose of finding a reliable method to be used by a recording system to autonomously record robot soccer matches. As mentioned in Chapter 5, optimization is not a priority in this thesis. However, the most suitable method, Haar-feature based detection, can be optimized through for example implementing a region of interest, as explained in Section 4.6. Applying this region of interest can lead to an increase in detection rate and frequency and a decrease in false detections, and would therefore be a very beneficial next step.

To enable a trustworthy comparison, various requirements have been set up, as explained in Chapter 2. The most influential requirement is that all methods are tested and compared based on a test dataset made at the TU/e RoboCup field. A trained classifier based on Haar-like features is very robust when trained for a specific setting, for example, the TU/e field. However, when this method is to be used in varying environments, such as different fields at different tournaments, this robustness is lost. A neural network, on the other hand, is very resistant to changing environments, if trained with sufficient and varying data. Therefore, a neural network is more suitable when looking at the purpose of developing an autonomous recording system that can be used during tournaments.

The convolutional neural network described in this thesis will not suffice for real-time ball-detection as shown in Section 4.5. Since a neural network is more likely to be sufficiently accurate in changing environments, various changes have to be made to the tested neural network for it to be used as an accurate detector. First of all, a larger dataset is required. Using the concept of transfer learning, 1000 images is a good indication for training an accurate detector. However, to ensure its robustness in changing environments, a dataset with varying data of approximately 10000 images made at different fields should be used. The issue of the insufficiently high detection frequency can be solved by increasing the computation power or decreasing the size of the network.

Once the detector is optimized to successfully detect the ball in changing settings, an algorithm should be developed that translates the position of the ball in the frame to the desired movement of the camera. On top of that, a controller has to be developed to ensure smooth movement of the camera. In addition to that, a link could be made between the recording system and the Referee Box(Refbox) to enable automated selection of recorded video.

Bibliography

[1] About us (2015). In *Tech United*. Retrieved on 15-06-2017.
http://www.techunited.nl/en/about-us

[2] Tech United Eindhoven (2016). Winner RoboCup 2016 MSL: Middle Size League.

[3] Hough Circle Transform (2016). In *OpenCV Documentation*. Retrieved on 15-04-2017.
http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_imgproc/py_houghcircles/py_houghcircles.html

[4] Ball Tracking and Detection using OpenCV (2012). In *Blog Aniket Tatipamula*. Retrieved on 15-04-2017. http://anikettatipamula.blogspot.nl/2012/12/ball-tracking-detection-using-opencv.html

[5] Jin, T., Xiong, L., Bin, X., Fangyan, C. and Bo, L. (2010). A method for traffic signs detection, tracking and recognition. *International Conference on Computer Science & Education*, 5, p. 189-194.

[6] Haar Feature-based Cascade Classifier for Object Detection (2014). In *OpenCV Documentation*. Retrieved on 15-04-2017.
http://docs.opencv.org/2.4/modules/objdetect/doc/cascade_classification.html#haar-feature-based-cascade-classifier-for-object-detection

[7] Pulli, K., Baksheev, A., Kornyakov, K. and Eruhimov, V. (2012). Real-time computer vision with OpenCV. *Communications of the ACM*, 55(6), p. 61-69.

[8] Pham, M. T. and Cham, T. J. (2007). Fast training and selection of Haar features using statistics in boosting-based face detection. *Computer Vision IEEE 11th International Conference*, 11, p. 1-7.

[9] Zheng, K., Zhao, Y., Gu, J. and Hu, Q. (2012). License plate detection using haar-like features and histogram of oriented gradients. *Industrial Electronics IEEE International Symposium*, p. 1502-1505.

[10] Train Your Own OpenCV Haar Classifier (2013). In *Coding Robin*. Retrieved on 15-03-2017.
http://coding-robin.de/2013/07/22/train-your-own-opencv-haar-classifier.html

[11] Viola, P. and Jones, M. J. (2004). Robust real-time face detection. *International journal of computer vision*, 57(2), p. 137-154.

[12] Local Binary Patterns Histograms (2017). In *OpenCV Documentation*. Retrieved on 15-04-2017.
http://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html

[13] Ahonen, T., Hadid, A. and Pietikainen, M. (2006). Face description with local binary patterns: Application to face recognition. *IEEE transactions on pattern analysis and machine intelligence*, 28(12), p. 2037-2041.

[14] Bilaniuk, O., Fazl-Ersi, E., Laganiere, R., Xu, C., Laroche, D. and Moulder, C. (2014). Fast LBP face detection on low-power SIMD architectures. *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, p. 616-622).

[15] LeCun, Y., Bengio, Y. and Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444.

[16] Krizhevsky, A., Sutskever, I. and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, p. 1097-1105.

[17] Tutorial: OpenCV haartraining(2008). In *Naotoshi Seo's Note Site*. Retrieved on 1-05-2017. `http://note.sonots.com/SciSoftware/haartraining.html`

[18] Cascade Classifier Class Reference(2017). In *OpenCV Documentation*. Retrieved on 3-05-2017.
`http://docs.opencv.org/trunk/d1/de5/classcv_1_1CascadeClassifier.html`

[19] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

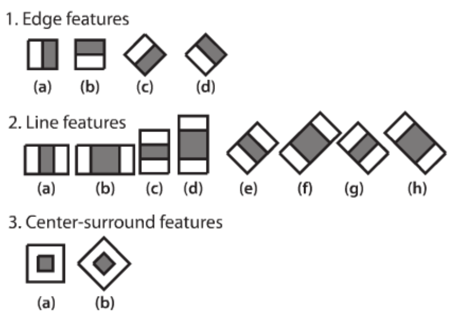
[20] Krizhevsky, A. and Hinton, G. (2009). Learning multiple layers of features from tiny images.

[21] Object Detection Using Deep Learning(2017). In *MathWorks Support*. Retrieved on 9-06-2017. `https://nl.mathworks.com/help/vision/examples/object-detection-using-deep-learning.html`

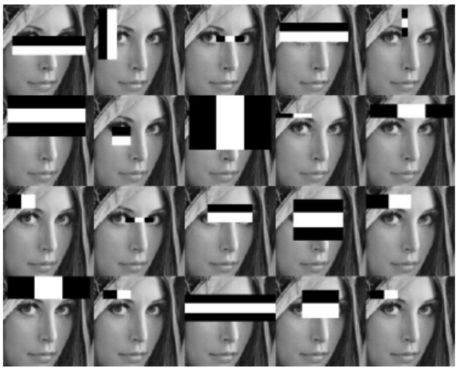
Appendix A

Clarification Haar Features

A Haar-like feature describes the difference in pixel intensity of neighboring rectangular regions. This difference thereby describes a contrast in intensity, as can be seen in Figure A.1a. These various features are then used to describe an object. By scanning an input image for those features, the object can be detected. Figure A.1b shows how these features are used to describe an object, in this case, a face.



(a) Definition of Haar-like features.



(b) Trained Haar features example.

Figure A.1: Clarification of Haar-like feature detection.

Appendix B

Clarification Neural Networks

B.1 The Concept of Neural Networks

Neural networks in computer vision are inspired by biological neural networks found in brains. Neural networks progressively improve performance by analyzing a large dataset of labeled examples. Neural networks are a collection of neurons that are organized in layers, these layers perform different kinds of transformations on their input. The neurons are then connected through synapses. These synapses carry weights that represent the strength of the dependence of the connected neuron to the previous neuron. By training a network, these weights are optimized to perform the desired task.

B.2 The Used Neural Network

A 15 layered network proposed by the Mathworks example is used. The architecture of the layers in the network can be found in Table B.1.

Layer	Layer Type
1	Image Input
2	Convolution
3	ReLu
4	Max Pooling
5	Convolution
6	ReLu
7	Max Pooling
8	Convolution
9	ReLu
10	Max Pooling
11	Fully Connected
12	ReLu
13	Fully Connected
14	Softmax
15	Classification Output

Table B.1: Description of the used network for ball detection.

The input layer describes data the neural network can process. The middle layers consist of repeated sets of convolutional, ReLU (rectified linear units), and pooling layers. The convolutional layers define sets of filter weights, which are updated when training the network. The ReLU layers add non-linearity to the network. The pooling layers downsample data as it flows through the network. The final layers use the output of the fully connected layer to determine the probability distribution over the various classes [21].

Appendix C

Results

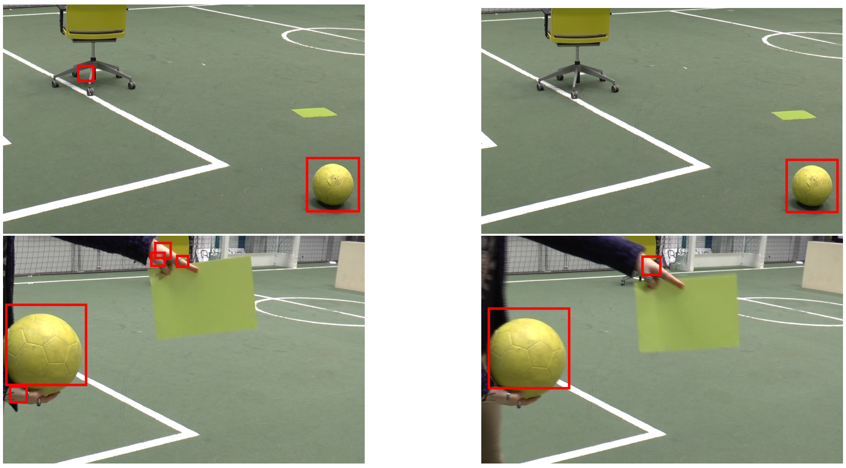
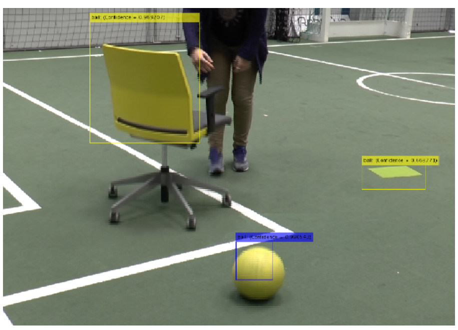


Figure C.1: Solving false detections by increasing value for MinNeighbors.



(a) Confidence; chair:0.969207, ball:0.990543, sheet:0.663771.

(b) Confidence; chair:0.590277, ball:0.669937, sheet:0.704173.

Figure C.2: Filtering based on confidence does not always lead to improvement, due to the ball not always having the highest confidence value.