Chapter 1

INTRODUCTION

Camera vision and machine intelligence have been the foci of researchers since the invention of computers & Cameras .Over recent years, researchers have been working on replacing humans with computers to take over the time-consuming tasks with more accuracy .One of the key areas is Human detection from images and videos .Several areas of applications have spurred the interest of human detection such as human computer interaction for video games, robotics, video surveillance ,smart vehicles etc. However, human detection is a challenging problem due to the huge intra-class variation stemming from color , clothing, pose and appearance variations etc .Furthermore, external conditions such as partial occlusions ,illumination and background clutter further increase the problem.

In order to overcome & handle these challenges ,the work on human detection are categorized under 2 approaches - Feature development and Classifier development.

Humans can either be detected wholly or by parts. There have been numerous features proposed by researchers for holistic human detection such as Haar Wavelet features [35], Haarlike features and motion information [46], edge templates [12],Implicit Shape Models [21], Adaptive Contour Features [11],[27], Histogram of Oriented Gradients (HOG) [3] and the Covariance descriptor [45]. Holistic human detection , in general, underperform when there are heavy occlusions or extreme pose variations in the image.In contrast to holistic human detection methods, parts-based human detection is able to handle occlusions more robustly. The main concern in parts-based methods is high false positives number. Hence, most research in this area is devoted to determining robust assembly methods to combine detections and eliminate false positives. Furthermore, for good performance, high resolution

images are needed to extract sufficient and robust features for each human part. In practice, these kind of images are usually not available.

Feature development methods for human detection usually use classifiers such as Support Vector Machines (SVM) and cascade-structured boosting-based classifiers. Among methods that use SVM classifiers, linear SVM classifiers are generally preferred for speed and to minimize the over fitting problem of non-linear SVM kernels.

Some work has been done on classifier development for human detection. The reason for such a direction stems from the problem of large intra-class variation of humans. Building features that can handle the large intra-class variation of humans is difficult. However, since classifiers use statistical methods for classification, intra-class variations can be better controlled. Some popular classifier frameworks include Cluster Boosted Tree, Intersection Kernel for SVM, Multiple Instance Learning, "seed-and-grow" scheme.

HOG is the most popular feature used for human detection. It densely captures gradient information within an image window and is robust to small amounts of rotation and translation within a small area. It was introduced to solve the issue of differentiation of a bright human against a dark background and vice versa by Histogram of Gradients (HG) [28]. HOG maps all gradients of opposite directions to the same orientation. However, for the same cell,HOG also maps all gradients of opposite directions to the same orientation. As such, it is unable to differentiate some local structures and produces the same feature for some different local structures.

1.1 Goal of the thesis

The goal of the work in this thesis is summarized below:

- We used five Features in combination with four classifiers machine for human detection in mean standard deviation, the classifiers are first trained & tested10 times with 50% of data sets
- All classifiers will be analyzed with the combined feature setup for DT , ANN , SVM and kNN

1.2 RELATED WORK

There are a wide range of different methods and projects to automatically detect humans. All projects have their different approach and mainly focus to detect humans. Some are focusing on finding better sensors, some are focusing on creating better features, and others on creating better classifiers. Inputs from other works done on the subject are considered when choosing the appropriate method to be used.

A commonly used way to detect humans is to use some kind of conventional camera combined with a feature called histogram of oriented gradients (HOG)[1, 2, 3] or Haar-like features[4, 5]. More information about the environment can be obtained using other sensors like thermal imaging[6, 7], IR-ash[8] or depth sensing technology such as LADAR[9], Kinect[10, 11] or stereo vision[12, 13].

A common factor in the reviewed reports is the use of some kind of heuristic classifier. Commonly used classifiers are artificial neural networks (ANN)[14] and support vector machines (SVM)[1, 9].

Comparison of different detection methods are difficult to perform due to different data sets and varying evaluation methods but an attempt is made in [15, 16].A lot of research has also been done on using motion information to facilitate identification of moving objects. Kamijo et al. presents an algorithm for on-board monocular cameras that tracks foreground objects like pedestrians with the use of motion difference [17].A common method used with stationary cameras is to subtract the background to a moving human [18], however this method is not possible to use in this project due to the fact that the camera will be placed on a moving vehicle.

Chapter 2

RESEARCH BACKGROUND

In this chapter we will describe about the technologies and algorithms which we will be using for our Methodology, these basic description will help to understand our thesis result

2.1 Image Setup and RGB Image

Gray scale digital images are constructed as matrices where each element in the matrix represents a pixel. In case of 8-bit images, each pixel can take a value from 0 to 255, where 0 is a black pixel and 255 a white. A RGB image can be constructed with the use of three such 8-bit matrices where each 8-bit channel represents one of the three colors; red, green and blue. An RGB color space is any additive color space based on the RGB color model. A particular RGB color space is defined by the three chromaticity of the red, green, and blue additive primaries, and can produce any chromaticity that is the triangle defined by those primary colors.[2]

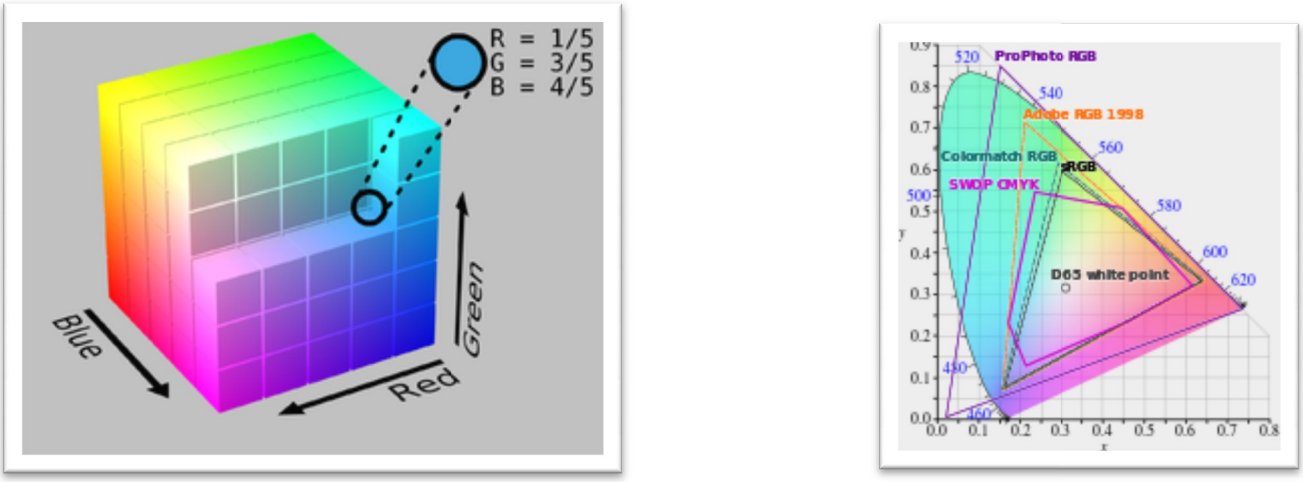


Figure 2.1 Illustration of RGB color space

2.2 YCbCr COLOR SPACE

YCbCr is a family of color spaces used as a part of the color image pipeline in video and digital photography systems. Y′ is the luma component and CB and CR are the blue-difference and red-difference chroma components. Y′ (with prime) is distinguished from Y, which is luminance, meaning that light intensity is nonlinearly encoded based on gamma corrected RGB primaries.Y′CbCr is not an absolute color space; rather, it is a way of encoding RGB information. The actual color displayed depends on the actual RGB primaries used to display the signal. Therefore a value expressed as Y′CbCr is predictable only if standard RGB primary chromaticities are used.

YCbCr is a digital color system, and a nonlinear color space which can be transferred from RGB color space. It has been widely used as a standard for video compression, and has since become a widely used model in digital video. RGB (Red-Green-Blue) values can be transformed to YCbCr color space using (1). Given that the input RGB values are within the range of [0,1] the range of the output values of the transformation will be [16,235] for Y and [16,240] for Cb

and Cr. The corresponding reverse operation is shown below

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 65.481 & 128.553 & 24.966 \\ -37.797 & -74.203 & 112 \\ 112 & -93.786 & -18.214 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1)$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 0.00456621 & 0 & 0.006625893 \\ 0.00456621 & -0.00153632 & -0.00318811 \\ 0.00456621 & 0.00791071 & 0 \end{bmatrix} \left(\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} - \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \right) \quad (2)$$

2.2.1. REASONS FOR CHOOSING YCBCR COLOR SPACES

Fig. 2.2.1 demonstrates the distribution of Cb-Cr plane of skin pixels we have generated in training set. From the figure we can observe that the skin pixels exhibit similar Cb and Cr values which cluster in a very specific region. That illustrates the reason why we prefer this color space for skin detection. Furthermore, it has been shown that skin color model based on Cb and Cr values can provide a good coverage of all human races [5]. This is based on the conjecture that different skin colors that viewers perceived from the image cannot be differentiated from the chrominance information of that image region. The apparent difference in skin color that viewers perceived is mainly due to the darkness or fairness of the skin. These features are characterized by the difference in the brightness of the color, which is governed by Y but not by Cb and Cr. We use (3) to get its Gaussian Mixture Model (GMM).

$$p(CbCr \mid skin) = (2\pi)^{-1} \mid \Omega_s \mid^{-d/2} e^{-(c-\mu_S)^T |\Omega_S|_S^{-1}(c-\mu_S)/2} \qquad (3)$$

Where $\mu_s = \begin{bmatrix} m_{Cb} \\ m_{Cr} \end{bmatrix}, \Omega_s = \begin{bmatrix} \sigma_{Cb}^2 & \sigma_{Cb}\sigma_{Cr}\rho \\ \sigma_{Cb}\sigma_{Cr}\rho & \sigma_{Cr}^2 \end{bmatrix}$, ρ is the

P is the cross-correlation coefficient. The corresponding figure is also shown in Fig. 2.2.1

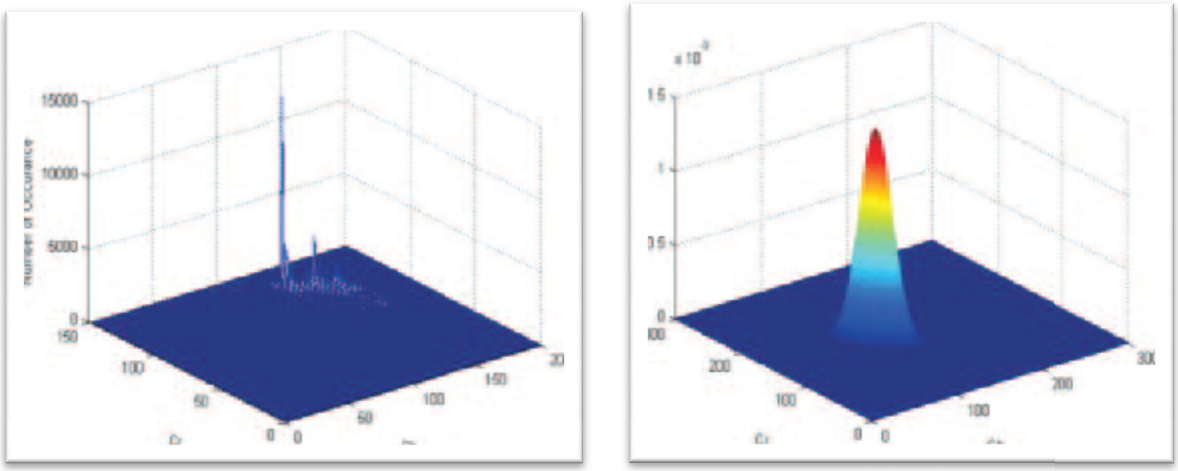


Figure. 2.2.1. Distribution of Cb-Cr of skin pixels and its Gaussian Mix

2.3 CLASSIFICATION

In contrast to deterministic classification, the heuristicall method uses \trial and error" to solve a given problem and uses training data in order to give the classifier the information it needs. This makes the classifier highly dependent on how well the training data represents the actual classification problem. The detection of a human is a complex problem. If deterministic methods were to be used the problem would probably need to be simplified to be able to prove that the problem is solved. The simplification means that the solved problem is not the actual problem but a simplified version of it. The simplification of the problem is not needed when using a heuristic method. The negative aspect of the heuristic method is that it is hard, if not impossible, to actually prove that an optimal solution is found. It is also often hard to know which information the classifier is using when the classification is done [25].

2.3.1 TRAINING A CLASSIFIER

This report will only describe the very basics of what is needed for a computer to be able to learn and to classify images. It will not cover the mathematics behind the learning theory or the used learning algorithms. The purpose of the classifier is to determine which class an image belongs to, in this case: human or not human. To be able to do this, the classifier is trained with a data set. A commonly used method is called supervised learning which is implemented by training the classifier with a set of images and the classes they belong to. Pre-processing and features help the classifier to interpret the images. Figure 2.3.1 shows the training procedure.

The classifier uses the inputs and the class labels of the training data to optimize classifier dependent internal parameters to correctly classify the training data. A common problem with supervised learning is over fitting, that is when the classifier becomes to specialized towards the training data. Over fitting is dealt with differently depending on the classifier but usually cross validation is used. Cross

validation means that some of the labeled data are used for validation instead of training. If the classification accuracy is improved for the training set but reduced for the validation set, the classifier is becoming to specialized and training should be stopped [26, 27].

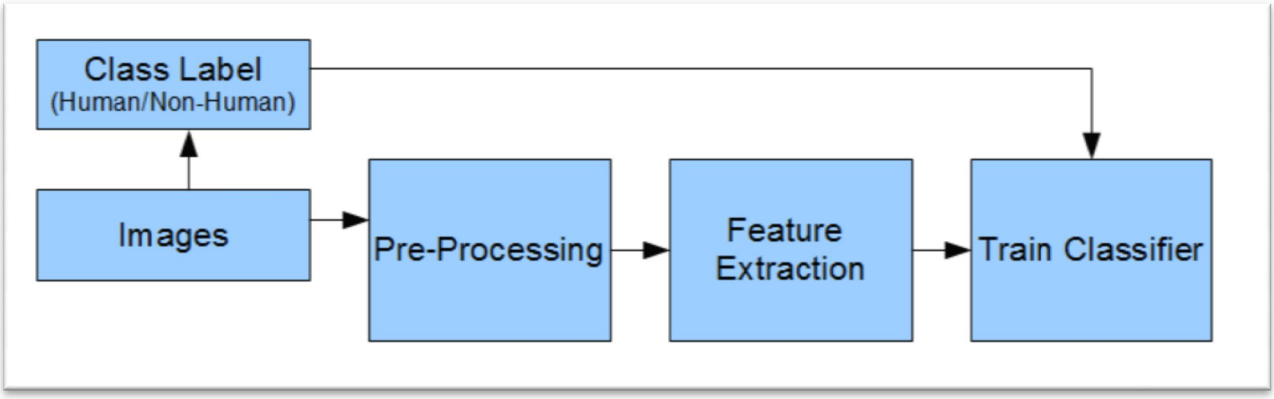


Figure 2.3.1: The procedure when training a heuristic classifier. The training set is preprocessed, features are extracted and presented to the classifier along with a class label that tells the classifier whether an image contains a human or not

2.3.2 PRE-PROCESSING ALGORITHMS

This section contains the theory behind three of the images created in the pre-processor, see Figure 3.2.4. The purpose of having the pre-processor, instead of doing these computations as a part of the feature, is to make sure that these computations are only executed once.

CONTOUR IMAGE

When working with object detection or object recognition it is often preferred to find the contour of the object of interest. The first step in finding the contour is to create a binary image. The binary image is then systematically scanned until a component is found. When a component is found its edge is followed and stored before the systematic scanning continues. This procedure continues until the whole image is scanned and all contours are stored [28]. In this project, the largest contour is kept and the rest discarded. The procedure of finding the contour is shown in Figure 2.3.2.

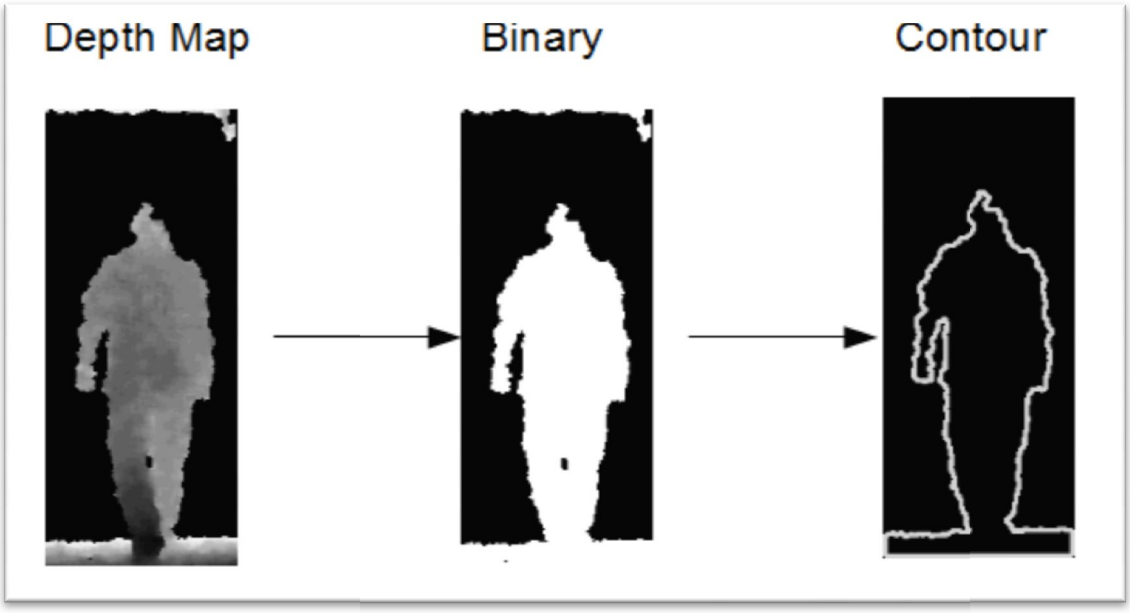


Figure 2.3.2: Illustration of the steps from depth map to contour image of a human

Integral image

The purpose of the integral image is to speed up calculations of features that uses rectangular filters, that is the sum of rectangular areas of an image .The integral image is used as a lookup table. It is constructed so that the element with index (i, j) in the integral image corresponds to the sum of all elements inside a rectangle with the corners (0, 0), (i, 0), (0, j) and (i, j) in the original imageThe use of integral images is only beneficial if the total area of the used rectangles are greater than the area of the image, or more precise, if the computations used to sum the total area of the used rectangles is less than the computations used to construct the integral image and perform the look ups [29].

2.3.3 FEATURES

This section explains some background theory of the used methods to extract features. In this report, a feature is defined as one value. Each method generates a feature vector that consists of several features. The word feature is sometimes also referring to a feature extraction method.

FOURIER TRANSFORM

The Fourier transform is an important tool that can be used in image processing to decompose an image into sine and cosine components. The output of the fourier transform will be the image representation in the frequency domain from which both the amplitude and the phase can be extracted as shown in Figure 2.3.3. The discrete fourier transform (DFT) is the sampled version of the fourier transform and is used for digital images.Using the frequency domain representation of the image, numerous operations can be done with different kind of image processing purpose [19]. In this report the phase and amplitude are used by the classifier without further processing.

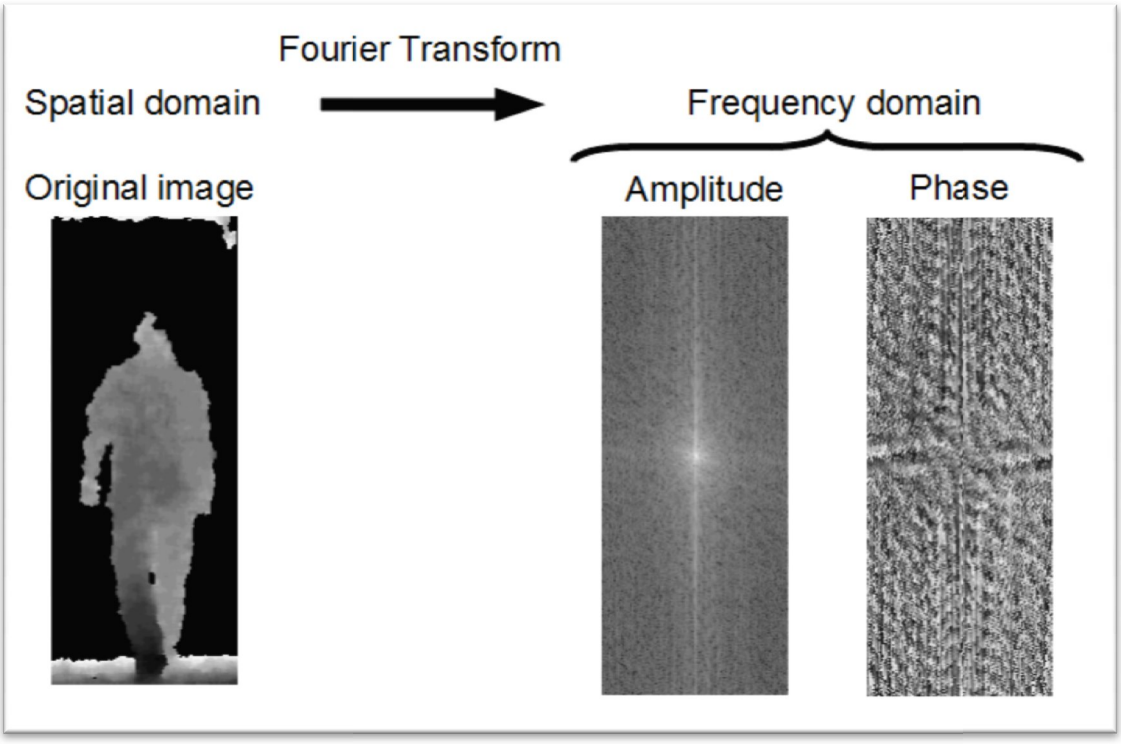


Figure 2.3.3: Illustration of the fourier transform. The amplitude and the phase are the output from the fourier transform. The shown amplitude image is in logarithmic scale. Both the amplitude and phase image are normalized.

FOURIER DESCRIPTORS

Fourier descriptors (FD) is a shape based feature that utilizes that the shape can be roughly described by the low frequency components of the contour. The contour is given by N points in a two-dimensional space. To be able to use the fourier transform on these points they are represented as complex number with the x-coordinate encoded as the real part and the y-component as the imaginary part. The resulting frequency spectra (F0, F1 ... F(N-1)) contains information about the shape, low frequency contains information about the general shape and high frequency describes finer details. For classification general information is desirable and for some classification methods, like neural networks, it is important to have the same number of inputs so a fixed number of the lowest frequencies are used to construct the feature vector. Figure 2.3.4 illustrates how much information that is contained in the coefficients.

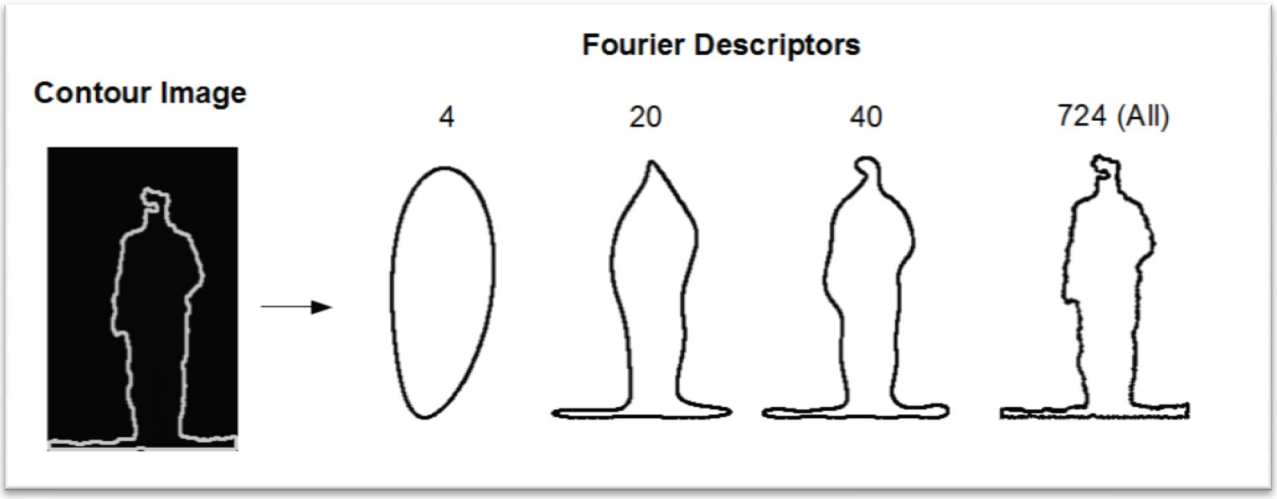


Figure 2.3.4: Illustration of the contained information in the coefficients. Fourier transform is applied on the contour image to the left. Then a number of low frequency coefficients are used to create the images to the right via inverse Fourier transform. The number above the images tells how many coefficients that are used.

An advantage of FD is that they are rotation invariant. It is also possible to make them translation and scale invariant. All translation information is contained in F0, the constant component. Thus FD is made translation invariant by setting F0 to zero. The scale invariance can be realized by dividing each component by F1 [30]. In this project 40 of the low frequencies are used to create the feature vector.

IMAGE MOMENTS

The image moments are shape descriptors that often play an important role in object recognition. There are three kind of moments that are used in this project; spatial moments, central moments and central normalized moments. These are calculated as shown in Equation 2.3.1 - 2.3.3 and uses a filled version of the contour image from the pre-processor [19].

$$M_{ji} = \sum_x \sum_y x^i y^j f(x,y) \tag{2.3.1}$$

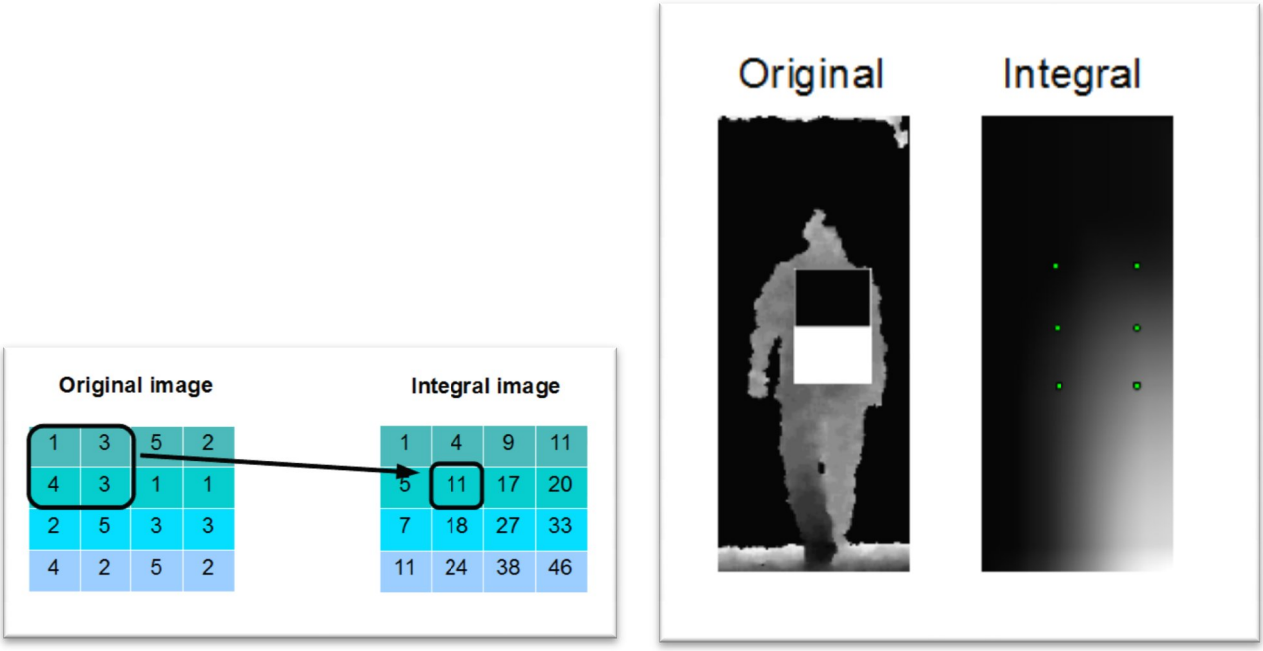
$$\mu_{ji} = \sum_x \sum_y (x-\bar{x})^i (y-\bar{y})^j f(x,y) \tag{2.3.2}$$

$$\eta_{ji} = \frac{\mu_{ij}}{\mu_{00}^{(1+\frac{i+j}{2})}} \tag{2.3.3}$$

Here, f(x,y) represents the intensity of an image and x and y is the pixel's position in the image. From these equations 24 moments have been proven to be useful and are shown in [31]. The central normalized moments calculated by Equation 2.3.3 are shown to be both translation and scale invariant. Another interesting moment feature that is often being used but not in this report are the Hu moments. These moments are invariant under translation, changes in scale and also rotation [19]. These are not used due to that they did not perform well.

HAAR-LIKE FEATURES

Haar-like features compare the pixel intensities at different parts of an image usually using rectangular filters.A rectangular filter is just the sum of all pixels inside a rectangular area of the image. The features are derived by adding or subtracting one or several of these sums [32]. In this project, several rectangular filters of random size are randomly placed. Then a random number between 2 and 6 of these are either added or subtracted from the final feature value. The features are created at random but the random generator is initialized with the same seed for all images, that is, the features used on the training set is the same as the ones used on the images that are classified



(a) How to create integral image. *(b) How to use integral image.*

Figure 2.3.5: (a) How the integral image is created. Each number is the value of a pixel. Each element in the integral image is the sum of all elements inside a rectangle with upper left corner in origin and the down right corner at the corresponding element from the original image. (b) How the integral image could be used. In this case the wanted feature is the value of sum of pixels inside the white rectangle subtracted by the sum of all pixels inside the black rectangle. If you have the integral image it is possible to calculate this value by using the points shown in the integral image. These points corresponds to the corners of the rectangles in the original image.

HISTOGRAM OF ORIENTED GRADIENTS

HOG is a widely used feature for pedestrian detection in color/grayscale images. This method uses intensity gradients for each pixel to represent shape information.A typical setup is to have an image size of 128x64 pixels and to divide the images into 8x8 pixel cells. The gradient is computed for each pixel in the cell and is used to build a histogram where the gradient orientation is divided into 9 bins in the range 0-180_. The cells are divided into larger blocks and their corresponding histograms are normalized within the block to take into account local variation of lighting. Each cell typically belongs to four blocks and will hence contribute with four versions of its histogram to the final feature vector, normalized in four different ways [2]. Figure 2.3.6 shows the procedure

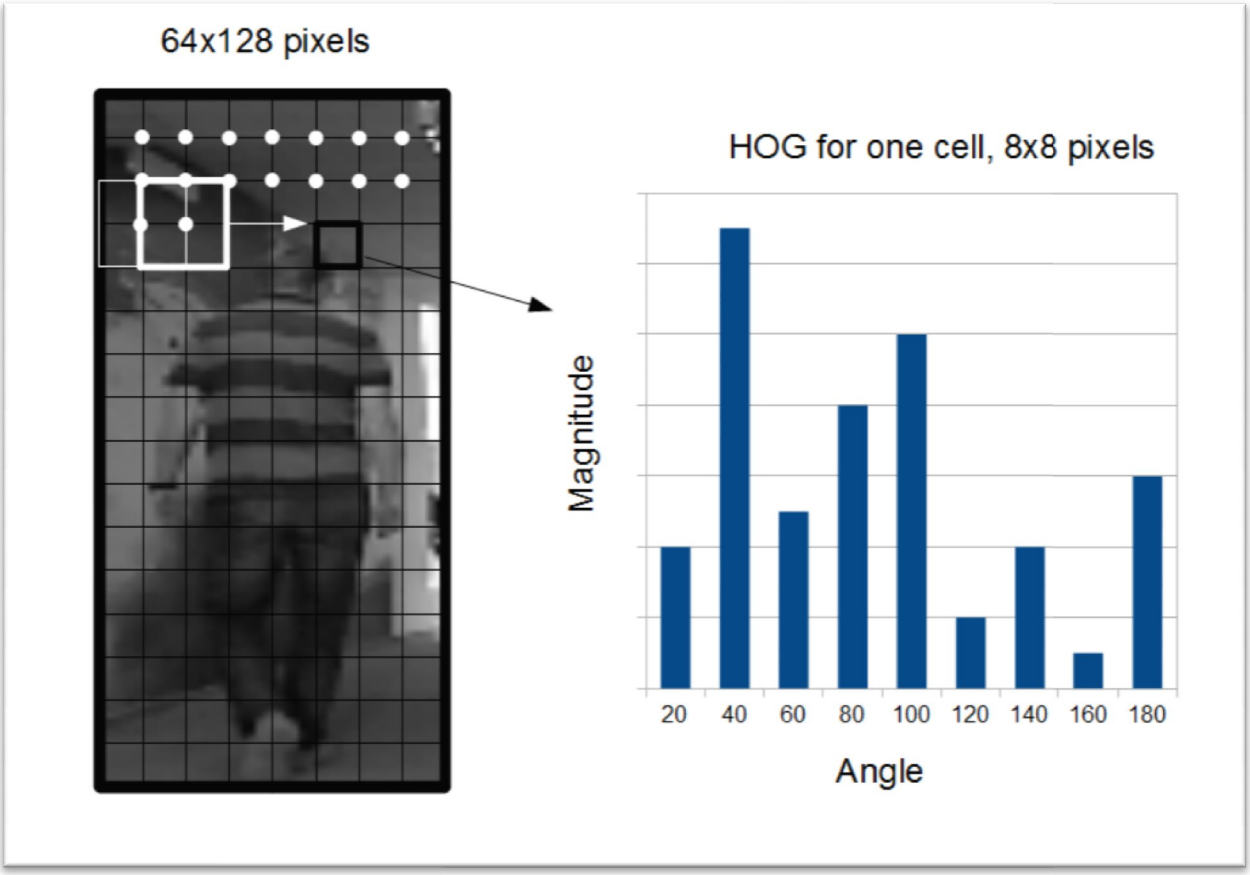


Figure 2.3.6: An illustration of how HOG works. The whole picture is 64x128 pixels and is divided into many 8x8 pixel cells. Each cell has a small corresponding HOG. The white square, called block, of size 16x16 pixels, covers four cells for which the histograms are normalized and sent to the final feature vector.

OVERVIEW OF HISTOGRAM FEATURE

The overview histogram feature was created with some inspiration from [33]. The purpose is to make a 2D -histogram of the depth map image by removing the Y-axis and projecting all points to the XZ-plane as shown in Figure 2.3.7. This means that if a point would be at (1,2) with depth value (Z) 3 it would increase the 2D histogram at point (1,3) by one.

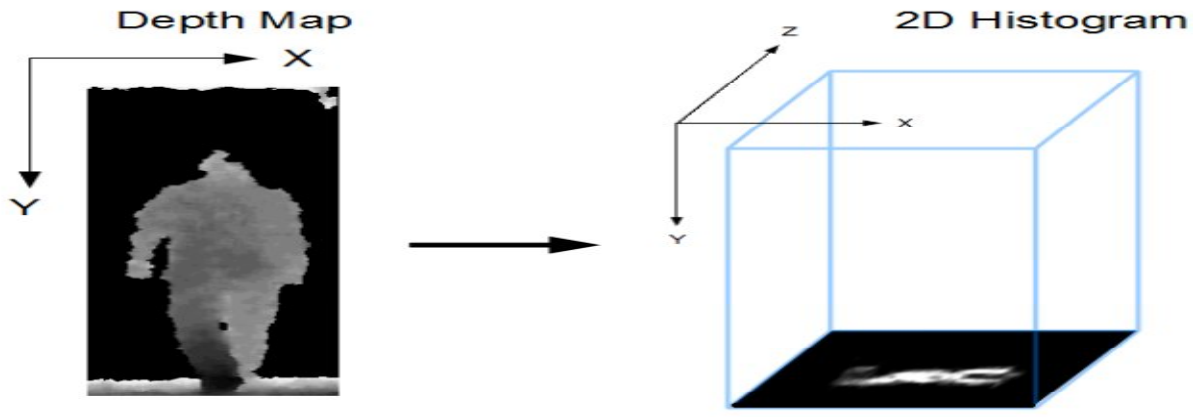


Figure 2.3.7: How the overview histogram is created. The histogram is the gray scale image in the bottom of the box. The whiter the pixel the more corresponding positions from the depth map exist.

After the histogram is created it is smoothed, threshold, contours are found and finally a rectangle and a ellipse are fitted to the contour as good as possible. The values from the width and length of the rectangle and the ellipse are used to create the feature vector. Figure 2.3.8 shows this procedure.

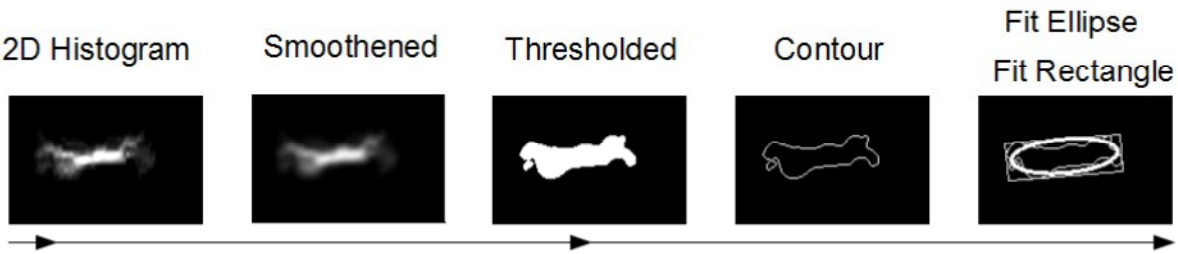


Figure 2.3.8: How the overview histogram is used. The final feature vector consist of the length and width of the ellipse and rectangle.

Unfortunately this feature did never work since it was discovered that the resolution of the depth, the Zaxis, decreased the further away an object was. The feature was included in the report anyway because it seems promising if the problem with the resolution could be fixed, this is further discussed in later chapters

2.3.4 CLASSIFIERS

This section contains a brief description of the analyzed classifiers. These classifiers are machine learning classifiers that all are trained using supervised learning.There are two different types of supervised learning models, classification classifiers and regression classifiers.Classification classifiers map the input space to a finite set of predefined classes, whilst regression classifiers map the input space to a continuous real valued output space [34].

K-NEAREST NEIGHBOR

The k-nearest neighbor (kNN) is a very simple classifier but is in many cases effective. The training procedure only consists of storing the features from the training data. When a new instance is to be classified it is placed in the feature space and is then given the label of the most common labels of the k nearest neighbors [35]

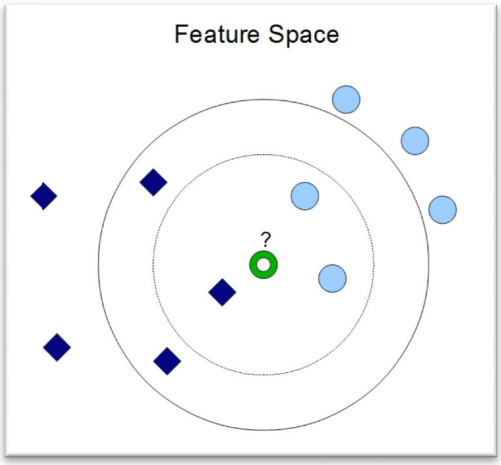


Figure 2.3.9: Example of how kNN-classifier works in a 2D feature space. The ring in the middle is to be classified as either a circle or a square. The already existing circles and squares are the used training data.How should it classify the ring in the middle? If k = 3 it would classify the ring as a circle but if k = 5 it would classify the ring as a square.

SUPPORT VECTOR MACHINE

SVM are primarily intended for two-class classification and uses a quite straight forward method. During the training phase, the SVM tries to find a (N-1)-dimensional hyperplane that separates the N-dimensional training data into the two classes with maximal margin, that is, with maximal distance to all nearby samples. A sample can then be classified depending on where the sample is located in relation to the hyperplane, see Figure 2.3.10. When the problem is non-linear it is impossible to separate the classes with a linear hyperplane. The solution is to map the problem into a higher dimension space2 which often makes the problem easier to separate in that space. The training samples that lie closest to the hyperplane are the only ones that affect the position of the hyperplane, hence they are called support vectors as they \support" the hyperplane. The other training samples do not contribute to the training. This makes the SVM somewhat vulnerable to noise which motivates the introduction of a soft margin that allows some of the training samples to lie inside the margin or even be misclassified [36]

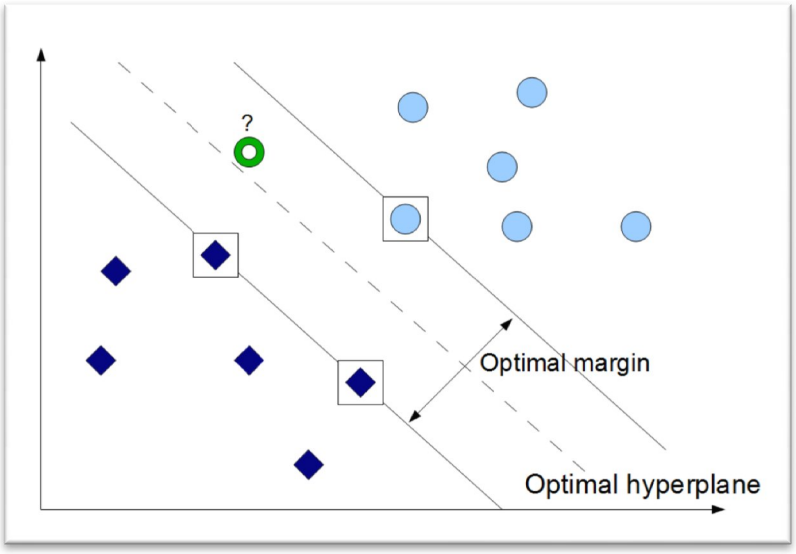


Figure 2.3.10: Example of how SVM classifier works in a 2D feature space. The already existing circles and squares are the used training data. The support vectors are marked with a square and lying on the line. The ring is in this case classified as a circle.

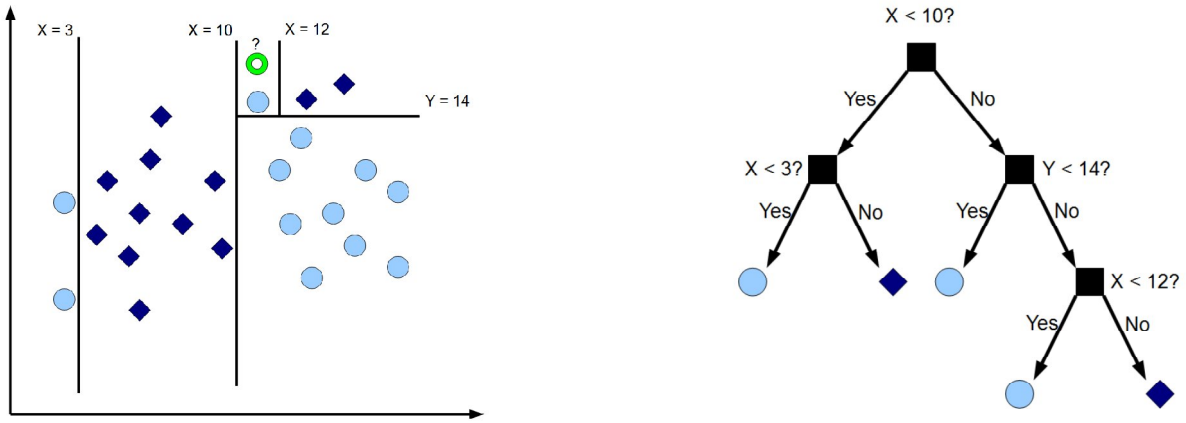
DECISION TREE

A decision tree (DT) consists of a tree of linked decision nodes and can be either a classification tree or a regression tree. The output of a classification tree is which class the input belongs to and the output of a regression tree is a floating point number which gives an indication of the probability that the input belong to some class. The input vector can consist of both continuous valued variables and discreet case variables. In the case of continuous valued variables the node checks which range a sample belongs to. The tree structure is created at the training stage. A set of labeled input vectors is recursively split into subsets until some stopping criteria is met, for instance if all samples in a subset belongs to the same class. Each split corresponds to a node in the tree, the first split is done in the root node, the second in one of its child nodes and so on. The split is based on the input parameter that can order the subsets the most which means that the depth of the node indicates how inertial the parameter used by that node is. The root node is most important and the importance decreases with each generation [34].

ARTIFICIAL NEURAL NETWORK

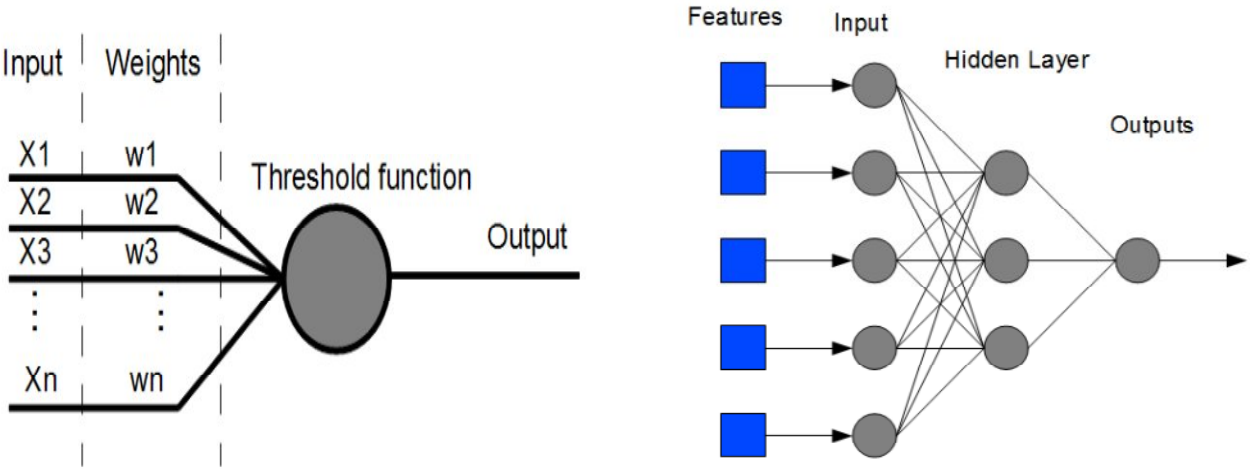
An ANN consists of neurons and connection between the neurons which are supposed to imitate a human brain. A neuron can take an arbitrary number of inputs, which are weighted individually and summed. The summed value is compared with a threshold function and the threshold value is weighted and then sent as an output. This output could either be the final output or used as input to one or several other neurons [37]. The neurons can be connected in many variations but the most common is that they are placed in layers and this is the used type of network in this project. Figure 2.3.12 illustrates a

neuron and an example of a network. This type of network is called a feed forward network which is arranged so that the output of the neurons in one layer are used as input in the next layer [37].



(a) (b)

Figure 2.3.11: Example of how DT classifier works in a 2D feature space. The already existing circles and squares in (a) are the used training data. The lines in (a) shows how the tree in (b) split the features. The method to classify the ring is just to follow the tree in (b) from the top until a class is found, in this case a circle.



(a) A neuron (b) A neural network

Figure 2.3.12: Example of how an ANN-classifier works. An illustration of a neuron can be seen in (a). In (b) an example is shown of how input and output can be connected together with one hidden layer. The numbers of hidden layers and neuron in the hidden layer can be changed.

Chapter 3

RESEARCH METHODOLOGY

In this chapter we explain the research methodology followed in this study. We describe the phases involved in the research methodology.

3.1 Setup

This section presents under which circumstances the project where performed , we collect several data sets of pedestrian which were taken by stereo or kinect type of sensors supported cameras from different locations.

3.1.2 SOFTWARE

The software used during this project are listed in Table 3.1.1. The C++ environment from MS Visual studio combined with OpenCV has been the main tools to solve the problem. MATLAB was added as a means of presenting data graphically and OpenNI is used to receive the Kinect image from the sensor.

Table 3.1.1: The software used in this project.

Software	Purpose	Comment
MS Visual Studio 2010 [39]	C++ environment	
MATLAB R2010b [40]	Plotting graphs	
OpenCV V 2.3.0 [41]	Image function	An open computer vision library.
OpenNI V 1.5 [42]	Kinect image grabbing	An open source device interface library.

3.2 CLASSIFICATION

Theory behind heuristic classification is described in section 2.3. This section contains the implementation of segmentation, pre-processing, feature extraction and classification. Figure 3.2.1 shows how these parts are connected. To get a working classifier it has to be trained before it can be used for classification, this is described in section 2.3.1.

3.2.1 SEGMENTATION

The sensor provide both a color image and a depth map image to the segmentation.The segmentation is performed by first dividing the 3D space into depth layers with a thickness of 1000 mm and an offset of 250 mm, meaning that the layers overlap. Equation 3.2.1 is used as an approximate way of determining the pixel height corresponding to 2100 mm at different depths. The constant C is calculated by using a measured value of the pixel height (442 px) at a depth of 2500 mm. The calculated height and an aspect ratio of 2.5 are used to construct a rectangle at each depth

Figure 3.2.1: Overview of the procedure to classify a human.

for which the pixel dimensions corresponds to the real dimensions: height = 2100 mm and width = 840 mm.The rectangle is used as a window which is swept, sometimes called sliding window, over the depth layer with some overlap, see Figure 3.2.2, and each resulting segment corresponds to a 3D box which is further processed.As the approximate size of a human is known at each depth, the percentage of depth pixels that should lie inside the box when the box contains a human is also known. Boxes that

contain little or no depth pixels at all can therefore be discarded without further processing which saves a lot of computations.

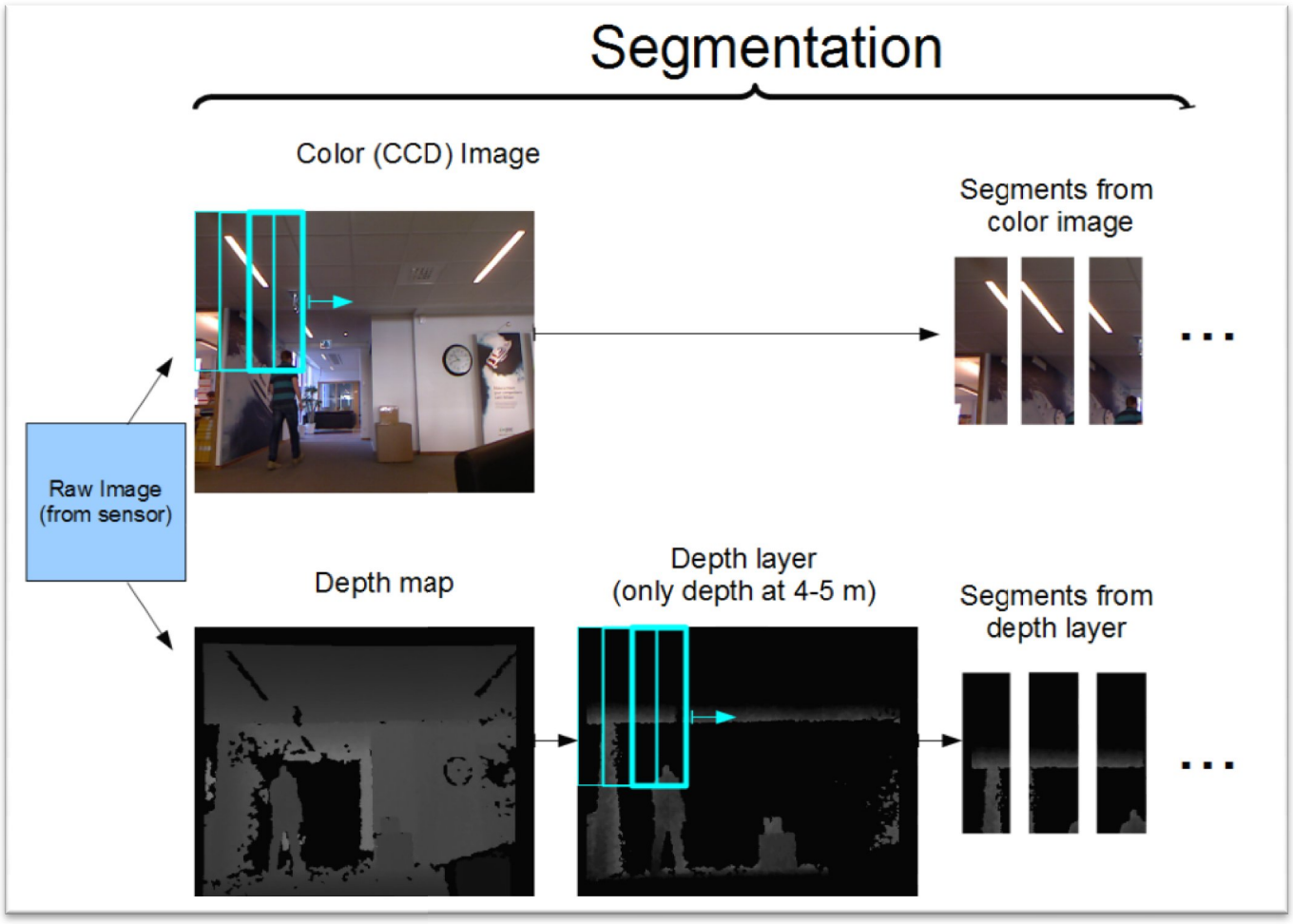


Figure 3.2.2: Overview of the segmentation procedure. First two images, the color image and depth map, are received from the sensor. A depth layer is extracted and a rectangle is swept over the images in order to obtain the segments from both the depth map and the color image. Then a new depth layer is extracted and the procedure continues until the whole space is segmented.

The classifier assumes that all segments have the same size. The solution used is that the change of window size is actually implemented by changing the size of the original image and keeping the window size fixed. In this way if the distance is swept from 2500 mm to 7500 mm about 2000 segments are created of which as many as 1800 segments contain little or no depth information and can be discarded1. This means that about 200 images are further sent to the pre-processor.

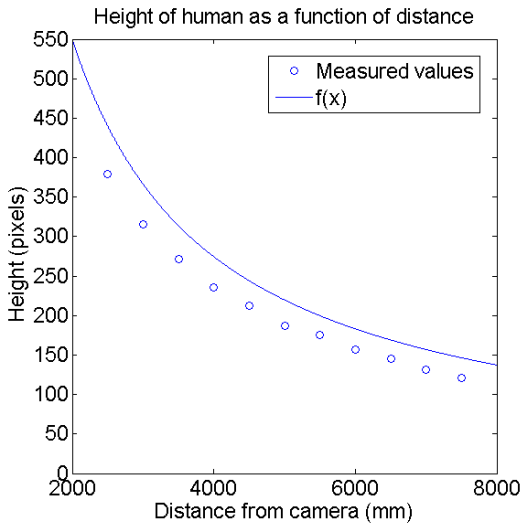


Figure 3.2.3: The height of a human as a function of the distance to the camera. The function in Equation 3.2.1, used to convert the real height of 210 cm at a certain distance to pixel height, is compared to measured values of a 180 cm tall person.

3.2.2 PRE-PROCESSING

The pre-processor receives two types of images from the segmentation part. These are the segments from the color image and the segments from the depth map. During the pre-processor step the color segments and depth segments are transformed to other types of images to make it easier for the feature extractor to extract the feature. Another aspect is that multiple feature extractors can use the same image type without having to recompute that image. The types of images created and sent forward from the pre-processor step are shown in Figure 3.2.4. Two of the image types are a gray scale image, which is derived from the color segment, and the color segment itself. The depth layer segment is converted to three other types of images; binary image, contour image and an integral image, which are described in section 2.3.2. These three, along with the original segment is sent forward to the feature extractor. Altogether there are six types of images sent from the pre-processor to the feature extractor

3.2.3 FEATURE EXTRACTION

This section will describe how the features are used. A more detailed description on how these features work can be found in section 2.3.3. Every feature uses one or more images from the pre-processing part. The implemented features along with their used image type can be seen in Table 3.2.1.

3.2.4 CLASSIFIERS

Four different heuristic classifiers are being analyzed, these are kNN, SVM, DT and ANN. How these work are described in section 2.3.4.

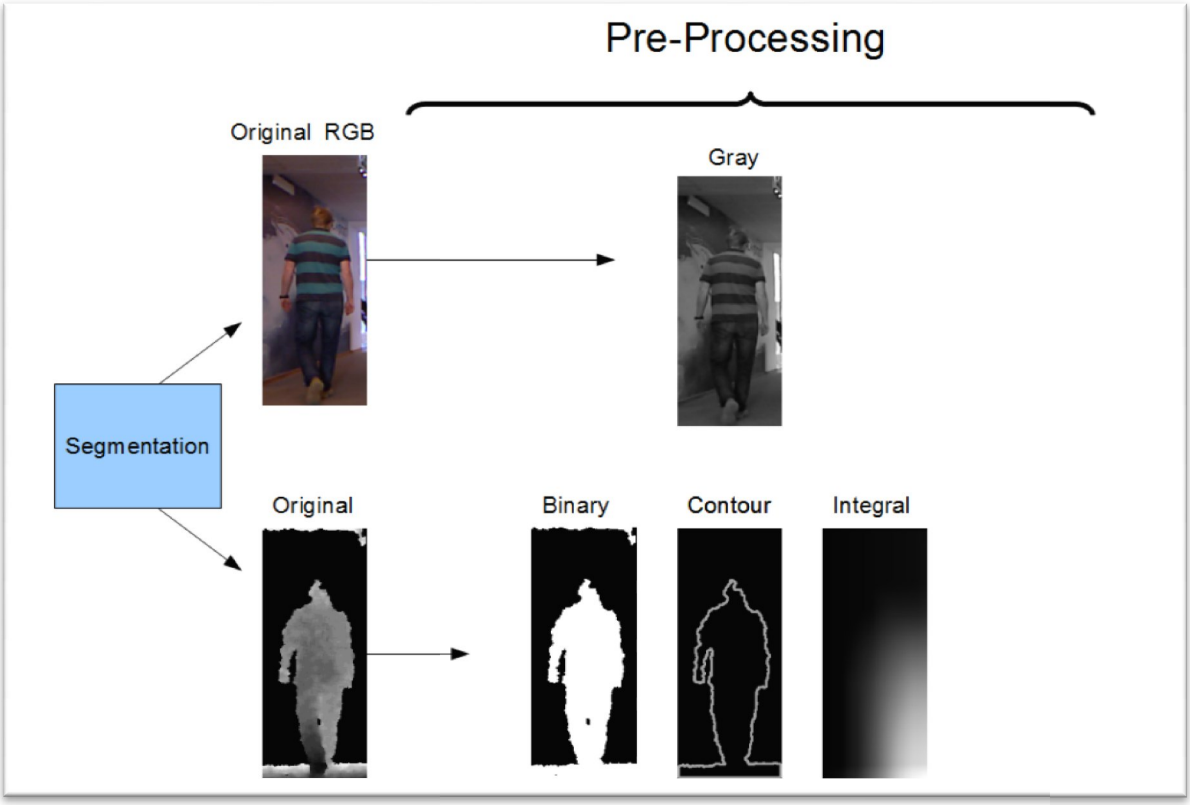


Figure 3.2.4: Overview of the pre-processing procedure. The binary image is not used in any feature itself but is used to determine whether a segment contains enough information to be a human. It is also used to derive the contour image

Table 3.2.1: The most interesting feature extraction methods.

Feature	Used Image	Comment
Depth Map	Depth Map	Depth map is downsampled
DFT	Depth Map	
FD	Contour Image	
Moments	Contour Image	
Haar-like features	Integral Image	
HOG	Gray Scale Image	
Overview Histogram	Depth Map	Does not work yet

There are several parameters that can be customized for each method and the purpose is to make them as good as possible for the chosen features. The problem here is that it is impossible to know if the parameters are optimized for a certain problem. This mean that it can be hard to compare classifiers if the parameters are badly chosen. The goal is to make the conditions for the classifiers as similar as possible. The base for all of these classifiers were already implemented in OpenCV. This did save a lot of time when creating the classifiers and also made it possible to use some classifiers even when the experience in those classifiers were low.

Combination of Features

Each feature has its own classifier in a \pre-classification" stage. When two or more features are used a new classifier of the same kind is created and uses the output from the \pre-classification" stage as input. This procedure is shown in Figure 3.2.5. This method makes it very easy to add and analyze additional features.

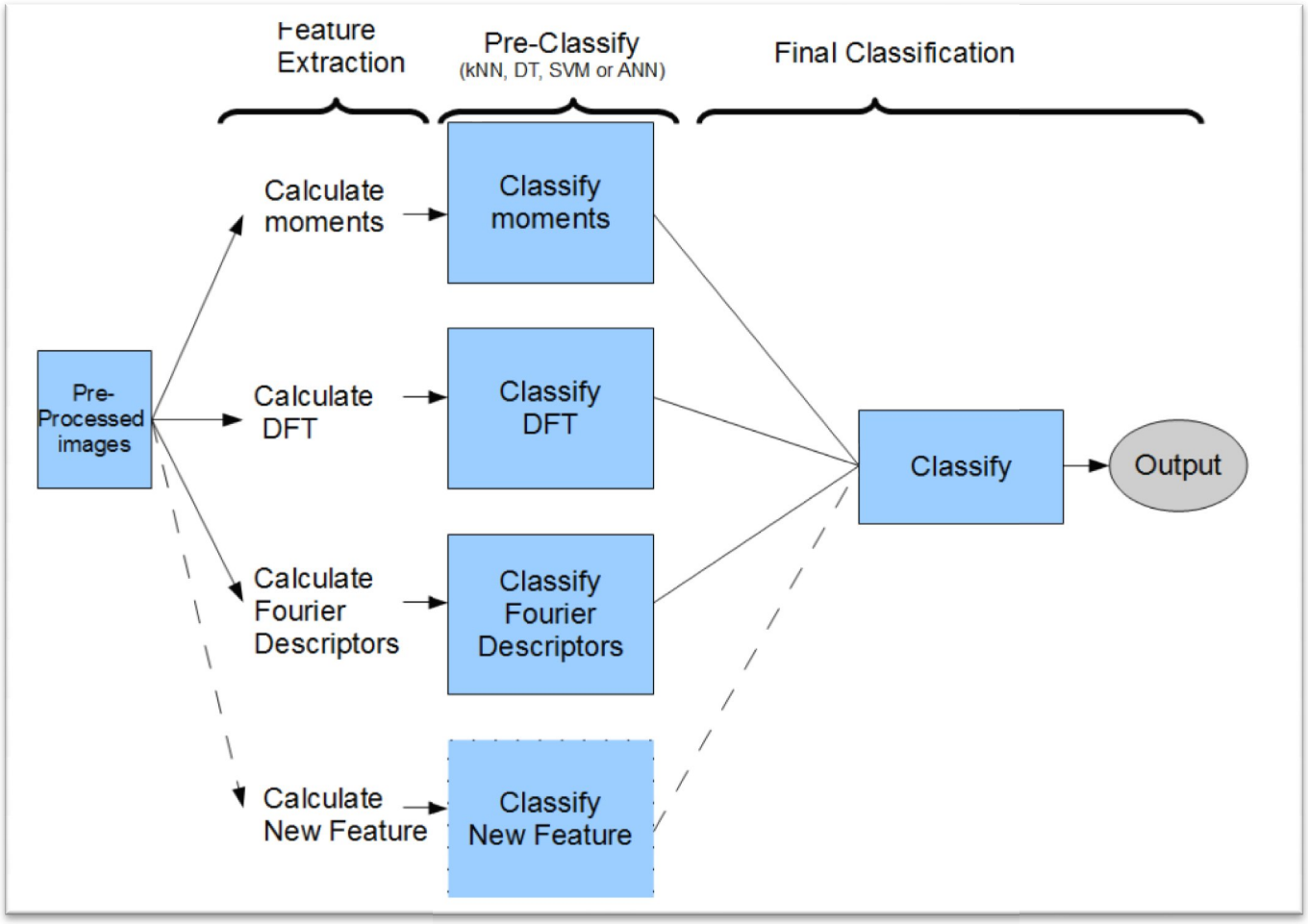


Figure 3.2.5: Shows the implemented setup which makes it easy to add and analyze new features.

3.2.5 DATA MINING

The data set is divided into two parts, a training set and a test set. As the name indicates the purpose for the training set is to train the classifier and the purpose of the test set is to test the performance of the classifier. The positive training set contains 1100 images of four persons and the positive test set contains 800 images of four other persons. The negative test set were collected by capturing 1500 images from an environment that was not used for the training set. The method to collect the negative training set follows a different procedure. First 3500 negative training images were captured at four different places. The classifiers were then trained and used in an environment without humans. Whenever an object in this environment were classified as a human it was added to the negative

training set. After several false classifications the classifier was retrained with the new training set and the procedure was repeated until a satisfying classifier was trained. At the end there were 5000 negative images.

3.3 PROOF OF CONCEPT

The purpose of this project is to implement an active safety system that should detect pedestrians from a moving vehicle. The implication of this is that the detector must be able to detect a pedestrian before a collision is inevitable or the consequences might be catastrophic. Image processing is often very suitable for parallelization, including this case. The same computations are done on several parts of the same image making it possible to run them concurrently. This will not make any difference if they are run on a single processing core. Running computations concurrently on a multicore processor however, can increase the computation speed considerably.

Multi-threading is implemented in the application to enable parallel classification of the different segments.A threadpool with a work queue is used to avoid unnecessary spawning of threads. When the image is segmented each segment is added to the queue and is processed as a thread becomes available.

Chapter 4

RESULTS

In order to evaluate the results of the thesis following measures have been used in this work.

This chapter presents the results from the evaluation. As we have assumed that camera used for capturing images is equipped with both Kinect & Stereo Vision sensors , after which the rest of the result chapter is focused on the image processing and algorithms to detect humans.

short list of advantages and disadvantages with Kinect and the implemented stereo vision system.

Kinect		Stereo Vision	
Advantages	**Disadvantages**	**Advantages**	**Disadvantages**
Night vision	Does not work in daylight	Works in daylight	Light dependent
Stable quality	Limited range, up to 9 meters	Works at long distance	Unstable quality of image
Sharp image	Problems with reflecting materials	Can see through glass	Needs calibration
			Texture dependant
			Computational heavy

4.1 EVALUATION OF ALGORITHMS

Early in the project the use of heuristicall classification algorithms was chosen. This makes it easy to change the classifiers intent by just changing the training data, for instance if classification of humans from a different angle or even classification of other objects would be wanted.

A comparison between this project and other projects is difficult due to differences in circumstances. The following results are therefore mostly interesting to be compared with each other. A comparison with other projects might not yield a fair result.

The results are presented as ROC curves and AUC values which are briefly explained in section 2.3.5. The results from the evaluation of algorithms are shown in Table 4.2.1. The AUC value should not be mixed up with the accuracy.

*Table 4.2.1: Performance of the different features using different classifiers. The result is presented as mean _standard deviation. The classifiers are first trained and tested 10 times on 50 % of the test set. These 50 % are randomly chosen each time. *The final classification stage in Figure 3.2.5 is a simple neural network and not a DT. A DT at the final classification stage could not create enough nodes to get a decent AUC.*

Feature	AUC for ANN	AUC for kNN	AUC for SVM	AUC for DT
Depth Map	0.905 ± 0.0079	0.971 ± 0.0038	0.882 ± 0.0105	0.965 ± 0.0047
DFT	0.922 ± 0.0065	0.944 ± 0.0028	0.940 ± 0.0035	0.939 ± 0.0028
FD	0.994 ± 0.0018	0.991 ± 0.0017	0.981 ± 0.0033	0.994 ± 0.0012
Moments	0.965 ± 0.0024	0.759 ± 0.0099	0.927 ± 0.0059	0.945 ± 0.0043
Haar-like features	0.962 ± 0.0057	0.916 ± 0.0065	0.937 ± 0.0044	0.972 ± 0.0040
Combination of Features	0.997 ± 0.0007	0.996 ± 0.0016	0.997 ± 0.0004	0.995* ± 0.0009

The values in Table 4.2.1 are highly dependent on how the training set and test set are formed. The method to create those is described in section 3.2.5.

4.2 FEATURE EXTRACTION

The performance of the features can be seen in Table 4.2.1. For most features there are at least some parameters that can be tuned to change the performance. The features are explained in section 2.3.3 and some choices of parameters are also stated there.The overview histogram is not included in Table 4.2.1 because the performance differed greatly depending on distance between object & sensors. The performance was very promising for distances shorter than about 5 m, but got increasingly worse with longer distances.. The reason why the overview histogram is kept in the report is because it is considered a very interesting feature that could perform very well if it is implemented correctly and the distance resolution problem is solved.

HOG is also not included because it was implemented as an already trained classifier. The performance of HOG on the test set gave an AUC of 0.925 which is a good value to compare with the other features. One should keep in mind that the HOG classifier is trained on a more general data set than the other used classifiers, which are only trained on the data that was generated in this project. The features in the list were the features that performed well and the most successful feature would be the fourier descriptor.

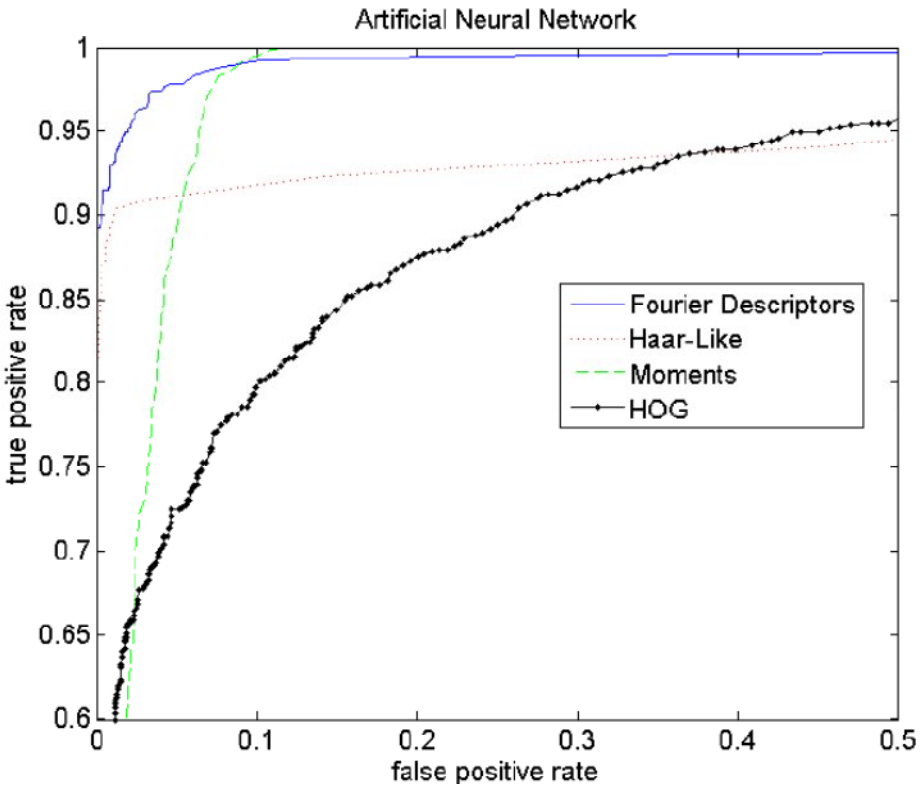


Figure 4.2.1 shows the ROC curves for four different features evaluated with an ANN classifier. It can beseen that the FD feature is the best performing feature of those four.

4.3 CLASSIFICATION METHOD

The four different classification methods used are ANN, SVM, kNN and DT which are explained in section 2.3.4. The kNN classifier is a simple classifier that actually performs quite well. Both kNN and DT were easy to configure. The ANN and SVM classifiers are more advanced than kNN and DT and

performs slightly better but is harder to configure.The result for the different classifiers shown in Table 4.2.1 differs a bit for each feature but is surprisingly similar for the combined version. All classifiers but kNN have several parameters that can be configured. The configurations that generated the results are optimized to some extent but can not be proved to be the optimal configurations. However it seems to be, as Dollfiar mentions in [16], that the choice of classification method plays very little role.

One of the best performing classifier was the ANN classifier with an AUC of 0.997. To get a better picture of how the classifier actually performs, one can look at the following example: Consider the ROC curve of the classifier as seen in Figure 4.2.2. The marked point of the curve have three values; X = 0.002, Y = 0.94 and Z= -0.4 which are false positive rate, true positive rate and the threshold. This means that 0.2 % of the false images are classified as true and 94 % of the true images are classified as true when the threshold is set to -0.4.

The confusion matrix for this threshold can be seen in Figure 4.2.3.

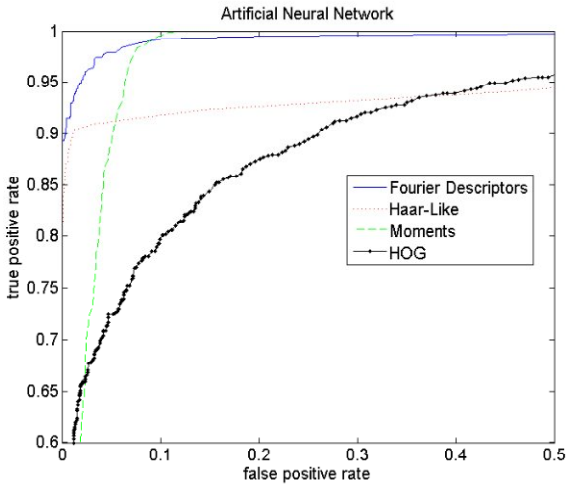


Figure 4.2.1: Four ROC curves for HOG and the three best performing features for the ANN. These features are FD, moments and Haar-Like features. The graph is zoomed in to make it easier to observe the differencesin the curves.

Consider that the segmentation will find about 200 negative segments each frame and the frame rate is 10 frames per second (FPS). This would be a total of 2000 negative images each second which would lead to about four misclassified images each second. If a human appear on in the environment the segmentation would probably have at least four segments of the human due to the overlap. This means that the probability to find a human where four segments are found would be 1 □ (1 □ 0:94)4 = 0:9999870 if these four segments would be independent. Of course many segments of the same human taken at the same time are not independent but the chance to find at least one of human in several segments is greatly increased. The conclusion of this is that the threshold could be increased to decrease the amount of false positives while still have a very high detection rate.

4.4 PROOF OF CONCEPT

The display for the proof of concept is shown in Figure 4.3.1 and the setup can be seen in Figure 4.3.2.

An important aspect when it comes to the proof of concept is the speed of the classifier. All classifiers

where analyzed with the combined feature setup and the FPS differed. DT was the fastest and achieved 13 FPS, ANN had 12 FPS, SVM had 4 FPS and the kNN had 0.5 FPS for k = 30 when running on a 2.2 GHZ Intel i5 quad core processor on windows7. Only minor optimization, such as multi-threading, has been done since a speed over 10 FPS was satisfying and the main focus was the classification performance.

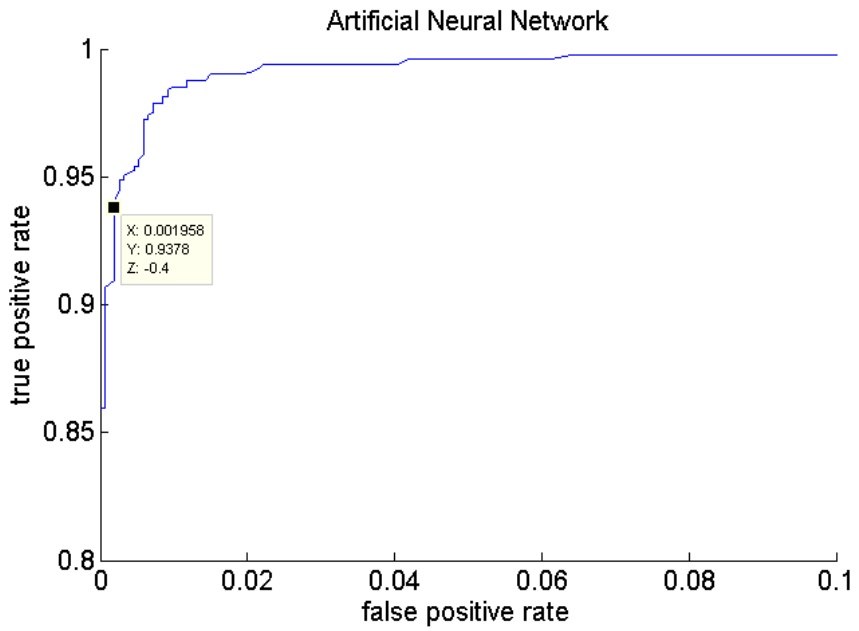


Figure 4.2.2: The figure shows a portion of the ROC curve for the ANN classifier when combining all features in Table 4.2.1. The X and Y value are the corresponding axes and the Z value represents the threshold at the given point.

		True Human: Positive	True Human: Negative
Hypothesized Human	Positive	**754**	**3**
	Negative	**50**	**1529**

Figure 4.2.3: The confusion matrix for the marked threshold of the ROC curve in Figure 4.2.2. The accuracy would be (1529 + 754)=(1532 + 804) = 97; 7%. It may be more interesting to decrease the false positive than to decrease the false negative due to several possible hits on each human.

Figure 4.3.1: The figure shows a screen capture of the proof of concept display. In the top image, found humans are marked with their contour in a color corresponding to their distance from the camera. The color coded depth map is shown in the bottom left image and an overview position display is shown in the bottom right image. The color bar in the middle corresponds to the distances shown in the position overview. Marked distances are 2.5 m, 5 m and 7.5 m respectively. The red region in the overview and teal regions in the top image are outside of the detection area.

Chapter 5

CONCLUSION & FUTURE WORK

This chapter contains the drawn conclusions and answers to the problem definition. It also contains suggestions for future research.

The quality of the features have a large impact on detection accuracy. The feature that showed the best

individual result was fourier descriptors. Of the used classifiers ANN had the best classification performance while using all features. ANN was also the best classifier when considering classification speed.There are a lot of possible improvements that could be implemented. Of course there could be even more efficient features waiting to be discovered, it could for instance be interesting to include the distance in some of the features that performs differently for different distances.

The fact that humans are unlikely to move very far between two frames can be used for motion trackingand concentrate the detection around interesting areas and probably prevent a lot of sporadic false positives.Another interesting improvement could be to group several positive hits, in close distance to each other, to one human. As for now, each human can give many hits because of the overlap during segmentation.

References

[1] Z.-R. Wang et al. \Pedestrian Detection Using Boosted HOG Features". In: Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on. 2008, pp. 1155 {1160. doi: 10.1109/ITSC.2008.4732553.

[2] N. Dalal and B. Triggs. \Histograms of Oriented Gradients for Human Detection". In: In CVPR. 2005, pp. 886{893.

[3] F. Suard et al. \Pedestrian Detection using Infrared images and Histograms of Oriented Gradients". In: Intelligent Vehicles Symposium, 2006 IEEE. 2006, pp. 206 {212. doi: 10.1109/IVS.2006.1689629.

[4] V. Vaidehi et al. \Multiclass object detection system in imaging sensor network using Haar-like features and Joint-Boosting algorithm". In: Recent Trends in Information Technology (ICRTIT), 2011 International Conference on. 2011, pp. 1011 {1015. doi: 10.1109/ICRTIT.2011.5972251.

[5] W. Yongzhi et al. \Pedestrian Detection Using Coarse-to-Fine Method with Haar-Like and Shapelet Features". In: Multimedia Technology (ICMT), 2010 International Conference on. 2010, pp. 1 {4. doi: 10.1109/ICMULT.2010.5630446.

[6] E.-J. Choi and D.-J. Park. \Human detection using image fusion of thermal and visible image with new joint bilateral _lter". In: Computer Sciences and Convergence Information Technology (ICCIT), 2010 5th International Conference on. 2010, pp. 882 {885. doi: 10.1109/ICCIT.2010.5711182.

[7] H. Sun, C. Wang, and B. Wang. \Night Vision Pedestrian Detection Using a Forward-Looking Infrared Camera". In: Multi-Platform/Multi-Sensor Remote Sensing and Mapping (M2RSM), 2011 International Workshop on. 2011, pp. 1 {4. doi: 10.1109/M2RSM.2011.5697384.

[8] H. Andreasson, R. Triebel, and A. J. Lilienthal. \Vision-based People Detection Utilizing Reective Vests for Autonomous Transportation Applications". In: IROS Workshop on Metrics and Methodologies for Autonomous Robot Teams in Logistics (MMART-LOG). 2011.

[9] L. E. Navarro-Serment, C. Mertz, and M. Hebert. \Pedestrian Detection and Tracking Using Threedimensional LADAR Data". In: The International Journal of Robotics Research 29.12 (Oct. 2010), pp. 1516{1528. doi: 10.1177/0278364910370216. url: http://dx.doi.org/10.1177/0278364910370216.

[10] L. Spinello and K. O. Arras. \Leveraging RGB-D Data: Adaptive Fusion and Domain Adaptation for Object Detection." In: Proc. of The International Conference in Robotics and Automation (ICRA). 2012.

[11] M. Luber, L. Spinello, and K. O. Arras. \People tracking in RGB-D data with on-line boosted target models". In: Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on. 2011, pp. 3844 {3849. doi: 10.1109/IROS.2011.6095075.

[12] A Howard et al. \Detecting Pedestrians with Stereo Vision : Safe Operation of Autonomous Ground Vehicles in Dynamic Environments". In: System (2007). url: http://www-robotics.jpl.nasa.gov/publications/Andrew_Howard/howard_isrr07.pdf.

[13] M. Bertozzi et al. \Stereo Vision-based approaches for Pedestrian Detection". In: PROCEEDINGS OF IEEE INT. CONF. ON COMPUTER VISION AND PATTERN RECOGNITION. 2005, pp. 23{28.

[14] L. Zhao and C. Thorpe. \Stereo and Neural Network-based Pedestrian Detection". In: IEEE Transactions on Intelligent Transportation Systems 1.3 (2000), pp. 148 {154.

[15] P. Doll_ar et al. \Pedestrian Detection: An Evaluation of the State of the Art". In: vol. 99. PrePrints. 2011. doi: http://doi.ieeecomputersociety.org/10.1109/TPAMI.2011.155.

[16] P. Doll_ar. Pedestrian Detection: The State of the art. Video seminar (http://research.microsoft.com/apps/video/?id=135046). [Online; accessed 5-June-2012]. 2010.

[17] S. Kamijo, K. Fujimura, and Y. Shibayama. \Pedestrian detection algorithm for on-board cameras of multi view angles". In: Intelligent Vehicles Symposium (IV), 2010 IEEE. 2010, pp. 973 {980. doi: 10.1109/IVS.2010.5548113.

[18] L. Zhang and Y. Liang. \Motion Human Detection Based on Background Subtraction". In: Education Technology and Computer Science (ETCS), 2010 Second International Workshop on. Vol. 1. 2010, pp. 284 {287. doi: 10.1109/ETCS.2010.440.

[19] R. C. Gonzales and R. E. Woods. Digital Image Processing. Pearson International, 2008, pp. 115, 205{255, 394{456, 839{840. isbn: 978-0-13-505267-9.

[20] Dimenco. http://www.dimenco.eu/2dz/. [Online; accessed 23-April-2012]. 2011.

[21] H. Levkowitz and G. T. Herman. \Color Scales for Image Data". In: IEEE Comput. Graph. Appl. 12.1 (Jan. 1992), pp. 72{80. issn: 0272-1716. doi: 10.1109/38.135886. url: http://dx.doi.org/10.1109/38.135886.

[22] PrimeSense. http://www.primesense.com/en/technology/115- the- primesense- 3d- sensing-

solution. [Online; accessed 16-April-2012]. 2011.

[23] A. Fusiello et al. A Compact Algorithm for Recti_cation of Stereo Pairs. 1999.

[24] M. Gosta and M. Grgic. \Accomplishments and challenges of computer stereo vision". In: ELMAR, 2010 PROCEEDINGS. 2010, pp. 57 {64.

[25] N. Kokash. An introduction to heuristic algorithms. 2005.

[26] S. B. Kotsiantis. Supervised Machine Learning: A Review of Classi_cation Techniques. Informatica 31:249{268. 2007.

[27] M. Wahde. Biologically Inspired Optimization Methods. WIT Press, 2008. isbn: 9781845641481.

[28] R. Laganiere. OpenCV 2 Computer Vision Application Programming Cookbook. Packt Publishing, 2011, pp. 182{185, isbn: 978-1-849513-24-1.

[29] K. G. Derpanis. Integral image-based representations. Tech. rep. Department of Computer Science and Engineering York University, 2007.

[30] M. Jie et al. \Fast Fourier Descriptor Method of the Shape Feature in Low Resolution Images". In: Wireless Communications Networking and Mobile Computing (WiCOM), 2010 6th International Conference on. 2010, pp. 1 {4. doi: 10.1109/WICOM.2010.5601317.

[31] Willows Garage. http://opencv.willowgarage.com/documentation/cpp/structural_analysis_ and_shape_descriptors.html. [Online; accessed 27-April-2012]. 2012.

[32] R. Lienhart. An Extended Set of Haar-like Features for Rapid Object Detection. Tech. rep. Intel Labs Intel Corporation Santa Clara USA, 2002.

[33] T. Kim et al. Pose Robust Human Detection in Depth Image Using Four Directional 2D Elliptical Filters. Tech. rep. Department of Computer Science, Engineering Pohang University of Science, and Technology, 2008.

[34] L. Rokach and O. Maimon. \Top-down induction of decision trees classi_ers - a survey". In: Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 35.4 (2005), pp. 476 {487. issn: 1094-6977. doi: 10.1109/TSMCC.2004.843247.

[35] S. Thirumuruganathan. http : / / saravananthirumuruganathan . wordpress . com / 2010 / 05 / 17 / a - detailed-introduction-to-k-nearest-neighbor-knn-algorithm/. [Online; accessed 26-April-2012]. 2010.

[36] C. J. Burges. \A Tutorial on Support Vector Machines for Pattern Recognition". In: Data Mining and

Knowledge Discovery 2 (1998), pp. 121{167.

[37] S. Haykin. Neural Network and Learning Machines. Pearson International, 2009, pp. 40{42,51{52. isbn: 978-0-13-129376-2.

[38] T. Fawcett. ROC Graphs: Notes and Practical Considerations for Data Mining Researchers. 2003.

[39] Microsoft. http://emea.microsoftstore.com/UK/en-GB/Microsoft/Design-+-Developer/Visual-Studio-2010. [Online; accessed 27-April-2012]. 2012.

[40] The MathWorks, Inc. http://www.mathworks.se/. [Online; accessed 27-April-2012]. 2012.

[41] Willows Garage. http://www.willowgarage.com/pages/software/opencv. [Online; accessed 27-April-2012]. 2012.

[42] DotNetNuke Corporation. http://75.98.78.94/default.aspx. [Online; accessed 27-April-2012]. 2012.