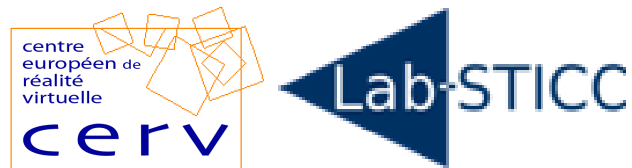




MASTER RESEARCH INTERNSHIP



MASTER THESIS

Hyperion

Robot detection and localization by machine learning and computer vision algorithms

Domain : Machine Learning - Artificial Intelligence - Computer Vision

Author:
Andreea-Oana PETAC

Supervisors:

Cédric BUCHE¹
Fabrice HARROUET¹
Panagiotis PAPADAKIS²
Mihai POLCEANU¹

1. IHSEV - Lab-STICC - France
2. IMT Atlantique - France



June 3rd 2017

Abstract

The topic of my research project is the study of various methods for image-based object detection and object localization. As an application to this broad area, it is proposed that the algorithms be tested on NAO robots, in the context of an international artificial intelligence and autonomous robotics contest. The robots must be able to detect each other within the field, as well as track their teammates' and enemies' movements. In addition to this, the robots should be able to identify the ball and various landmarks both within and outside the field. The research question resides in whether the objects and robots within the field can be detected and localized, taking into consideration all the encountered constraints. Therefore, the purpose is to find the best possible solution with the help of machine learning and computer vision algorithms, having in mind the limited resources such as the CPU, time, real-time. In this work, different methods from the machine learning and computer vision areas are announced and compared. It is also presented and evaluated the proposed solution to the problem.

Contents

1	Introduction	2
1.1	Context	2
1.2	Problem specification	2
1.3	Research question	3
1.4	Report outline	4
2	State of the Art	4
2.1	The learning process	4
2.2	Feature extraction methods	5
2.2.1	Histogram of oriented gradients	5
2.2.2	Haar-like features	6
2.2.3	Discrete wavelet transformation	6
2.2.4	Object detection by regression	6
2.2.5	Speeded-up robust feature	7
2.2.6	Scale invariant feature transform	7
2.2.7	Principal component analysis	8
2.3	Feature extraction comparison	8
2.4	Bag of Words model	9
2.5	Classification and localization methods	10
2.5.1	Support vector machines	10
2.5.2	k-Nearest Neighbour	10
2.5.3	AdaBoost	11
2.5.4	Simultaneous localization and mapping	11
2.5.5	Particle filter	12
2.6	Classification and localization methods comparison	12
3	Contribution	13
3.1	Proposition	13
3.2	System description	14

4	Evaluation	17
5	Conclusion	25
5.1	Synthesis	25
5.2	Limitations	26
5.3	Future work	27

1 Introduction

1.1 Context

The present master thesis showcases the work I undertook during my master research internship that was conducted at the Centre Européen de Réalité Virtuelle (CERV). This internship was in collaboration with the RoboCanes team [Masterjohn et al., 2015] from the University of Miami (UM).

1.2 Problem specification

NAO robot The NAO robot represents the platform selected for our research. It is an autonomous, humanoid robot produced by the Japanese company SoftBank Robotics SAS (the former Aldebaran Robotics). NAO is 58 cm tall and weights 4.3 kg. It has a built-in Linux based operating system (NAOqi OS [Naoqi, 2016]), making it a completely programmable and interactive robot. Starting from 2007, NAOs have been chosen to be used in the Robot Soccer World Cup (RoboCup), an annual international robot soccer competition. Our main focus is the RoboCup SPL (Standard Platform League), whose aim is to encourage researchers and students to work on robotics and AI challenges, developing more and more efficient approaches in order to solve the problems faced by autonomous robots [Kitano et al., 1997]. For example, the robot's balance when moving.

SPL contest Within this RoboCup league, two teams of 5 robots each play against each other a robot soccer game. During a typical SPL match, the robots recognize a number of different objects situated not only within the football field but also outside of it. Apart from the other NAOs, humans may also be present on the field, for example referees. As a result, the robot must know what happens in its vicinity and must see the other objects within the field in order to be able to localize itself with respect to the other objects. The robot will utilize the detection in order to do the localization. However, the robot is not interested in where it is located from the global point of view, but rather it is required to determine its position with respect to surrounding objects such as people, obstacles, etc.

Constraints Within this context, some constraints must be taken into consideration. Because there are various hardware and software specific issues that need to be addressed within this project, we need to identify the constraints so that the encountered limitations can be overcome. To start with, the latest version of NAO robots have a **1.6 GHz CPU**, which raises the challenge to develop algorithms with the lowest possible computational cost [Niemüller et al., 2011].

It must also be taken into consideration that the robot needs to split its **limited processing power** (Intel Atom 1,6 GHz (V4)) between various tasks such as vision, localization, behavior and motion control and at the same time while remaining highly responsive enough to play soccer [Khandelwal et al., 2010]. Thus, the **detection process should not take more than 20 ms/cycle**.

Each NAO robot (see Figure 1) is equipped with 2 cameras located along the center of its face at different angular offsets with a **maximum resolution of 1280 x 720 pixels**. Moreover, during the game, **the robots are constantly moving**. Due to the initial manual camera calibration, every unexpected change in the camera settings requires a readjustment of the stable conditions.

As a result, the NAO vision system is **not very robust to illumination variations**, that may have a strong influence on the performance of the algorithms.

Another constraint is easily drawn up by taking into consideration that **the code runs on board the robot**. Therefore, it is advised to have a static training dataset, meaning that the learning should be done exclusively offline.

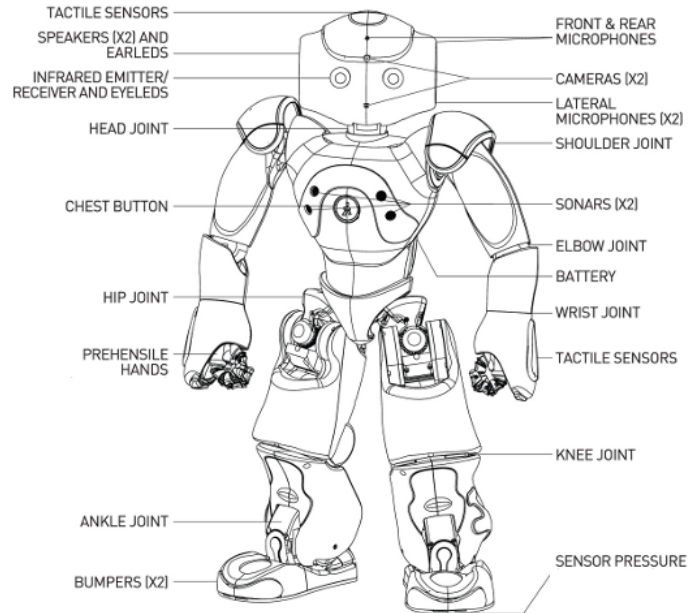


Figure 1: Overview of NAO’s hardware parts [Naoqi, 2016]

We can deduce that the objective of the project is to analyze the various methods for object detection and localization, and to implement them within the RoboCup contest. Among these methods, and after testing and optimization, the goal is to find the best possible solution that respects all the constraints presented above.

1.3 Research question

Visual object detection and tracking plays an important role within the RoboCup contest. The challenge is to focus and implement on less sensitive methods to image rotation, scaling and illumination progressions. The purpose of object tracking is to continuously focus the position of the object of interest throughout the frames. In order to obtain the tracking, the other elements within the scenes must be ignored.

To conclude this subsection, the research question is placed within the context of whether the robots within the field can be detected and tracked by making use of machine learning and computer vision algorithms.

1.4 Report outline

The research question is focused on the detection and localization of robots and landmarks within the field. It is crucial that the constraints be taken into consideration. Thus, a good balance should be found between good results and low computational cost.

In order to localize itself, the robot must first detect the objects and landmarks from its point of view. Because there is a high number of inputs, it is important to be chosen the most discriminative ones within the dataset in order for the results to be accurate. We need to use the feature extraction because the classification techniques may need extracted values. Briefly, through feature extraction it is understood a method that establishes accurate combinations of traits. This will end by an improvement related to the results. The extracted features will be used in combination with the classification methods. Next, by using a learning algorithm, an object detection and localization could be done.

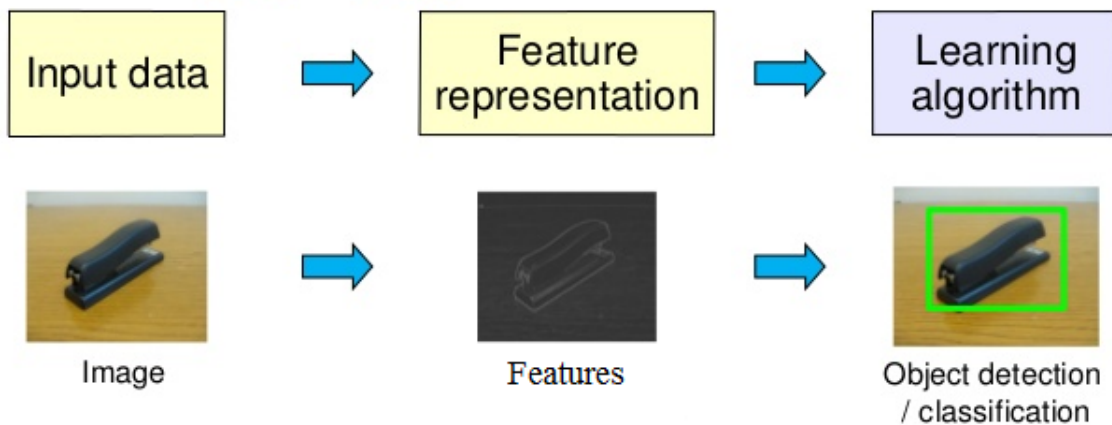


Figure 2: Overview of the learning process by making use of feature extraction

2 State of the Art

In order to achieve our goal, we first have to train a model to find features, namely specific structures in the image such as points, edges or objects within the football field. The action of extracting the features amounts to reducing the total amount of data needed to characterize a large dataset.

2.1 The learning process

Within the following diagram (Figure 3) presented the processes of training and prediction are presented. In the example shown below, a picture serves as input to machine learning models. Each machine learning model makes use of an algorithm, depending also on the dataset. Assuming that there will be a high number of inputs, it is important to extract those that have distinctive characteristics.

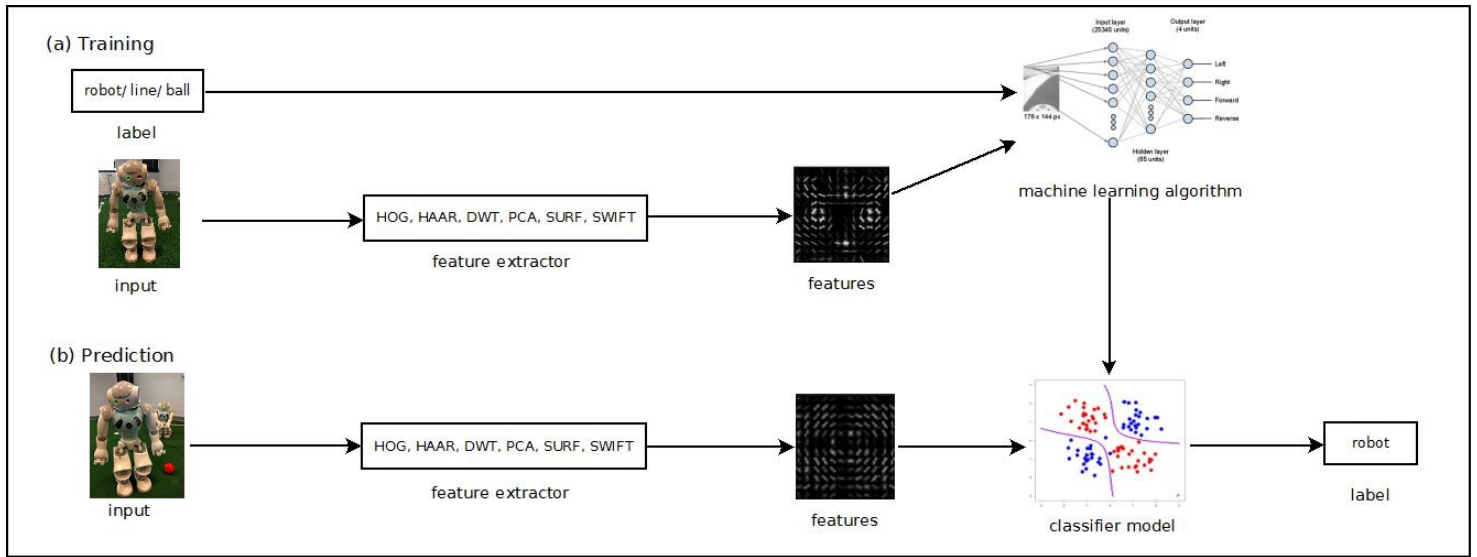


Figure 3: Learning diagram (adapted from [Bird et al., 2009])

The purpose of a feature extractor during the training phase is to convert each input value to a set of features. Within these sets of features, the essential information about each input is grabbed that is going to be used to classify it. Next, pairs formed by feature sets and their corresponding labels are provided to the machine learning algorithm, a model being then generated.

As far as machine learning is concerned, it is very important for the right features to be chosen and the right way to represent them, this having a major impact on the ability of the learning method to extract a good model.

Regarding the prediction phase, the trained model is used in order to predict labels of unseen inputs. By using the same feature extractor algorithm, the unseen inputs are converted to feature sets. After these feature sets pass through a classifier model, the predicted labels are generated.

2.2 Feature extraction methods

The extraction of the features is done mainly because various extracted values through the classification techniques may enhance their performance. By performing this action, there are several advantages such as invariance provided to transformations, noise filtering and a capture of the discriminant.

2.2.1 Histogram of oriented gradients

This is one of the most popular techniques to retrieve shapes from an image. The histogram of oriented gradients (HOG) extracts features within all locations in the image, or within the area of interest. This feature extraction can be used in order to detect shapes of any kind [Dalal and Triggs, 2005]. For example, within some of the classic examples are included the circle detection [Skibbe and Reisert, 2012] and polygon detection [Zeng and Ma, 2010].

By making use of a kernel, it takes subregions from the image, checking then the orientation

gradient, namely the slope. After adding it into a bin, it goes further to another region repeating the process. It will stop when the whole image has been sampled. Then it goes into the bin in order to determine the edges [Zhu et al., 2006]. In order to capture the shape of the structures within the region, the algorithm divides the image into small pixels cells and blocks of cells.

Even though HOG does not provide great robustness regarding neither the motion nor the lighting conditions, it has a relatively good performance for various classes of objects. Moreover, by using HOG in accordance to our constraints enunciated above, real time detection can be performed.

2.2.2 Haar-like features

This type of features are made of a class of features that are determined by the difference of the unselected feature region and its subregion. Thanks to the straightforward calculation within an integral image, these features are considered to be very efficient.

By using the values of change in contrast between adjacent groups of pixels, it can be determined the relative light and the dark areas. A Haar-like feature is formed by two or three adjacent groups. These features can easily be scaled by changing the size of the pixel group, helping to detect objects of several dimensions. [Wilson and Fernandez, 2006] Figure 4 gives some examples of Haar features. Very important to take into consideration is the fact that this type of features can be conveniently rescaled [Lienhart et al., 2003].

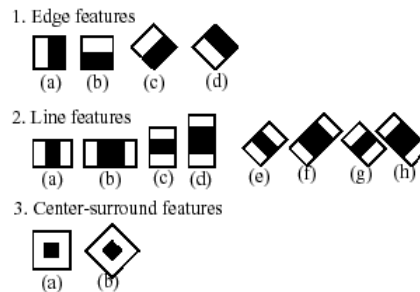


Figure 4: Examples of Haar features (adapted from [Wilson and Fernandez, 2006])

2.2.3 Discrete wavelet transformation

Within this method, a signal is decomposed into several wavelets, namely a combination of wavelets [Chaovalit et al., 2011]. There are various types of wavelets transformations, such as the Harr wavelet or the Daubechies wavelet. DWT can be broken into two types: 1-dimensional (1D) and 2-dimensional (2D). Figure 5 shows a representation of a DWT 1D which has 2 extracted components: approximation and detail [Yaji et al., 2012]. The main advantage obtained by using this method is that it captures the location in time [Mohan and Kumar, 2013].

2.2.4 Object detection by regression

Object detection by regression (ODR) detects the position of an object within an image. The use of this algorithm is common within the RoboCup environment [Visser, 2016]. It works by creating

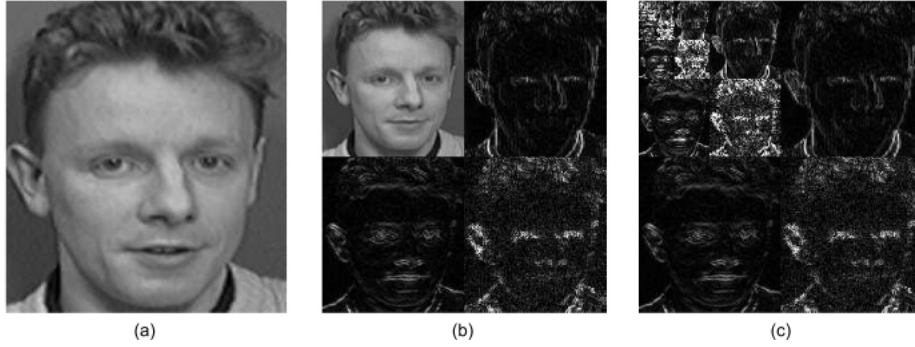


Figure 5: Example of DWT (from [Nicholl et al., 2010])

a statistical model of the relation between an image and the position of a given object in that image [Brandão et al., 2012]. This statistical model permits image sampling, all of this being done either offline or online. Its output is represented by the position of the object and a weight of confidence that the object exists (Figure 6). It is known from [Brandão et al., 2012] that ODR performs very well within a precomputed environment and detection of objects.

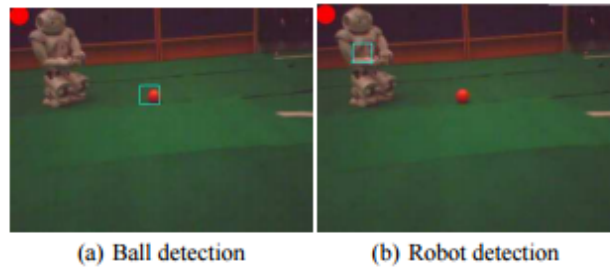


Figure 6: Examples of ball and robot detection using ODR - square indicates the location and detection of a) the ball and b) the robot (from [Brandão et al., 2012])

2.2.5 Speeded-up robust feature

Speeded-up robust feature detector (SURF) is a scale and rotation invariant interest point detector and descriptor [Bay et al., 2006]. This method can be used not only to locate and recognize objects, but also to track them [Shuo et al., 2012]. The original image is converted to a set of coordinates with a technique called the multi-resolution pyramid technique. A new image results with the same size but with reduced decreased bandwidth. A blurring effect is then created, provided that the interest point are scale invariant.

2.2.6 Scale invariant feature transform

Scale Invariant Feature Transform (SIFT) represents a method to extract feature descriptors that are invariant to rotation, scaling, quality and lighting [Lowe, 2004]. It is represented by a chosen area within the image, which is called a keypoint [Lowe, 2004], combined with a descriptor. The

SIFT detector takes care of extracting the keypoints, while the descriptors are computed using the SIFT descriptor.

2.2.7 Principal component analysis

Principal component analysis (PCA) is a way to identify patterns in data and to express data in order to highlight similarities and differences between them. Because the patterns in high-dimensional data are hard to find and as we cannot view the data, PCA is a powerful tool for analyzing them. Another advantage of the PCA method is that once found the patterns, we can reduce the number of dimensions without losing much information. A great advantage obtained by using this method is that it does not make use of large computations.

2.3 Feature extraction comparison

We have identified various extraction techniques in the previous sections that we summed up in Table 1. Based on their strengths and weaknesses, the following table shows them taking into consideration some of the constraints presented above. Because datasets are known to be large regarding the number of the measured variables within each one of them, computational complexity has a very big role. Moreover, we added the parameters within the table. An implementation can also be judged on how many parameters it makes use of, because generally, with more parameters involved comes a higher computational cost. Note that the detection process (1) and the time for each algorithm (2) have not been included in this table because they differ considering the implementation.

Feature	Robust to:			6. Offline learning	Computational complexity	Parameters
	3. Motion	4. Camera quality	5. Lighting conditions			
HOG	X [Churchill and Fedor,2014]	X [Churchill and Fedor,2014]	X [Churchill and Fedor,2014]	✓ [Kaaniche and Brémont,2009]	$O(4n^2)$	8 [Kim and Cho,2014]
HAAR	✓ [Lienhart and Maydt,2002]	X [Lienhart and Maydt,2002]	✓ [Gong et al.,2009]	✓ [Gong et al.,2009]	$O((14/3)n^2)$ [Porwik and Lisowska,2004]	2 [Chun-Lin,2010]
DWT	X [Ahmad et al.,2010]	✓ [Ahmad et al.,2010]	✓ [Ahmad et al.,2010]	✓ [Ahmad et al.,2010]	$2(2^n - 1)$ [Shukla and Tiwari,2013]	2 [Shukla and Tiwari,2013]
ODR	✓ [Brandão et al.,2012]	✓ [Brandão et al.,2012]	✓ [Brandão et al.,2012]	✓ [Brandão et al.,2012]	$O(n)$ [Brandao et al.,2010]	4 [Brandão et al.,2012]
SURF	✓ [Bay et al.,2008]	✓ [Bay et al.,2006]	✓ [Bay et al.,2006]	✓ [Sergieh et al.,2012]	$O(n^2)$ [Oyallon and Rabin,2015]	4 [Pedersen,2011]
SIFT	✓ [Lowe,2004]	✓ [Lowe,2004]	✓ [Lowe,2004]	✓ [Lowe,2005]	$> O(n)$ [Vinukonda,2011]	9 [Vinukonda,2011]
PCA	✓ [De la Torre and Black,2001]	✓ [De la Torre and Black,2001]	X [Ramamoorthi,2002]	✓ [Boutsidis et al.,2015]	$O(p^3 + p^2n)$ [Aspremont et al.,2008]	$2p + p^2/2$ [Natrella,2010]

Table 1: Feature extraction comparison - check mark (it meets the constraint) and cross (it does not satisfy the constraint)

By using the HOG method, better contour of the object can be acquired. On the other hand, if Haar-like features are used, then the regions with a bigger difference in shading will be described better. It can easily be observed that the HOG extractor is not performing as the Haar-like features, mainly because HOG detects the object’s shape rather than static objects. Since the robots are moving within the SPL game, it would be extremely hard to get a static shape. DWT is a suitable chosen method for feature extraction because of its low computational complexity. ODR is a robust and computationally efficient algorithm. According to [Panchal et al., 2013], SIFT is said

to be more robust than SURF but in the same time, it is slower.

To conclude this section, we will take into consideration the qualities and disadvantages of the studied feature extraction algorithms in order to combine them with the appropriate classification method.

2.4 Bag of Words model

The "Bag of Words" (BoW) model is an algorithm used for the first time within the classification of text documents area [Ko, 2012]. Within this model, a set of representative words, referred to as "vocabulary", is selected and then a text histogram is created for each text document. For a simpler understanding, one can associate the vocabulary as being a book, and a word with a vector. These histograms are then categorized using classification algorithms. Starting from this basic algorithm, the BoW model has been applied to various areas of computer vision: image classification [Csurka et al., 2004], as well as for the classification and the recognition of various actions [Wang et al., 2011].

The main idea of the BoW model is that keypoints are considered to be similar to words in text documents. The descriptor vector will contain a histogram of occurrence of "words" in an image, after which these histograms will be classified by classifiers. The new algorithm is called "Bag of Visual Words" (BoVW) [Csurka et al., 2004]. At the same time, the BoVW algorithm is inspired by the human form recognition system. A person can recognize certain objects even if they view only certain parts of the object.

The BoVW algorithm involves four steps: extracting visual words from an expanded set of images, creating vocabulary of visual words, calculating word histograms, and training a classifier. The schematic diagram of a BoVW system is shown in Figure 7. Initially, keywords are extracted from an image using algorithms such as SIFT or SURF. These key points are added to a keyword vector. Then, the number of keywords will be reduced by using K-means, a clustering algorithm. Each resulting centroid will be considered a word from a vocabulary of visual words. The number of vocabulary words varies, being dependent on the application, from a few thousand of words [Lazebnik et al., 2006] to hundreds of thousands of words [Zhao et al., 2006].

Next, the visual word histograms must be generated. For the calculation of the descriptors, the following steps have to be followed:

- For each image in the dataset, the keywords will be extracted and the minimum distance between them and the words in the dictionary will be calculated;
- Each keyword will be assigned to a cluster of the dictionary based on a criterion of maximum similarity; the measure of similarity is calculated with the Euclidean distance;
- A histogram for the occurrence of words in the dictionary will be created.

The main advantages of Bag of Words are the invariability of scaling, rotation and translation (no matter the spatial arrangement of visual words in an image). Moreover, it shows good performance even if partial object occlusions occur and is intuitive (due to the analogy with the classification of text documents and similarity with the human mode of object recognition).

However, one issue to take into consideration is the size of the vocabulary that must be studied: if the vocabulary size is too small, then the resulted visual words are not representative of all patches. If its size is too large, then we are dealing with overfitting.

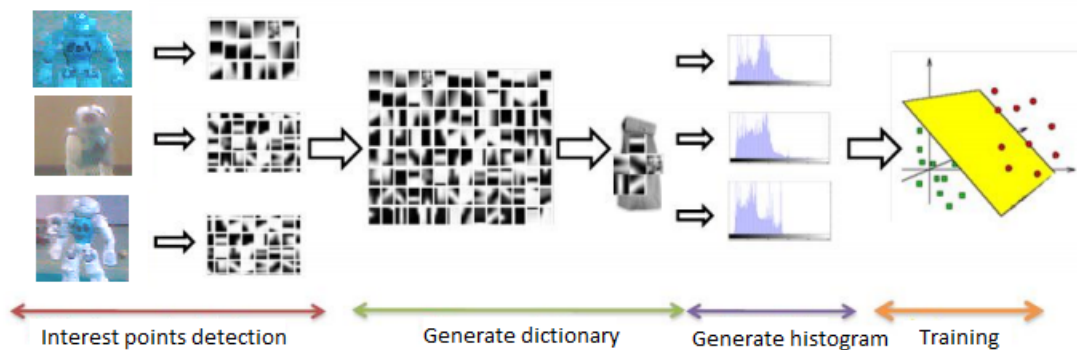


Figure 7: The training process of BoW algorithm

2.5 Classification and localization methods

Having in mind the learning diagram presented above (Figure 3), we will now take the classification methods into discussion. The classification methods take the extracted features as inputs. Within the machine learning area, the goal of an algorithm is the learning of a set of rules that have the ability to describe the set of inputs and outputs. As humans, we know those rules, but we are not able to describe them mathematically with enough concreteness. We will use the detection in order to perform the localization.

2.5.1 Support vector machines

Support vector machines (SVM) are statistics models that separate two data sets [Burges, 1998]. The classes are divided through an optimal separating hyperplane (OSH). The limits of these classes of data and the OSH are called support vectors.

Intuitively, for a set of points divided into two classes, the SVM method finds on one side the hyperplane that separates the highest possible fraction of points that belong to the same class, and on the other side it maximizes the distance between the classes and hyperplane. For a binary classification problem (see Figure 8), the aim is to separate the two classes using a function that is obtained from the available examples.

By using SVM, the user can avoid over-fitting due to its regularization parameter. The real help the SVM comes in is when there is randomly distributed data, taking into account a good training.

2.5.2 k-Nearest Neighbour

The k-nearest neighbour (kNN) algorithm [Murty and Devi, 2011] is also a classification algorithm and involves the finding of not just one, but a number of neighbors k . All the inputs are classified, meaning that belong to a class.

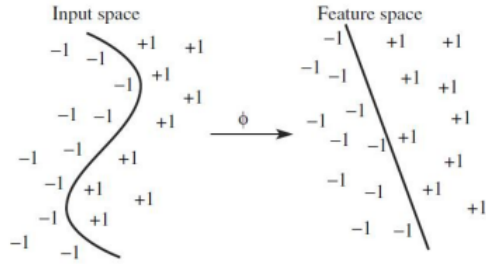


Figure 8: Example of SVM separation (from [Ivanciuc, 2007])

As an example, for the face recognition area, all images belonging to the same person forms a class. Then, the person looked for, will belong to the majority class among the k neighbors. For example, if you take the 7 nearest neighbors and 5 of them belong to a class, and the remaining 2 to another class, it can be deduced that the person sought will belong to the first class.

The value of k can be determined from experiments, being chosen the value that gives the least number of errors regarding the classification. For large data sets, to reduce classification errors, a higher value of k can be chosen. k NN is known for its good performance on basic recognition problems, such as face recognition using the ORL data set¹. Though, k NN uses its data for classification only, rather than learning from the training part.

2.5.3 AdaBoost

The method Adaptive boosting (AdaBoost) [Guo and Zhang, 2001] is based on the idea of creating a predictor with high degree of accuracy by combining multiple "weak" classification functions. AdaBoost is an adaptive algorithm that combines a sequence of classifiers for which the weights are updated dynamically depending on the errors that occur prior to learning. AdaBoost is a classifier with a high margin of error. Within this method, for training the new iteration it uses the results from the previous ones, in order to improve the performance.

This classifier is known for being simple to implement, improving the accuracy of the classification. It performs the selection of features that result in a simple classifier. A major difference between SVM and AdaBoost is that the latter selects only those features that are known to elevate the predictive power of the model. Through this, it reduces the dimensionality and improves the execution time as the irrelevant features do not need to be computed.

2.5.4 Simultaneous localization and mapping

Self-localization and localization of other robots and objects is an important aspect, especially regarding the RoboCup SPL contest. It is very crucial for the robot to know where he is placed within the field so as to finish doing the next decision making tasks such as passing the ball or plan its future path. The first fact that we have to take into consideration is that the robot does not observe its surroundings in a completely way at a given time, this not being enough to determine its precise position. Moreover, the landmarks within the field can be often vague.

¹<http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>

Algorithm	Handling of missing values	Computational scalability (large n)	Ability to extract linear combinations of features	Predictive power
SVM	X [Hastie et al., 2003]	X [Hastie et al., 2003]	✓[Hastie et al., 2003]	✓[Hastie et al., 2003]
k-Nearest Neighbours	X [Hastie et al., 2003]	X [Hastie et al., 2003]	fair [Hastie et al., 2003]	✓[Hastie et al., 2003]
AdaBoost	✓[Hastie et al., 2003]	✓[Cooper and Reyzin,]	✓[Hastie et al., 2003]	✓[Hastie et al., 2003]
SLAM	✓[Hastie et al., 2003]	✓[Hastie et al., 2003]	✓[Hastie et al., 2003]	✓[Hastie et al., 2003]
Particle filter	✓[Hastie et al., 2003]	✓[Hastie et al., 2003]	✓[Hastie et al., 2003]	✓[Hastie et al., 2003]

Table 2: Classification methods comparison (adapted from [Hastie et al., 2003]) - fair (provides medium fulfillment), check mark (it meets the constraint) and cross (it does not satisfy the constraint)

The SLAM method discusses the possibility for a robot to be placed in an unknown environment [Durrant-Whyte and Bailey, 2006], having to move around to collect data of landmarks within the environment. This should be done by using its sensors and by recording its position with respect to that created map. This method is known for working well with a small number of features and various landmarks. Though, it performs slow within high dimensional maps [Aulinas et al., 2008].

We will use this just for localization. Thus, we could also take into consideration the method described in [Coaguila et al., 2016] which uses HOG with structural SVM. Afterwards, by using the method proposed in [Kazemi and Sullivan, 2014] we can localize the landmarks in real time.

2.5.5 Particle filter

Particle filter represents a popular choice regarding the localization within the area of robotics [Thrun, 2002]. This method implies that the probability of the robot to be somewhere taking into consideration the observations, i.e. what it is detected within its environment. The purpose of using this method within our context is to track the actual position of the robot, knowing that its precise location is not known by the algorithm.

2.6 Classification and localization methods comparison

For each particular method there are situations for which it is particularly well suited, and others where it performs badly compared to the best that can be done with that data. We have attempted to characterize appropriate situations in our discussions of each of the respective methods. However, it is seldom known in advance which procedure will perform best or even well for any given problem. We took into consideration methods both for detection and localization. So, as characteristics that describe and classify the alg we choose as it follows. To start with, the handling of missing data was taken into consideration due to the fact there are a high number of missing values within the observations. One is seldom able to find a complete observation. Because the variables within the dataset are usually measured on various scales, thus different ones, it is important for the algorithm to be scalable from the computational point of view.

The ability to extract linear combinations of features is important because the entire model can be completely represented by a simple two-dimensional graphic (binary tree) that is easily visualized. Usually only a small fraction of the large number of predictor variables that have been included in the analysis are actually relevant to prediction. Also, unlike many applications such as pattern recognition, there is seldom reliable domain knowledge to help create especially relevant features and/or filter out the irrelevant ones, the inclusion of which dramatically degrades the performance of many methods.

3 Contribution

Object detection based on visual content uses detection techniques of the visual features, a way of expressing the images through visual descriptors and metrics of similarity based on descriptors that allow the identification of the closest images in terms of certain visual aspects.

Having in mind the objective of the research project, there will be two steps to acquire: first, the detection of the object followed by its tracking. For these to be acquired, a method based on extracting the SIFT features combined within a "Bag-of-Words" model is used. For the classification, SVM method will be implemented.

3.1 Proposition

The proposed solution to this problem includes the use of the SIFT feature extraction technique within the Bag-of-Words model and SVM classification method.

Within the field, one of the best identified solutions for identifying visual concepts in images are the SIFT descriptors - Invariant Feature Transform Scale [Lowe, 2004], along with SVM-based classification.

The SIFT descriptor is used with very good results in this area for extracting features that are, to a certain extent, invariant to changes such as light intensity, image noise, rotation, scaling, etc. According to [Lowe, 2004], the process of detecting such features implies:

- Scale-space extreme detection: the detection of interest points represented by extreme points of DoG (difference-of-Gaussian) at different scales;
- Localization of the key points: minimum/ maximum local points for DoG; comparison of each pixel in the converted DoG image with the 8 neighbors of a certain scale plus the neighbors from the other scales; After the extreme points are calculated, the low contrast points and less outlined edges will be eliminated. The remaining points represent the points of interest of the image. These are invariant to scaling the image or adding different forms of noise. A keyword descriptor is a 128-dimensional vector (one byte for each feature);
- Assignment of the orientation for each key point, which is based on the calculation of a gradient orientation histogram in the vicinity of the key points;
- Creation of descriptors for the key points; based on the orientation of the key points, the descriptor is obtained as a set of orientation histograms on 4x4 pixel neighborhoods.

After generating the descriptor, a classification algorithm is used. The BoW (the concise representation of the image), serves as an input to a classification algorithm. In this proposal it is made use of the SVM classification technique (Figure 9).



Figure 9: The classification process of BoW algorithm

The SVM classification can be done for two classes, the case of a single class binary classifier (OneClass SVM), being used in anomaly detection, for several multi-class classes, where an instance belongs to a single class and for multiple overlay classes, called multi-label, where an instance can have multiple labels. Two approaches are briefly stated that belong to the multi-class classification: one-versus-one and one-versus-all. The technique used here will be one-versus-all.

The one-versus-one approach involves the reducing of the multi-class classification problem to a set of binary classification problems in which $N * (N-1) / 2$ binary classifiers (where N represents the total number of classes) are constructed. A classifier allows the training of each pair of class - "one against one". In order to determine the class for a given instance, a voting method is used, where every classifier gives a vote to the class it predicts for that particular instance. The class with the most votes wins and becomes the final predicted class for the instance.

The one-versus-all approach involves the training of N classifiers which groups a class against all other classes - "one against the rest". The classification of an instance will be based on predicting that class that maximizes the edge between the instance and the trained separation plan of the classifier.

3.2 System description

This subsection gives an overview on the development part for image and video capture for the testing part of the solution, for the detection of the BoW descriptors and for the SVM classification.

The development of this project took part in C++ by making use of OpenCV 3.2.0 (Open Source Computer Vision Library) ², an open source computer vision and machine learning software libraries. The code of the implementation resides at the Github link ³.

1. First, within the ImageReader class, the positive and negative images from specific folders corresponding to the training part are being read, and the video that corresponds to the

²<http://opencv.org/>

³<https://github.com/AndreeaOana/NaoDetect>

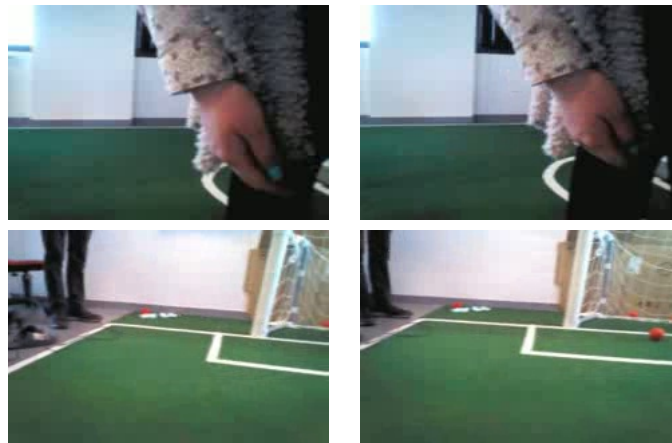
testing side. Then, a grayscale `cv::Mat` vector is created and their labels are stored in the `TrainLabels`, respectively `TestLabels` vectors.

Also, within this class is implemented the function `rotate90n` that rotates the video that was considered as an input. We needed this function in order to fully test our solution. Note that the training is performed with cropped positive and negative images. Below there are some examples of the used training images:

Example of positive training images:



Example of negative training images:



2. After all the inputs have been read and assigned labels, the `BagOfSIFT` class extracts the features that correspond to every whole image (or frame). `cv::Mat` arrays are used to store all the descriptors extracted from the images and frames. After the SIFT features are detected, the descriptors are computed. Another array will be used to store the keypoints that will be extracted by SIFT. SIFT descriptors are sampled from each training image by scanning the image with a design parameter `StepSize` (which signifies the sampling rate). The value of the `StepSize` in our case is set to 200. The higher the value of the `StepSize`, the fewer SIFT descriptors from each image.

By using 2D Features Framework from the OpenCV library, the SIFT feature extractor and SIFT descriptor extractor are created. Note that by doing feature extraction we work on a representation of the characteristic features that were extracted from an image or frame. The descriptor extractor is used to compute the descriptors for an input image and its keypoints. Next, the extracted features are matched by using use of the FlannBasedMatcher interface in order to perform a quick and efficient matching by using the FLANN (Fast Approximate Nearest Neighbor Search Library) of OpenCV. This step is required in order to search for the nearest word of the trained vocabulary for each descriptor within the image. Figure 10 shows an example:



Figure 10: Descriptors matching

We can now create the Bag-of-Words descriptor extractor, by making use of the SIFT feature extractor and descriptor matcher. For this step, the constructor of the OpenCV class that implements the extractor is: `BOWImgDescriptorExtractor::BOWImgDescriptorExtractor(const Ptr<DescriptorExtractor> & dextractor, const Ptr<DescriptorMatcher> & dmatcher)`, where `dextractor` is the descriptor extractor and `dmatcher` is the descriptor matcher

3. K-means algorithm is used to cluster and compute vocabulary. It located centers of clusters and groups the inputs around the clusters. With the help of the vocabulary, compute histograms for training.

In this part we use a `DictionarySize` variable, that gives the `K` value to be used in K-means algorithm. This determines the influence of the misclassification on the objective function. We set the dictionary with the created vocabulary. `BOWKMeansTrainer` is constructed, having the following constructor: `BOWKMeansTrainer::BOWKMeansTrainer(int clusterCount, const TermCriteria& termcrit=TermCriteria(), int attempts=3, int flags=KMEANS_PP_CENTERS)`, where we use the following values:

- `clusterCount` is represented by the `dictionarysize`,
 - the `termcrit` which is the maximum number of iterations is 100 and the change in parameters at which the iterative algorithm stops is 0.001
 - the number of attempts is 1
 - the necessary flags are calculated by `cv::KMEANS_PP_CENTERS`
4. Classification is done with the SVM technique. The training is based on a set of trained cropped images represented by a set of visual descriptors on the one hand and the class information associated with each image.

The class weights control the trade-off between achieving a low error on the training data and minimizing the norm of the weights, being used in order to reduce the bias. Note that for very tiny values of this parameter, misclassified examples will appear. The value of the C_value parameter is 120, chosen empirically.

For every frame loaded, a sliding-window approach is used: having two loops that correspond to the rows and the columns of the image, a rectangle with the dimensions of 80x80, slides through the picture with a StepSlide equal to 50. This result in a region of interest (ROI). For every ROI, the BoW descriptors are extracted, SVM is performed, and then the predict function determines the class label. For every frame it is decided where there is a robot within it, or not.

For the real-time system version, the same steps are being followed, with the differences that that training data set is read, having the BoW descriptors extracted and them SVM being trained on them. Then, for every frame read the steps describes above are being followed.

4 Evaluation

Several experiments have been done to evaluate the proposed object detection and tracking algorithms in the context of various scenarios.

Hardware resources For this thesis, two pieces of hardware equipment were used in order to validate and test the proposed solution: the NAO robot's cameras and a Hercules Dualpix HD Webcam. As described in the introduction of this thesis, the robot's cameras are not of a very good quality. Moreover, from their physical location, we deduce that the two cameras do not overlap sufficiently to allow stereo vision. As an example of what the robot is able to capture (Figure 11), in its default head position, the bottom camera only gives images form the floor and the bottom parts of the robot, including its hands, while its top camera will capture what lies in the forward areas of the robot.

However, due to technical constraints, we used regular webcam for our tests instead of the tests. The camera was chosen so as to be of a lower quality and programmatically downgraded the image quality to simulate the robot's conditions.

First, the proposed solution was tested with the help of the camera. Measurements were carried out for minimum and maximum detection corresponding to different poses of the robot.

For the side pose of the robot, the minimum distance that the robot can be detected at is 44 cm, whereas the maximum distance is 280 cm.

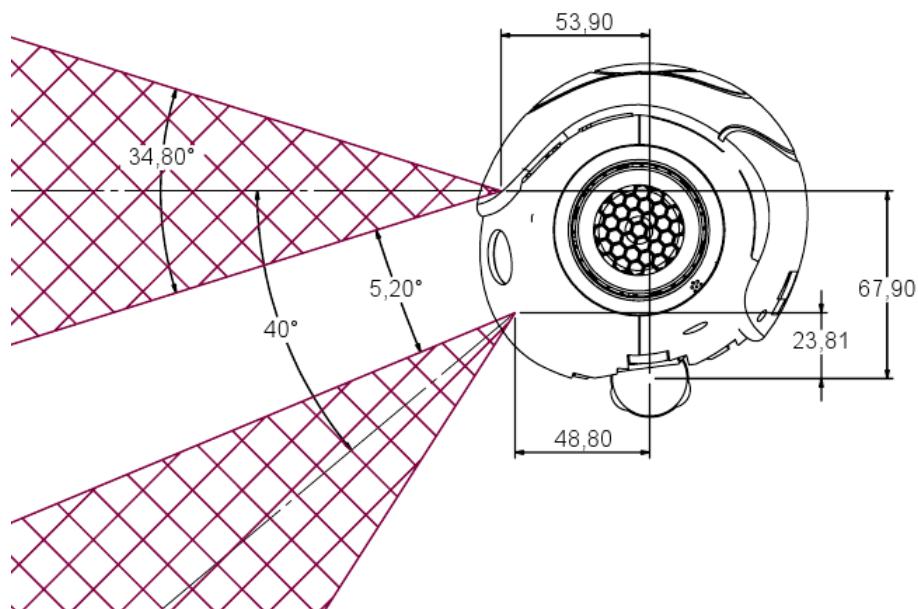
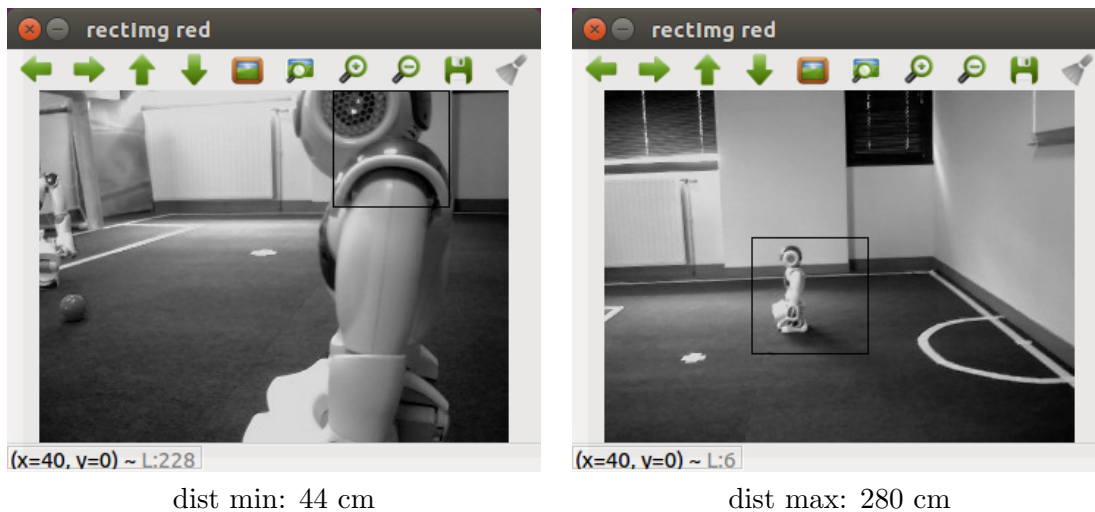
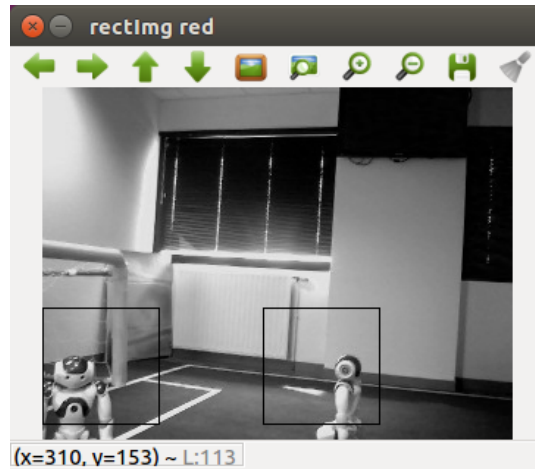
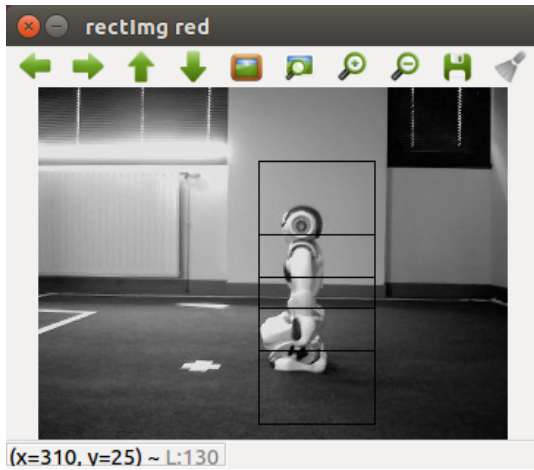


Figure 11: Position and orientation of cameras on Nao's head with respect to head coordinate frame.

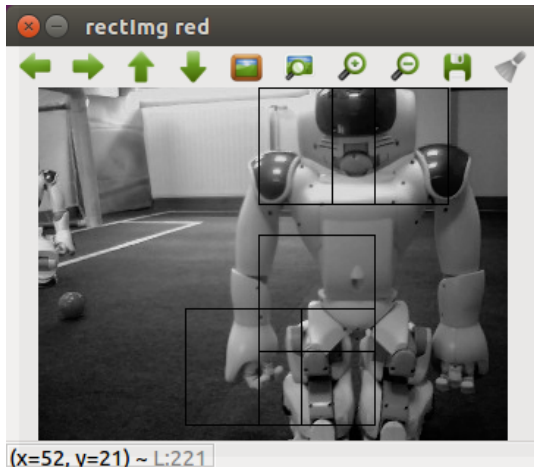


Below there are some examples of several detections of the robot's side in between the minimum and maximum distance. As observed, the algorithm is able to detect not only the whole robot, but also the parts of the robot. By using the sliding window approach, we will inevitably have to deal with overlapping. For the moment, they are not grouped together in the program, though not being considered as 1 robot. In the second figure, two robots are detected.

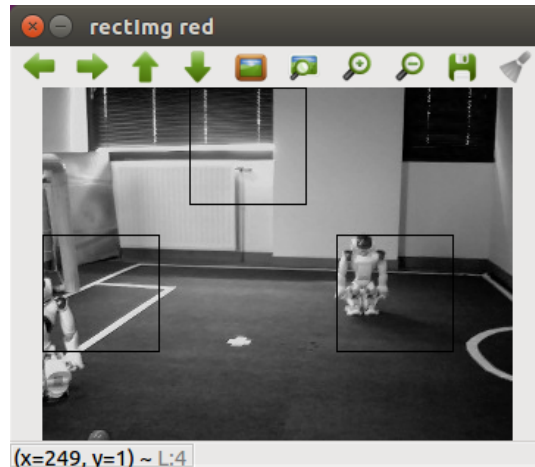


Regarding the situations where the back of the robot must be detected, the minimum distance is 60 cm, and the maximum distance where the detection happens is 300 cm.

In the second figure, an example of false positive detection can also be seen.

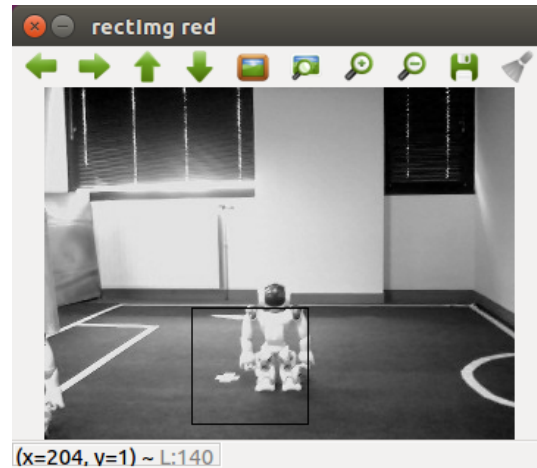
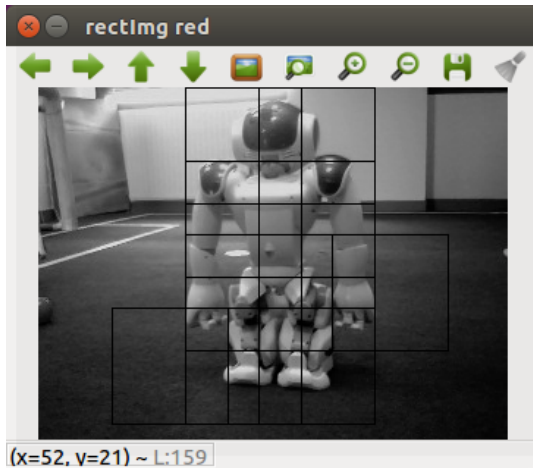


dist min: 60 cm

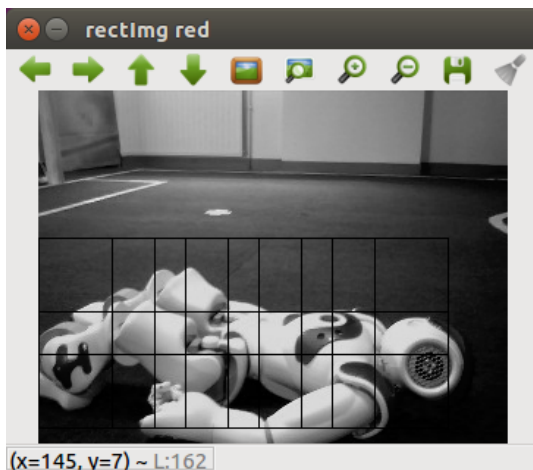


dist max: 300 cm

Next, some examples of several detections of the robot's back are shown. In the first picture, the robot is standing at a distance of 87 cm away from the camera, whereas in the second one the robot with the back stands at 240 cm.



Because during the football match the robots will fall various times, it is therefore necessary to test our solution for when the robot is lying on the field. The minimum detection distance is 44 cm, whereas the maximum distance is 170 cm.



dist min: 44 cm

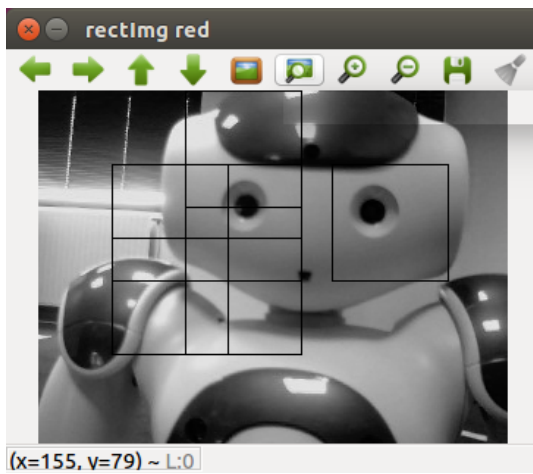


dist max: 170 cm

In what it follows, two more examples of this pose detection are shown. In the first figure it can be observed that it is also detected very well the second robot in the standing position. However, the second figure has some false positives.



Lastly, the front standing position was also analyzed. Its minimum detection distance is 26 cm and its maximum detection is at 440 cm.

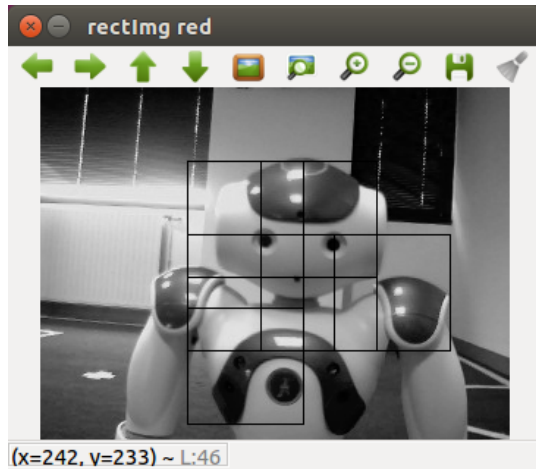


dist min: 26 cm

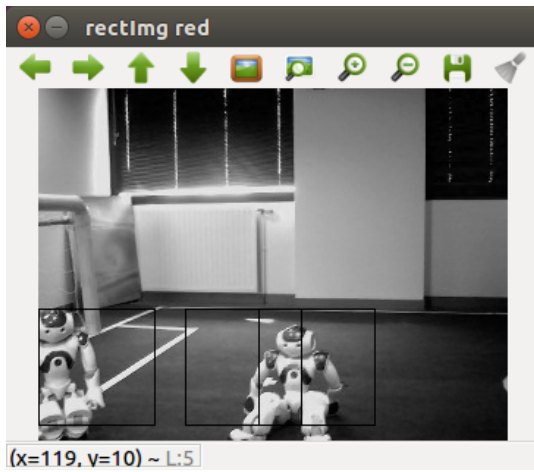
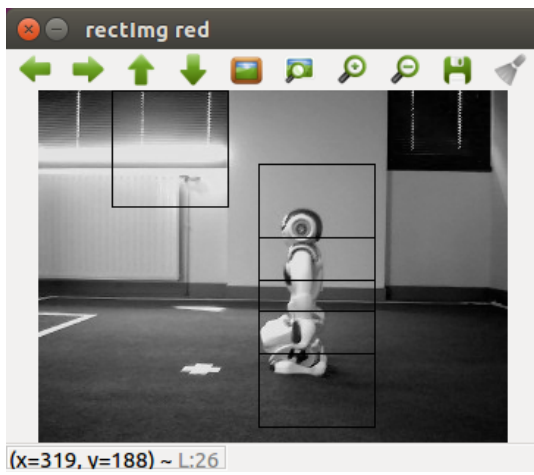


dist max: 440 cm

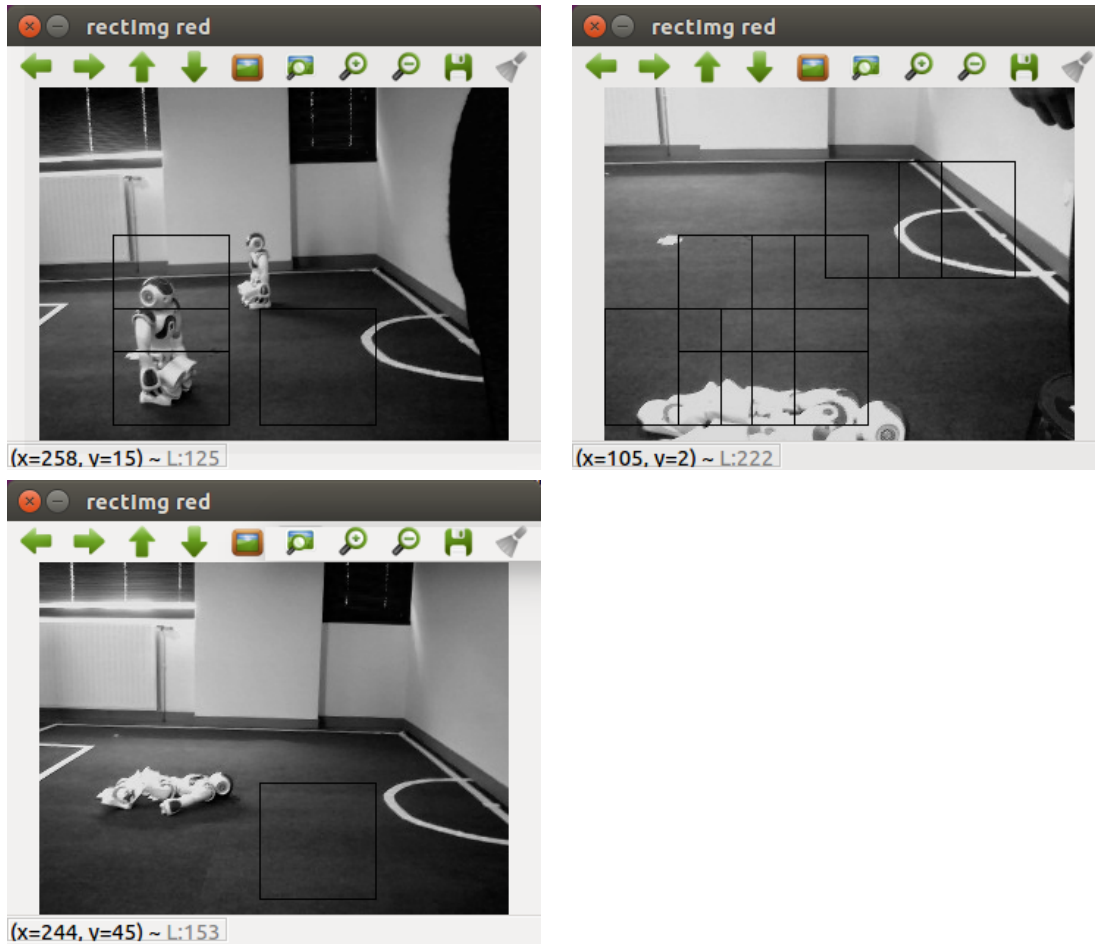
Another detection of the robot's front at 45 cm is shown below:



More detections are presented below.



Examples of false detection:



Secondly, multiple tests had been done to evaluate the detection algorithm. During the experiments, there were taken into consideration three situations so that the proposed solution can be fully evaluated. Three videos were taken with the NAO robot's cameras according to three different scenarios:

- moving robot and moving target (Table 3)
- standing robot and moving target (Table 4)
- standing robot and standing target (Table 5)

In order for the efficiency of the classifier to be measured, criteria has to be used to perform an evaluation. The performance measures, applied to our case, can be divided into four parts:

- True positive (TP): Robot detection that is classified as the robot.
- False positive (FP): Robot detection that is classified as non-robot.

- False negative (FN): Non-robot detection that is classified as the robot.
- True negative (TN): Non-robot detection that is classified as non-robot.

To this are added the precision, which measures the proportion of correctly classified data, and the recall, which calculates the proportion between the true positives and true positives plus false negatives.

$$Precision = \frac{TP}{TP+FP}$$

$$Recall = \frac{TP}{TP+FN}$$

Both the precision and the recall have some disadvantages, such as the precision not talking about the data belonging to a given label and the recall not taking into account the data that does not belong to the given category. Therefore, we also take into consideration the F-measure.

$$F - measure = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

Speed The speed is also a very important factor in evaluating the solution. The average detection time is **134,75 ms**, whereas the average time for feature extraction is **138 ms per sliding window**.

The same training dataset was used for this experiment as presented above. In order to perform the testing of the proposed solution, it was necessary to have an annotated dataset. For the annotations of the frames, imglab tool (Figure 12) from the Dlib toolkit was used⁴. This tool created a simple XML file that contains the list of relative paths for all the images. Next, we used it to specify where the objects are in the images.

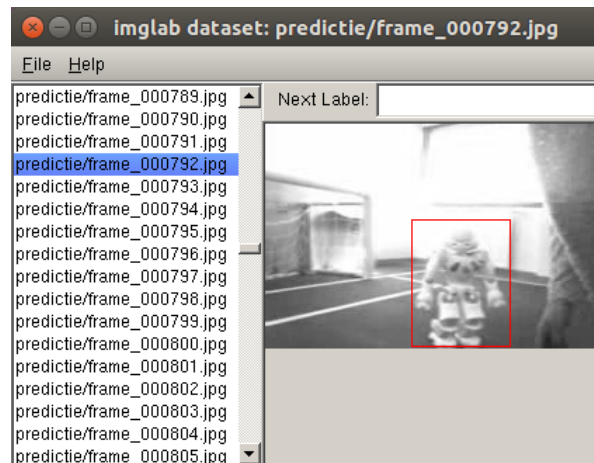


Figure 12: DLIB imglab

⁴<http://dlib.net/>

Performance measure	Values
True positive (TP)	1073
False positive (FP)	983
False negative (FN)	2002
True negative (TN)	11182
Precision	52,18%
Recall	34,89%
F-measure	41,81%

Table 3: moving robot and moving target

After marking with the bounding boxes the robots within the frames, the XML file has the following structure, containing the bounding boxes for the objects of interest (Figure 13).

```

1 <?xml version='1.0' encoding='ISO-8859-1'?>
2 <?xml-stylesheet type='text/xsl' href='image_metadata_stylesheet.xsl'?>
3 <dataset>
4 <name>inglab dataset</name>
5 <comment>Created by inglab tool.</comment>
6 <images>
7 <image file='predictie/frame_000001.jpg'>
8 <box top='76' left='127' width='39' height='60' />
9 </image>
10 <image file='predictie/frame_000002.jpg'>
11 <box top='79' left='129' width='39' height='55' />
12 </image>
13 <image file='predictie/frame_000003.jpg'>
14 <box top='71' left='125' width='48' height='65' />
15 </image>
16 <image file='predictie/frame_000004.jpg'>
17 <box top='72' left='124' width='45' height='62' />
18 </image>
19 <image file='predictie/frame_000005.jpg'>
20 <box top='76' left='121' width='53' height='59' />
21 </image>
22 <image file='predictie/frame_000006.jpg'>
23 <box top='75' left='120' width='50' height='58' />
24 </image>
25 <image file='predictie/frame_000007.jpg'>
26 <box top='78' left='126' width='42' height='57' />
27 </image>
28 <image file='predictie/frame_000008.jpg'>
29 <box top='78' left='121' width='50' height='56' />
30 </image>

```

Figure 13: XML file

During the experiment, tables for the three situations have been established to verify the reliability of results in object detection and object tracking (Tables 3, 4, 5).

5 Conclusion

Summarizing, the paper's focus is on the detection and tracking of robots within the field. Taking into consideration all the literature that had been studied and put into discussion, the chosen method for the implementation were described and evaluated.

5.1 Synthesis

The work presented focuses on the problem of detection and localization of the humanoid robot NAO within the RoboCup SPL match. The first chapter introduces the context by presenting the problem and the challenges faced. It also exposes the software and hardware constraints. Chapter

Performance measure	Values
True positive (TP)	279
False positive (FP)	233
False negative (FN)	195
True negative (TN)	933
Precision	54.49%
Recall	58.86%
F-measure	56.58%

Table 4: standing robot and moving target

Performance measure	Values
True positive (TP)	123
False positive (FP)	0
False negative (FN)	81
True negative (TN)	796
Precision	100%
Recall	60.29%
F-measure	75.22%

Table 5: standing robot and standing target

two reports the state of the art of the domain. It starts with a description of the learning process. It also carries out a brief description of each method we used both for feature extraction but also for classification. Furthermore, these methods are discussed with the help of the two tables within which they are compared based on existing literature review.

In the third chapter the proposed solution is described in more detail, followed by the introduction of the resources that were used during the work on this internship. The aspects of testing with videos taken with the robot and with the webcam are discussed and lastly some examples are presented.

Chapter four describes the experiments that were conducted with the purpose of evaluating the proposed solution with the help of various scenarios.

Lastly, chapter five concludes the paper, presenting the limitations of the proposed method, as well as some planned future work.

5.2 Limitations

The research question is placed within the context of whether the objects and robots within the field can be detected and localized, taking into consideration the constraints. Therefore, the purpose is to find the best possible solution with the help of machine learning and computer vision algorithms. In this state-of-the-art, different methods from the machine learning and computer vision areas are presented and compared.

The analyzed methods are made up of steps in order to perform the learning process. Firstly, the inputs are obtained and the detection of objects is done. Afterwards, the object that was detected is characterized and classified.

We proposed the implementation of Bag-of-Words model with SIFT descriptors classified with SVM. The solution has been extensively tested for view point invariance and scale invariance. Although it has better results than other methods regarding the changes in the illumination condition, this method does not have a high accuracy for the case when the both the robots are moving.

It is a possibility for the result to be better, if the training dataset is improved. Images from different angles should be added, and with more distance variation.

Another idea would be to perform some parameter search in order to get the best performing ones.

The main disadvantages of the Bag of Words algorithms are:

- There is no rigorous method of representing the component objects, of the spatial distribution of certain pairs of words;
- Segmentation and localization of components are unclear;
- There are many words that are not relevant;
- The computational cost increases with the size of the vocabulary of words. The challenges that may arise with this is that if it is desired a real time image processing on the robot, it will be highly resource consuming.

In order to solve some of the shortcomings, several changes have been proposed to the classic Bag-of-Words model. In [Uijlings et al., 2009], different ways have been proposed to increase the calculation speed.

5.3 Future work

One thing to consider is to actually put the code on board of the robot, not only testing with videos taken with the robot's cameras. This rises some problems, considering that the OpenCV's version that is on the robot is 2.3.1 and it lacks various functions from the machine learning library.

Although this aims to mark a period of research, the work described in this thesis is far from complete. A first application that can be improved is the detection and tracking of the robot. In this case, I intend to improve the performance of the algorithm and to adapt to other types of objects, such as a soccer ball.

Moving on to the localization, we consider it as future work. We have a classifier that takes as input an image and outputs a probability for a given class. Instead of simply feeding the classifier with the whole image, we loop over a lot of windows in the image and feed each sub-image to the classifier. Now, we have a probability for every sub-image in the original image. In the future, we could implement non-maximal suppression in order to get the locations of the desired object.

References

- [Ahmad et al., 2010] Ahmad, W., Hacihabiboglu, H., and Kondoz, A. M. (2010). Discrete wavelet transform based shift-invariant analysis scheme for transient sound signals. In *13th Int. Conference on Digital Audio Effects*.
- [Aulinas et al., 2008] Aulinas, J., Petillot, Y. R., Salvi, J., and Lladó, X. (2008). The slam problem: a survey. In *CCIA*, pages 363–371. Citeseer.
- [Bay et al., 2006] Bay, H., Tuytelaars, T., and Van Gool, L. (2006). Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer.
- [Bird et al., 2009] Bird, S., Klein, E., and Loper, E. (2009). *Natural language processing with Python*. ” O’Reilly Media, Inc.”.
- [Boutsidis et al., 2015] Boutsidis, C., Garber, D., Karnin, Z., and Liberty, E. (2015). Online principal components analysis. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 887–901. Society for Industrial and Applied Mathematics.
- [Brandão et al., 2012] Brandão, S., Veloso, M., and Costeira, J. P. (2012). *Fast Object Detection by Regression in Robot Soccer*. Springer Berlin Heidelberg, Berlin, Heidelberg.
- [Burges, 1998] Burges, C. J. (1998). A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167.
- [Chaovalit et al., 2011] Chaovalit, P., Gangopadhyay, A., Karabatis, G., and Chen, Z. (2011). Discrete wavelet transform-based time series analysis and mining. *ACM Computing Surveys (CSUR)*, 43(2):6.
- [Chun-Lin, 2010] Chun-Lin, L. (2010). A tutorial of the wavelet transform. *NTUEE, Taiwan*.
- [Churchill and Fedor, 2014] Churchill, M. and Fedor, A. (2014). Histogram of oriented gradients for detection of multiple scene properties.
- [Coaguila et al., 2016] Coaguila, R., Sukthankar, G. R., and Sukthankar, R. (2016). Selecting vantage points for an autonomous quadcopter videographer. In *FLAIRS Conference*, pages 386–391.
- [Cooper and Reyzin,] Cooper, J. and Reyzin, L. Improved algorithms for distributed boosting.
- [Csurka et al., 2004] Csurka, G., Dance, C., Fan, L., Willamowski, J., and Bray, C. (2004). Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, pages 1–2. Prague.
- [Dalal and Triggs, 2005] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, volume 1, pages 886–893. IEEE.
- [dAspremont et al., 2008] dAspremont, A., Bach, F., and Ghaoui, L. E. (2008). Optimal solutions for sparse principal component analysis. *Journal of Machine Learning Research*, 9(Jul):1269–1294.
- [De la Torre and Black, 2001] De la Torre, F. and Black, M. J. (2001). Robust principal component analysis for computer vision. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 362–369. IEEE.
- [Dray and Josse, 2015] Dray, S. and Josse, J. (2015). Principal component analysis with missing values: a comparative survey of methods. *Plant Ecology*, 216(5):657–667.
- [Durrant-Whyte and Bailey, 2006] Durrant-Whyte, H. and Bailey, T. (2006). Simultaneous localisation and mapping (slam): Part i the essential algorithms.
- [Gong et al., 2009] Gong, H., Li, C., Dai, P., and Xie, Y. (2009). Object tracking based on the combination of learning and cascade particle filter. In *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*, pages 978–983. IEEE.

- [Guo and Zhang, 2001] Guo, G.-D. and Zhang, H.-J. (2001). Boosting for fast face recognition. In *Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, 2001. Proceedings. IEEE ICCV Workshop on*, pages 96–100. IEEE.
- [Hastie et al., 2003] Hastie, T., Tibshirani, R., and Friedman, J. (2003). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer; 1st ed. 2001. Corr. 3rd printing edition (July 30, 2003).
- [Ivanciuc, 2007] Ivanciuc, O. (2007). Applications of support vector machines in chemistry. *Reviews in computational chemistry*, 23:291.
- [Kaaniche and Brémond, 2009] Kaaniche, M. B. and Brémond, F. (2009). Tracking hog descriptors for gesture recognition. In *Advanced Video and Signal Based Surveillance, 2009. AVSS'09. Sixth IEEE International Conference on*, pages 140–145. IEEE.
- [Kazemi and Sullivan, 2014] Kazemi, V. and Sullivan, J. (2014). One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1867–1874.
- [Khandelwal et al., 2010] Khandelwal, P., Hausknecht, M., Lee, J., Tian, A., and Stone, P. (2010). Vision calibration and processing on a humanoid soccer robot. In *The Fifth Workshop on Humanoid Soccer Robots*.
- [Kim and Cho, 2014] Kim, S. and Cho, K. (2014). Fast calculation of histogram of oriented gradient feature by removing redundancy in overlapping block. *J. Inf. Sci. Eng.*, 30(6):1719–1731.
- [Kitano et al., 1997] Kitano, H., Asada, M., Kuniyoshi, Y., Noda, I., Osawa, E., and Matsubara, H. (1997). Robocup: A challenge problem for ai. *AI magazine*, 18(1):73.
- [Ko, 2012] Ko, Y. (2012). A study of term weighting schemes using class information for text classification. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 1029–1030. ACM.
- [Lazebnik et al., 2006] Lazebnik, S., Schmid, C., and Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, volume 2, pages 2169–2178. IEEE.
- [Lienhart et al., 2003] Lienhart, R., Kuranov, A., and Pisarevsky, V. (2003). Empirical analysis of detection cascades of boosted classifiers for rapid object detection. In *Joint Pattern Recognition Symposium*, pages 297–304. Springer.
- [Lienhart and Maydt, 2002] Lienhart, R. and Maydt, J. (2002). An extended set of haar-like features for rapid object detection. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 1, pages I–900. IEEE.
- [Lowe, 2004] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110.
- [Lowe, 2005] Lowe, D. G. (2005). Demo software: Sift keypoint detector.
- [Masterjohn et al., 2015] Masterjohn, J. G., Polceanu, M., Jarrett, J., Seekircher, A., Buche, C., and Visser, U. (2015). Regression and mental models for decision making on robotic biped goalkeepers. In *Robot Soccer World Cup*, pages 177–189. Springer.
- [Mohan and Kumar, 2013] Mohan, S. and Kumar, S. S. (2013). *Proceedings of the Fourth International Conference on Signal and Image Processing 2012 (ICSIP 2012)*, volume 1. Springer Science & Business Media.
- [Murty and Devi, 2011] Murty, M. N. and Devi, V. S. (2011). *Pattern recognition: An algorithmic approach*. Springer Science & Business Media.

- [Naoqi, 2016] Naoqi (2016). Aldebaran documentation, <http://doc.aldebaran.com/2-1/naoqi/>. [Online; accessed 11-January-2016].
- [Natrella, 2010] Natrella, M. (2010). Nist/sematech e-handbook of statistical methods.
- [Nicholl et al., 2010] Nicholl, P., Ahmad, A., and Amira, A. (2010). Optimal discrete wavelet transform (dwt) features for face recognition. In *Circuits and Systems (APCCAS), 2010 IEEE Asia Pacific Conference on*, pages 132–135. IEEE.
- [Niemüller et al., 2011] Niemüller, T., Ferrein, A., Eckel, G., Pirro, D., Podbregar, P., Kellner, T., Rath, C., and Steinbauer, G. (2011). *Providing Ground-Truth Data for the Nao Robot Platform*, pages 133–144. Springer Berlin Heidelberg, Berlin, Heidelberg.
- [Oyallon and Rabin, 2015] Oyallon, E. and Rabin, J. (2015). An analysis of the surf method. *Image Processing On Line*, 5:176–218.
- [Panchal et al., 2013] Panchal, P., Panchal, S., and Shah, S. (2013). A comparison of sift and surf. *International Journal of Innovative Research in Computer and Communication Engineering*, 1(2):323–327.
- [Pedersen, 2011] Pedersen, J. T. (2011). Surf: Feature detection & description. *Technická zpráva, AARHUS University*.
- [Porwik and Lisowska, 2004] Porwik, P. and Lisowska, A. (2004). The haar-wavelet transform in digital image processing: its status and achievements. *Machine graphics and vision*, 13(1/2):79–98.
- [Ramamoorthi, 2002] Ramamoorthi, R. (2002). Analytic pca construction for theoretical analysis of lighting variability in images of a lambertian object. *IEEE transactions on pattern analysis and machine intelligence*, 24(10):1322–1333.
- [Sergieh et al., 2012] Sergieh, H. M., Egyed-Zsigmond, E., Döllner, M., Coquil, D., Pinon, J.-M., and Kosch, H. (2012). Improving surf image matching using supervised learning. In *Signal Image Technology and Internet Based Systems (SITIS), 2012 Eighth International Conference on*, pages 230–237. IEEE.
- [Shukla and Tiwari, 2013] Shukla, K. and Tiwari, A. K. (2013). Filter banks and dwt. In *Efficient Algorithms for Discrete Wavelet Transform*, pages 21–36. Springer.
- [Shuo et al., 2012] Shuo, H., Na, W., and HuaJun, S. (2012). Object tracking method based on surf. *AASRI Procedia*, 3:351–356.
- [Skibbe and Reiser, 2012] Skibbe, H. and Reiser, M. (2012). Circular fourier-hog features for rotation invariant object detection in biomedical images. In *ISBI*, pages 450–453.
- [Thrun, 2002] Thrun, S. (2002). Particle filters in robotics. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, pages 511–518. Morgan Kaufmann Publishers Inc.
- [Uijlings et al., 2009] Uijlings, J. R., Smeulders, A. W., and Scha, R. J. (2009). Real-time bag of words, approximately. In *Proceedings of the ACM international Conference on Image and Video Retrieval*, page 6. ACM.
- [Vinukonda, 2011] Vinukonda, P. (2011). *A study of the scale-invariant feature transform on a parallel pipeline*. PhD thesis, Louisiana State University.
- [Visser, 2016] Visser, U. (2016). 20 years of robocup. *KI - Künstliche Intelligenz*, 30(3):217–220.
- [Wang et al., 2011] Wang, H., Kläser, A., Schmid, C., and Liu, C.-L. (2011). Action recognition by dense trajectories. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3169–3176. IEEE.
- [Wilson and Fernandez, 2006] Wilson, P. I. and Fernandez, J. (2006). Facial feature detection using haar classifiers. *Journal of Computing Sciences in Colleges*, 21(4):127–133.

- [Yaji et al., 2012] Yaji, G. S., Sarkar, S., Manikantan, K., and Ramachandran, S. (2012). Dwt feature extraction based face recognition using intensity mapped unsharp masking and laplacian of gaussian filtering with scalar multiplier. *Procedia Technology*, 6:475–484.
- [Zeng and Ma, 2010] Zeng, C. and Ma, H. (2010). Robust head-shoulder detection by pca-based multilevel hog-lbp detector for people counting. In *2010 20th International Conference on Pattern Recognition*, pages 2069–2072.
- [Zhao et al., 2006] Zhao, W., Jiang, Y.-G., and Ngo, C.-W. (2006). Keyframe retrieval by keypoints: Can point-to-point matching help? In *International Conference on Image and Video Retrieval*, pages 72–81. Springer.
- [Zhu et al., 2006] Zhu, Q., Yeh, M.-C., Cheng, K.-T., and Avidan, S. (2006). Fast human detection using a cascade of histograms of oriented gradients. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1491–1498. IEEE.