Viewer-Aware Dynamic Advertising

by

Seán Bruton, B.A.(mod)

Dissertation

Presented to the

University of Dublin, Trinity College

in fulfillment

of the requirements

for the Degree of

Master of Science in Computer Science

University of Dublin, Trinity College

September 2011

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

Seán Bruton

September 14, 2011

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

Seán Bruton

September 14, 2011

Acknowledgments

I would like to thank my supervisor, Kenneth Dawson-Howe for his advice, guidance and time taken in supervising this dissertation. I would also like to thank my classmates, friends and family for their continued support and encouragement.

Seán Bruton

University of Dublin, Trinity College
September 2011

Viewer-Aware Dynamic Advertising

Seán Bruton
University of Dublin, Trinity College, 2011

Supervisor: Kenneth Dawson-Howe

Display advertising represents a significant portion of the overall advertising market. However, unlike other forms of advertising, such as online, the targeting of specific outdoor advertisements to customers is limited to aspects such as geography and local knowledge.

As part of this dissertation, we explore methods of targeting advertising using various techniques from computer vision to determine statistics about the viewers of the advertisements. Such statistics may include gender, age, whether wearing accessories such as hats or glasses, etc. For the purposes of the dissertation, the statistic that will be focussed upon is that of gender.

In order to determine this statistic, the Adaboost algorithm was used to create a classifier for gender. This classifier is trained and tested on a sample dataset derived from the FERET image database using five-fold cross validation and found to achieve an accuracy of 90.8%, which is comparable to similarly tested methods in the literature.

This classifier is used as part of an encompassing system that includes a display and a camera. The system tracks an advertisement viewer's face and a probabilistic model is used to ascertain the gender of the viewer over a number of image frames from the camera. As part of the system, an advertisement is displayed that can be changed dynamically depending on the results of the classification of the display viewers, thus targeting the advertising more appropriately.

Furthermore, the system can easily be extended to measure other similar statistics such as age. It can also be used to determine statistics such as numbers of viewers of an advertisement over a time period.

Contents

List of Figures

List of Algorithms

Chapter 1

Introduction

1.1 Motivation

The global outdoor advertising market represents a projected 24 billion US Dollars for the year 2010 [1]. The targeting of such outdoor advertising however is limited to knowledge in relation to the location of the advertisement. Online advertising utilises information about the viewer in order to target advertisements more efficiently. There are currently no ways of being able to do this effectively with outdoor advertising. Should a system be developed that could generate statistics regarding the viewers of an outdoor advertisement, the advertisement displayed could be changed to best match the criteria of the viewer. This would increase the effectiveness of an outdoor advertisement in reaching the target audience.

Similarly, current display advertising technology does not allow for the accumulation of information in relation to the number of viewers that an advertisement may have. Such information would be valuable to the entities placing the advertisements as it would allow for the measurement of how popular certain locations are, as well as the appeal of certain advertisements that are displayed. If this information where to be combined with the statistics regarding the viewers of the advertisement, the entities placing the advertisement could allocate their resources much more effectively and increase the appeal of advertisements through identifying the characteristics of an advertisement that makes it appealing.

1.2 Project Goals

The project goals can be broken down as follows

1. Investigate methods to allow for display advertising to be more targeted to a viewer of a display advertisement. The methods themselves may only satisfy certain aspects of the task and so it may be necessary to break down the investigation of these methods into relevant areas. These methods should enable statistics regarding a viewer of an advertisement to be generated.

2. Utilise state-of-the-art statistical and computer vision techniques to classify attributes, such as gender, of an advertisement viewer. The classification schemes should achieve a high degree of accuracy to fulfil any useful task. The classification scheme should be rigorously tested to determine its accuracy

3. Develop a system framework that would perform such classifications and dynamically change an advertisement being displayed based upon the results. The framework should be able to work for a general classifier. It may also be necessary that the system perform additional tasks, or indeed tasks to enable it to perform its primary function.

1.3 Dissertation Structure

The remainder of this dissertation is structured as follows:

Background

In this chapter we wish to identify the key research that has been performed that is relevant to the task of providing a description of a viewer of an advertisement. An essential task that needs to be performed to enable such a description to be formulated is that of people detection. We will look at the key papers that deal with this problem, including any recent papers that address a similar issue. In formulating a description of a person, probably the most pertinent aspect is the gender of the person. We will examine the key research to date that has been performed to solve the problem of identifying the gender of a subject in an image. Another aspect that would be desirable in the description of a person is that of the person's age. We shall examine the literature that deals with solving this problem for an image of a person. Finally, there exists numerous datasets of images that are regularly used in methods outlined in the literature. We will outline the key features of these databases with a view to using one or more in a proposed methodology.

Training a Classifier

Here, we will discuss the relative merits and downsides of identified classification schemes for identifying a person's gender based on an facial image. Based on this analysis we will select one to form the basis of a proposed methodology. We will identify the technologies that will be used in the construction of the classifier. We will also discuss certain aspects regarding the training data that will be used in the training of the classifier. The training of the classifier is performed in a number of stages. The early stages relate to ensuring that the data that will be used in the training of the classifier is standardised. We outline how we perform this standardisation via a rigid transformation and histogram normalisation. We describe the classification scheme itself in significant detail. Finally, we detail how the actual classifier is trained.

Evaluation of the Classifier

In this chapter, we are looking to determine how good a classifier has been trained. Before we proceed to test the classifier, we introduce some testing techniques that will ensure that the testing of our classifier is done to a rigorous statistical standard. We introduce the concepts by which we will be measuring the performance of our classifier with. We provide the results of the testing of the classifier. We compare the results obtained with the results that are stated in the literature. We provide commentary for the potential reasons for any significant differences in these results.

Application Design

One of the goals of this project is to provide a system that can perform classifications and dynamically change an advertisement based on the results. Here, we detail the specific requirements of such a system. We provide a conceptual set-up of the system and identify the components that will make such a system. We also detail any other features that we may desire in the system and provide potential system components to provide these features.

Application Implementation

This chapter details how each of the components of the system was constructed. In implementing a system, we utilise certain concepts, theories, algorithms and technologies. These include Mutual Information, Mahalanobis distance, SURF features and background detection, with descriptions of how they work and the reasoning behind their use as part of the system.

Application Testing

Here, we review the performance of the system in light of the requirements that were set out for it. We assess whether it succeeded in satisfying these requirements and provide commentary on its performance. We will detail some of the issues that were encountered in the testing of the system. We will provide a similar analysis of the system with respect to the additional desired features of the system.

Conclusion and Further Work

Finally, we assess the results of the research in light of the goals of the project. We detail the aspects that worked well and those that may not have performed as well. We identify areas that we would concentrate our further efforts on if we were to improve the system.

Chapter 2

Background

We wish to identify the key research that has been performed that is relevant to the task of providing a description of a viewer of an advertisement. An essential task that needs to be performed to enable such a description to be formulated is that of people detection. We will look at the key papers that deal with this problem, including any recent papers that address a similar issue. In formulating a description of a person, probably the most pertinent aspect is the gender of the person. We will examine the key research to date that has been performed to solve the problem of identifying the gender of a subject in an image. Another aspect that would be desirable in the description of a person is that of the person's age. We shall examine the literature that deals with solving this problem for an image of a person. Finally, there exists numerous datasets of images that are regularly used in methods outlined in the literature. We will outline the key features of these databases with a view to using one or more in a proposed methodology.

2.1 State of the Art

2.1.1 Detection of People

In order to be able to describe an individual in an image, one must first be able to detect that a person is present in the image. Various techniques have been employed to solve this problem.

In Pfinder[2], a system was developed that allows for the tracking of people and the interpretation of their gestural behaviour. This system utilises the differences in image values that arise from the movement of people across a scene to be able to determine the presence of a person. These differences are measured in YUV space and updated for each frame. A person, once detected, is described under a colour model using a small number of clusters which are determined from the pixels identified to be part of the person. A limitation of the system for the purposes of people detection is the rigidity of certain assumptions. Incorrect classifications may result from such things as a large change in lighting in the scene or a change in the scene background.

In Hydra[3], a real-time system was developed that could detect and track multiple people when they appear in a group. The system used background detection to determine coarse representations of groups of people. The system then utilised a shape model, a motion model and correlation-based matching methods to classify whether or not a foreground contained multiple people, and to segment the region into its constituent people and track them.

In [4], Haar wavelet transforms were used as object classes for people. A support vector machine was trained with positive and negative samples for the object class, and subsequently used to classify test images for the presence of that object.

Many techniques utilise the method outlined in [5] for detection of objects (including people) in real-time. The method involves calculating what is known as an integral image for test images and identifying weak classifiers out of these test images using the AdaBoost algorithm [6]. Successive weak classifiers are combined by linear combinations to form strong classifiers. The method was demonstrated with detection of human faces, and showed a considerable speed improvement over other published methods at the time.

In [7], a method for segmenting the foreground and background is proposed to allow for the detection of people in a scene. In order to initially determine the background, a variation of median filtering is implemented. To update the background, two approaches are taken: a temporal filtering process is used to remove small or gradual changes (such as lighting) and a re-initialisation of the background model every 50-100 frames. A subtraction of this background image gives the resulting foreground objects. In order to determine whether these objects are humans, the authors implement a method which utilises skeletonisation of the detected region and determination of the centre of gravity of the region.

A similar method of performing people detection, i.e. by determination of foreground objects, was developed by Zhou and Hoang[8]. Their system uses a running average to determine the background. The system also updates a foreground mask image in order to reduce the effect of small changes in the image. The system also implements shadow detection and removal in order to improve the accuracy of detection. To identify people in a scene feature vectors of foreground objects are compared against predetermined thresholds. An algorithm for reducing false positives is also introduced.

A method for the detection of people which combines multiple methods was introduced by Carbajales et al. [9]. This method combines various other methods which implement foreground segmentation. The determination of whether the foreground object is human is assessed by analysing the characteristics of the object (and its contours) such as the height to width ratio, the minimum bounded ellipse and the convexities of the object.

2.1.2 Gender Classification

Once a person has been detected with a scene, a description of the person would desirably include the gender of the person. Various methods have been employed to determine the gender of people within a scene. Most methods assume the case that a frontal facial image of the person is available and hence base the classification of the gender solely an image of the face.

In [10], such images of frontal faces are analysed in terms of a set of geometrical features such as pupil to eyebrow separation, nose width etc. These descriptions, along with the gender data of the subjects in the images, are used to train two competing HyperBF networks, in order to perform the gender classification. The authors state that they were able to achieve 79% accuracy on images of faces that were not used in the training stage.

In [11], the authors set out to be able to classify gender ethnicity and expression based on frontal facial images. The algorithm that the authors propose consists of two steps. The first step is to perform registration of a grid with the face and secondly to perform the classification based upon feature values extracted at certain points. The grid registration can be performed manually or with other automated methods listed in the paper. In order to perform the classification, images are transformed using a set of Gabor filters at various scales and orientations, for two different grid types. The amplitude of the complex valued Gabor transform coefficients are sampled on the grid and combined into a single vector. In order to reduce the dimensions of the resulting vector, a PCA step is performed. The resulting smaller vectors subsequently undergo a Linear Discriminant Analysis (LDA) in order to identify clusters of similar attributes. For classification, a standard metric was used as a distance measure to identify the closest cluster. Stated classification performances of 92% for gender, 91% for expression, and 95% for race were achieved using this method.

In [12], the authors attempted to solve the problem of gender classification based on a low resolution image of a face. Their proposed solution utilises Support Vector Machines, a learning algorithm for pattern classification, to generate a hyperplane classification boundary based upon the sample data set used for human faces[13], along with their associated gender. A Gaussian radial basis function was used as the kernel function in the SVM. Using the method, an overall error rate of 3.4% was achieved on the sample data set. It should be noted however, that the data used in the sample data was also included in the test data. Another interesting result of their analysis was that the difference in classification performance with low resolution (21×12 pixels) versus higher resolution (84×48 pixels) was 1%.

An application of the AdaBoost algorithm to perform the task of face classification was proposed in [14]. The authors utilised the methodology outlined in [5] for the Viola-Jones rapid detection of objects using AdaBoost, to perform classification of face images under gender and ethnicity. It was found that an error rate of 21% was achieved for gender classification using their method compared to a result of 24.5% using the method outlined in [12], under 5-fold cross-validation. The respective results for ethnicity classification were 20.8% and 22.6%. A method for performing the tracking of faces in a real-time application was also outlined. This method involved a variant of deterministic particle filtering which uses the depth reached in the Viola-Jones stage by searching in the regions that have at least 35% overlap with previous face locations as the probabilities of a face region.

A method for classifying gender involving lookup table ("LUT") weak classifiers combined using the AdaBoost algorithm was proposed in [15]. Their approach was similar to that of Shakhnarovich et al. [14], but rather than using threshold weak classifiers they used LUT weak classifiers based on the reasoning that LUT weak classifiers are better suited for AdaBoost training due to the nature of the distribution of samples often being multi-Gaussian. The highest gender classification accuracy results were 88%, achieved using 36×36 resolution images and a LUT size of 16.

Other facial features were detected in still images and video streams by [16]. The features included gender, race, presence of glasses and a moustache. The method employed a combination of PCA and SVMs in order to build classifiers for each of the features. The paper also introduces a tracking method for video streams whereby face detection [5], skin colour detection, and tracking were used to identify the feature descriptions that were common across the video and hence to reduce any extra calculations of the features across the video.

An experimental analysis of proposed gender classification methods was performed in [17]. The system proposed that a number of initial stages be performed prior to the classification of gender. These stages included the detection of faces that would handle faces that are not entirely frontal, as well as a mapping of 88 landmark points of a face using a face alignment module to a template face image. In the testing of different classifiers, the authors chose to test those of Moghaddam and Yang [12] and Shakhnarovich et al. [14], as well as a method utilising Fischer Linear Discriminant ("FLD"). Testing on the FERET face image database [13], it was found that under 5-fold cross-validation testing the SVM, FLD and Real AdaBoost methods achieved accuracies of 92.2%, 85.7% and 93.8%, respectively.

Another method implemented was that of [18]. In order to perform the task of gender classification, an adaption of the AdaBoost algorithm was used to create a classifier using an image database [13]. The training data was transformed to match a standard image size and normalised for image brightness. The algorithm worked well with 94.4% accuracy achieved using 100 weak classifiers in the boosting algorithm. When compared to Support Vector Machine (SVM) methods, the classifier outperforms the method and is less computationally expensive (c. 50 times).

In [19], the issue of classifying gender based on facial images is tackled using a combination of Independent Component Analysis (ICA) and Support Vector Machines (SVMs). Prior to performing any analysis, the training set undergoes a pre-processing stage which includes image normalisation. In order to represent the training images, Independent Components are used to represent each image as a feature vector in a lower dimensional subspace. Support Vector Machines are used to define a hyperplane in ICA space in order to perform the classification. Using 200 training images, the authors were able to achieve 96% accuracy for gender recognition.

In [20], the authors sought to be able to classify gender, age and ethnicity from facial images. Under their proposed method, firstly some pre-processing is performed, namely rotating and scaling of the sample images. Gabor filters are then applied to the facial images. The AdaBoost algorithm is used to identify the key Gabor features. SVMs are trained using a bespoke method (Sequential Minimal Optimisation) based on the results of the previous steps. The resulting hyperplanes are used to classify the test images. There were three possible classifications for ethnicity. Various frequencies for the Gabor transformation were tested. A result of 91.6% for gender recognition was achieved. This was increased by performing a pre-processing step to 92.8%. Ethnicity recognition accuracy was stated as 90.6% (96.2% with pre-processing).

More recently, Fu et al. [21] sought a new method to perform gender classification. A feature extraction method, centralised Gabor Gradient Histogram (CGGH) was proposed for facial gender recognition. The advantage of this method is that it captures discriminative information at different scales and orientations. A Centre-Based Nearest Neighbour (CNN) measure was used to perform classification. As part of the method the encoded images are subdivided into sub-images. The method achieved an accuracy of 96.6% on the FERET database.

Another method [22] combined dimension reducing methods such as Principle Components Analysis ("PCA") and Local Linear Embedding ("LLE") with Support Vector Machines ("SVMs") to perform gender classification. PCA and LLE were utilised to reduce a face's data to 128 and 99 eigenvectors respectively. SVMs were used to determine the state space classification. It was found that SVMs performed well with 5% classification error, when used with PCA. It was found that LLE did not perform as well as PCA methods.

In [23], Scale Invariant Feature Transforms (SIFT) and Gabor transforms at dense grid pixels of a face image are used to represent a face. The SIFT descriptors are modified to overcome certain problems that exist in this application. The Gabor representations are fused with the dense SIFT at the feature-level to improve the accuracy. AdaBoost is used to form a strong classifier based on these features. It was found that dense SIFT and Gabor descriptions of facial images provide the information necessary to train a highly accurate gender classifier.

2.1.3 Age Classification

As part of a description of a person, one would also like to be able to determine an estimated age of the person. Several methods have been proposed to perform this task on facial images.

An early method proposed was that of [24]. In order to solve the problem of age estimation, this paper proposed utilising facial features, namely wrinkles, to estimate the age of a subject. The wrinkle maps of the subject are extracted using image processing methods. The estimation of age is performed using a Look-Up Table based on a training dataset.

In [20], the method outlined in Section 2.1.2 was also used to perform age estimation The age estimation was broken into only two sets however, young and old. Age recognition accuracy was stated as 91.8% (94.1% with pre-processing).

In [25], an automatic age estimation method known as AGES was introduced. The authors propose an age estimation method through the modelling of the facial ageing pattern. The method works by constructing a representative subspace for the individuals facial images, ordered in time. When an unseen face is tested, a projection in the subspace that can reconstruct the face with minimum reconstruction error is used to determine the age. The authors state a result of a Mean Average Error of 6.77 years was achieved when tested with the FG-NET database.

Another method that was proposed was that of Guo et al.[26]. This method was based on a statistical framework that uses Bayesian statistics to propose a fusion of Support Vector Machines and Support Vector Regressions in order to provide an estimation of a subject's age. The authors state that they achieve a Mean Average Error of 4.97 years using their proposed method on the FG-NET database. This is an improvement over other tested methods such as the Locally Adjusted Robust Regressor, which achieved 5.07 years.

A more recent paper [27] addressed the issue of age estimation when gender is known compared to when the gender is not known. It was found that pre-classifying gender significantly improves the accuracy of the age estimation for a variety of age classification methods. The best result achieved was a combination of biologically inspired features and Marginal Fischer Analysis, yielding a mean absolute error in years of 3.2, which was decreased to 2.63 when the gender was given. It was found however, that overall, when the gender is unknown and is identified prior to the age estimation step, a combination of biologically inspired features and Locality Sensitive Discriminant Analysis was the best performing of the tested methods with a mean absolute error of 2.95 years. The accuracy of this method was increased when the classifications are first separated into three separate age groups (young, adult, senior). Finally the authors propose a data fusion approach to further increase the accuracy by combining several of the tested methods (the above mentioned Marginal Fischer Analysis and Locality Sensitive Discriminant Analysis as well as Orthogonal Locality Preserving Projections) and using the method which performed best for the particular gender and age group classification (e.g. adult male). A mean average error of 2.66 years was achieved using this method.

Another recent paper [28], attempted to solve the problem of assessing the age of a subject across different lighting conditions. The classification scheme separated the ages into four coarse groups (baby, child, adult and old) rather than building a unified age regression model. Gabor features were extracted from test images taken from the internet, and used in a LDA classifier. A fuzzy version of LDA is introduced through the definition of age membership functions. Comparative experiments performed using SVM, AdaBoost and LDA showed that the LDA method worked better for age classification with a precision of 92.5%.

2.1.4 Testing Data

Various databases containing facial images have been collated by research groups for the purposes of computer vision algorithm testing and development. Many of these databases are available under licence to be used by other researchers.

One of the most widely used databases is the Facial Recognition Technology (FERET) Database [13]. The database consists of 14,051 images of human heads with views ranging from frontal to left and right profiles. The database consists of 994 unique people (591 male, 403 female). The database has been used as a training dataset for the purposes of gender classification [18, 12, 20, 21, 23].

Another database that is regularly used is that of FG-NET ageing database [29]. The FG-NET Aging Database contains 1,002 face images from 82 subjects, with a minimum age of 0 and a maximum age of 69. It is often used in the training and testing of algorithms to determine the age of a subject from a facial image [25] [26].

2.2 Remarks

The literature has shown that although the classification of human subjects based on their gender and age is achievable using current techniques, the methods are suboptimal and there remains significant work to be done in improving such classification schemes to achieve higher accuracies.

Chapter 3

Training a Classifier

3.1 Approach

3.1.1 Methodology

In order for us to perform the task of the classification of a viewer of an advertisement, we need to decide on an approach to performing this classification. For the purposes of this dissertation, it was decided that we would focus on the task of classifying a viewer's gender. This was decided based on a number of factors. These include the constraint of the limited time available to perform the research and the fact that a classification scheme that can be shown to work in a real-time environment with a camera input can present a potential framework for other classification schemes (such as age, ethnicity, etc.) to be subsequently introduced to.

Of the methods listed in Section 2.1.2 that attempt to solve the problem of classifying gender, it was necessary to select one that would potentially work best in a more general environment, such as that of a mounted camera receiving a video stream of a scene with people in it.

After careful consideration, it was decided that the method that best suited this was that of Baluja and Rowley [18]. We will now outline the reasons why this approach was selected over other methods listed in Section 2.1.2.

In the paper, the authors trained a classifier using the Discrete variant of the AdaBoost algorithm as implemented by Viola and Jones [5], as well as a classifier utilising Support Vector Machines [30]. As weak classifiers for the boosting schedule, the authors used a number of pixel comparisons. The approach to the training of the classifier that utilises Support Vector Machines was in line with that of Moghaddam and Yang [12].

In the analysis of the two classification schemes, the authors state that the classifier trained using Discrete AdaBoost performed better than that of the classifier trained using the Support Vector Machines approach. It was found in a 5-fold cross-validation test that the best SVM result was found to be 93.5% and the best AdaBoost approach was 94.4%. In the original paper illustrating the SVM approach [12], the authors state that they achieved an accuracy of 96.4%. This accuracy was based on the classification of images that were used in the training of the classifier and hence the result obtained by Baluja and Rowley [18] should be given greater consideration due to the data independence of the training and testing samples.

Furthermore, it was found by Yang et al. [17], that the approach utilising Real AdaBoost [14] outperformed that utilising SVM [12].

The authors also tested different variations of a pre-processing stage in the training of the classifier. One variation involved using masked regions so that only pixels corresponding to face regions would be used. It was found that the approach that yielded the best results was when no image masks were used. Another variation that was tested was whether it would be preferable to normalise the intensities of the sample images prior to their use in the training of the classifier. It was found that the classification results were more accurate when this normalisation stage was performed. A test was also performed using two different image sizes for the samples. It was found that better classification results were achieved when a 20×20 image size was used rather than a 12×21 image size.

Another analysis that is performed in this paper is that of the performance of the classifiers in terms of the time required to classify an image as male or female. It was found that for a comparable accuracy across the AdaBoost and SVM approaches, AdaBoost classifiers are approximately 1.6% as expensive for 20×20 images. For the purposes of a real-time application, in which frame rate is a large consideration, this information may be considered important. It should be noted however, that the average time for classification of a single face for the SVM classifier was 581μs (compared to 9.47μs for the AdaBoost classifier), which may not represent a significant overhead for an application that is required to run at close to real-time rates (50000μs per frame for a 20 frames per second application).

As part of the analysis performed in [18], the authors also found that the AdaBoost classifier was less sensitive than the SVM classifier for each of in-plane rotations, scalings and translations of the images. This aspect would be considered highly relevant in the designing of a real-time application to perform gender classification. It may often be the case for a real-time application that a perfectly aligned face will not be available and the robustness of a classifier to any such variations is a highly desirable characteristic.

There have been other classification schemes that purport to achieve higher accuracies [23] and be invariant to scale and rotation [21]. It should be noted in the case of Fu et al. [21], that the proposed classification scheme is reliant on an accurate description of the face being available (to allow for the breaking up of a face image into 15 sub-regions). For the reason that these operations have only been performed manually in the cases of the research and that there remains significant research to be performed to identify methods of performing such methods automatically, this method is rejected for the consideration of a training method for a classifier in this case.

Another recent proposed method which achieved high accuracy in gender classification, is that of Wang et al. [23]. The training of the classifier is performed using the AdaBoost algorithm. The proposed weak classifiers to be used in the training schedule are dense SIFT and Gabor descriptions of face images. Their method achieves highly accurate results on roughly aligned faces. However, there remain issues in relation to the amount of processing time that is required to generate the SIFT descriptors and perform the Gabor filtering prior to being able to classify a face. The authors state that they achieved performance of 6 frames per second for a prototype real environment set-up involving a web-cam. For the purposes of our research, it may be deemed that this performance may be below a desired performance threshold which one would require.

It may also be interesting to be able to provide a baseline standard for a real environment setup, using Haar-like features as weak classifiers in an AdaBoost training schedule, prior to implementing more complex solutions to allow for comparisons to be made in the performances of such methods.

3.1.2 Technology

It was decided that the technology that would be used in the training of a classifier would be C++ and the OpenCV library [31].

OpenCV is an open-source computer vision library initially developed by Intel and now supported by WillowGarage. The library contains many useful classes for performing common image processing and computer vision tasks. Such techniques include image registration and image histogramming. The library provides many utility functions to perform such common tasks as loading and saving images, and can handle many popular image formats. The library also includes implementations of many computer vision algorithms such as optical flow and rapid object detection using a boosted cascade [5]. The library is coded in C/C++ and bindings exist for Python and Java. The codebase itself is mature and highly optimised, including implementations that can make use of multi-core processors and the Graphics Processing Unit ("GPU") of a user's computer.

This library is particularly attractive for our purposes as it contains an implementation of the AdaBoost training algorithm. The code provided can be compiled to perform different variants of the AdaBoost training algorithm, including Gentle, Discrete, and Real AdaBoost. The standard distribution itself includes some pre-trained classifiers that have been trained using the AdaBoost algorithm via the method outlined by Viola and Jones [5] for objects such as frontal human faces, eyes and mouths. These pre-trained classifiers may provide certain functionality that we may desire in our system, such as the detection of faces.

Some useful references regarding the library include [32] and [33]. The former deals with the details of the functions available under the library and provides some explanations of how they work. It deals with each in a tutorial manner, with exercises for the reader to allow them to better grasp the underlying mechanics of the library. The latter reference deals more with programming architectures using OpenCV such as model-view-controller, and illustrates the use of these architectures by solving common computer vision tasks using the OpenCV library. It is particularly useful however as it deals explicitly with the new C++ interface introduced to OpenCV from version 2.0 onwards, replacing the old C style interface.

3.2 Data Acquisition

For the purposes of the project, the FERET image database [13] was selected to be used as the sample training and testing set database. This database is widely used in the research area of classification of human faces [18, 12, 20, 21, 23]. The database consists of 14,051 images of human heads at different viewing angles (frontal, left and right profiles), comprising images of 994 unique people (591 male, 403 female).

For the purposes of our research, it was decided that we would concentrate on the classification of frontal human faces. This was based on the reasoning that we wished to be able to classify a viewer of an advertisement, and hence it can be assumed that the face is frontal. The face image filenames labelled fa, fb, ba and bj in the database, corresponding to frontal faces, were extracted for use. In total, the size of the sample set extracted from the database was 990 female face images and 1402 male face images. The images used were the grayscale versions as colour should not be needed for our purposes.

An application was built using C++ that would take the names of the images to be used and pre-append the relevant location on disk. The output of this application was a text file listing the file locations of all of the sample images to be used in the training and testing of the classifier.

3.3 Data Standardisation

To allow for the training of the classifier, it is necessary that the sample data is standardised. This process involves numerous steps. These steps were to find the locations of the left and right eyes and the mouth in each of the sample face images, to compute a rigid transformation to a 20×20 image template, to perform an inverse mapping to a 20×20 image and to normalise the intensities of the resulting images.

In order to perform these tasks, an application was built using C++ and the OpenCV library [31]. The application was able to process each face image in the order as read in from a text file as described in section 3.2. Each image was processed by the application and saved as a 20×20 image. The file locations of these saved images were saved to a text file so that they could be later used in a training schedule. The different processing stages of this application shall now be detailed.

3.3.1 Feature Location

As part of the application is was necessary that we identify the locations of the eyes and mouth of a face in an image. In order to perform this task, we used the pre-trained classifiers from the OpenCV distribution [31]. Firstly, a face detection step was performed with the classifier for frontal faces. This method returns the locations of a rectangle in an image indicating where a face has been detected. If a face was not detected (a false negative as each image contains a face), a face was assumed to be present and the rectangle indicating the location of the face was set to be the full dimensions of the image.

Secondly, within that image rectangle (the region of interest), classifications for left eyes, right eyes and mouths were performed using pre-trained classifiers. The parameters for each of these classifiers were set so that it would return the largest positive classification response for an object in the image (i.e. a single response). The results of these classifications were, as before, rectangles indicating the locations. In order to ensure that the left eye was not classified as the right eye or vice versa, those classifiers were only run on their respective halves of the region. The locations of each of the features were then taken to be the centre of each of the respective rectangles (to sub-pixel accuracy). It should be noted that if a classifier gave an unsatisfactory result, it was possible for the user of the application to manually correct the result. The three feature locations were now available to allow a rigid transformation to be performed.

3.3.2 Rigid Transformation

For the purposes of the data standardisation, it was necessary to find a rigid transformation between the face in an original image to the 20×20 template image size, with standardised eye locations. It is necessary to first give an explanation of the type of rigid transformation that we use. We will do this by comparing three different transformations. A detailed reference that will more fully explain the forthcoming concepts is [34].

Consider $\mathbf{x}$ to be a point in the image with the coordinates (x, y). We define the origin of the image to be the uppermost left-hand pixel. Thus, x corresponds to the number of rows in from the left-hand side of the image and y corresponds to the number of columns down from the top of the image.

We shall use Θ to denote the transformation parameter in the following transformation and $F(\mathbf{x}, \Theta)$ to represent the transformed pixel position, which we also denote by $\mathbf{x}'$.

Translation

The simplest 2D (planar) transformation we can represent is that of a *translation*. Using our notation, a translation can be represented by

$$F(\mathbf{x}, \Theta) = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} d_x \\ d_y \end{pmatrix} \tag{3.1}$$

Here, $\Theta = (d_x, d_y)$. Thus, there are two parameters to estimate. The transformation has 2 degrees of freedom. It preserves orientation, lengths, angles, parallelisms and straight lines of the original image. This transformation has the ability to match a rectangle on to a rectangle of equal size, length and width.

Affine

Another transformation we can represent is the *affine* transformation. Once again, using our notation, an affine transformation can be represented as

$$F(\mathbf{x}, \Theta) = \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} \mathbf{x}^T + \begin{pmatrix} d_x \\ d_y \end{pmatrix} \tag{3.2}$$

Here, $\Theta = (a_1, a_2, a_3, a_4, d_x, d_y)$. Thus, there are six parameters to estimate, corresponding to six degrees of freedom. This transformation has the properties of preserving straight lines and parallelisms of the original image. It has the ability to map a parallelogram on to a square.

Projective Linear

The final type of transformation that is examined is the *projective linear* transformation. It is also known as a perspective transform or homography. Using our notation, a projective linear transformation can be represented as

$$F(\mathbf{x}, \Theta) = \begin{pmatrix} \frac{a_1x+a_2y+d_x}{1+c_1x+c_2y} \\ \frac{a_3x+a_4y+d_y}{1+c_1x+c_2y} \end{pmatrix} \tag{3.3}$$

Here, $\Theta = (a_1, a_2, a_3, a_4, c_1, c_2, d_x, d_y)$. Thus, there are eight parameters to estimate, corresponding to eight degrees of freedom. This transformation has the properties of preserving parallelisms of the original image. It has the ability to map a general quadrangle to a square.

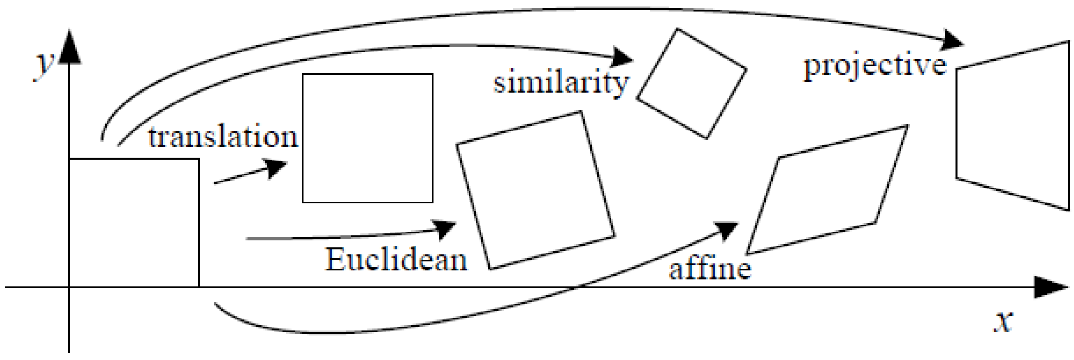


Figure 3.1: Basic set of 2D planar transformations.

Figure 3.1, taken from [34], gives a visual summary of the properties of the three transformations analysed.

In our case, we are limited to using an affine transformation by the number of correspondences between the two images, which is three (two eyes and mouth). It would be necessary to have four correspondences to be able to perform a projective linear transformation. It may be reasoned that there would not be significant warping of a frontal face to warrant a projective linear transformation and that parallelisms of a face that may be slightly off frontal or rotated about the axis coming out of the image will be preserved and hence an affine transformation should suffice.

From the feature points found in the original image the correspondences are illustrated in Figure 3.2.

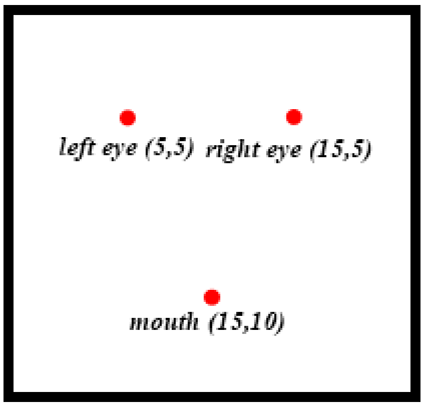


Figure 3.2: The correspondence points in the template image.

In reality, the transformation is performed in reverse. We transform the pixel from the registered image to a location in the original image, using the following equation.

$$F(\mathbf{x}', \Theta^{-1}) = \mathbf{x} = \begin{pmatrix} a'_1 & a'_2 \\ a'_3 & a'_4 \end{pmatrix} \mathbf{x}'^T + \begin{pmatrix} d'_x \\ d'_y \end{pmatrix} \tag{3.4}$$

Here, $\Theta^{-1} = (a'_1, a'_2, a'_3, a'_4, d'_x, d'_y)$. These parameters are solved in a system of linear equations with the values for the correspondences substituted in. OpenCV provides functionality to allow this to be performed efficiently.

The value that each pixel in the template image is transformed to will be a real valued location $\mathbf{x} = (x, y)$. In order for us to attain a pixel value at this location, it is necessary to perform some interpolation of the neighbouring pixels of this location. In line with the methodology used in [18], bilinear interpolation is used.

3.3.3 Image Normalisation

Now that we have a 20×20 image representation of a face, it is necessary to preclude any differences that may exist across the images due to different lighting conditions. This is done by normalising the intensities of the images.

Firstly we generate a histogram, $h(z)$, where $z = 0, \ldots, N$ (in our case $N = 255$), of the intensities of the 400 pixels in the image. We subsequently scale this histogram using the equation

$$h_n(z) = \frac{h(z)}{\sum\limits_{z=0}^{N} h(z)} \tag{3.5}$$

so that $\sum\limits_{z=0}^{N} h_n(z) = 1$. We loop through this histogram, summing the previous value in the histogram, (for $z = 1, \ldots, N$, $h_n(z) = h_n(z) + h_n(z-1)$), to obtain a cumulative histogram.

Secondly, we generate a histogram, $g(z)$ for $z = 0, \ldots, N$, based on a Gaussian normal distribution, using the standard equation.

$$g(z) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(z-\mu)^2}{2\sigma^2}} \tag{3.6}$$

Here, the value for μ, the mean is taken to be 127.5 and we use a value for σ, the standard deviation, of 64, in line with [18]. As before we scale the histogram, to obtain a histogram $g_n(z)$ with the property that $\sum\limits_{z=0}^{N} g_n(z) = 1$, and loop through this histogram, summing the previous value in the histogram, (for $z = 1, \ldots, N$, $g(z) = g(z) + g(z-1)$), to obtain a cumulative normal distribution.

Finally, for each pixel in the image, $\mathbf{x}$, we replace the previous intensity $f(\mathbf{x})$ with a new value $f'(\mathbf{x})$, based on the equation

$$f'(\mathbf{x}) = \arg\min_{z} \, (h_n(f(\mathbf{x})) - g_n(z)) \tag{3.7}$$

The image will now have normalised intensities. Some pre-processed male and female faces can be seen in Figures 3.3 and 3.4, respectively.

3.3.4 Remarks

We have now obtained a standardised 20×20 image for each of the face images that we are using from the FERET database. A summary of the processes involved can be seen in Figure 3.5.

The male and female faces have been separated manually and the file locations for each have been added to separate text files. We can now proceed to the training of the classifier.

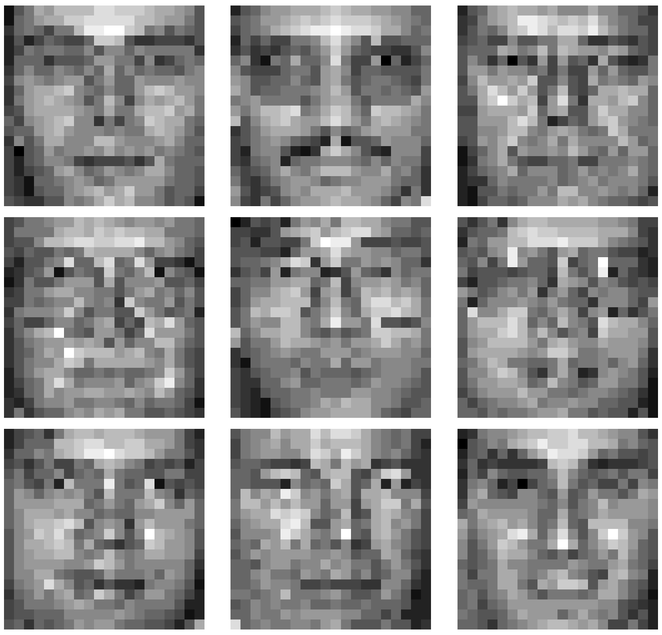


Figure 3.3: Some 20×20 pre-processed male faces.

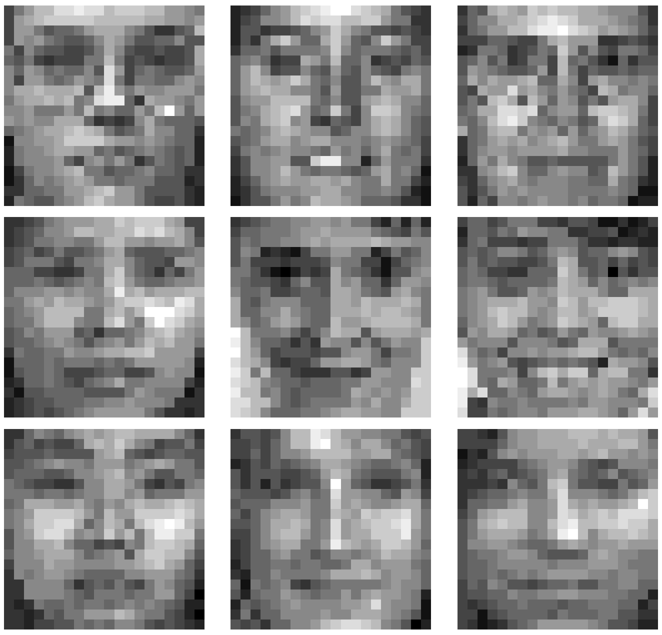


Figure 3.4: Some 20×20 pre-processed female faces.

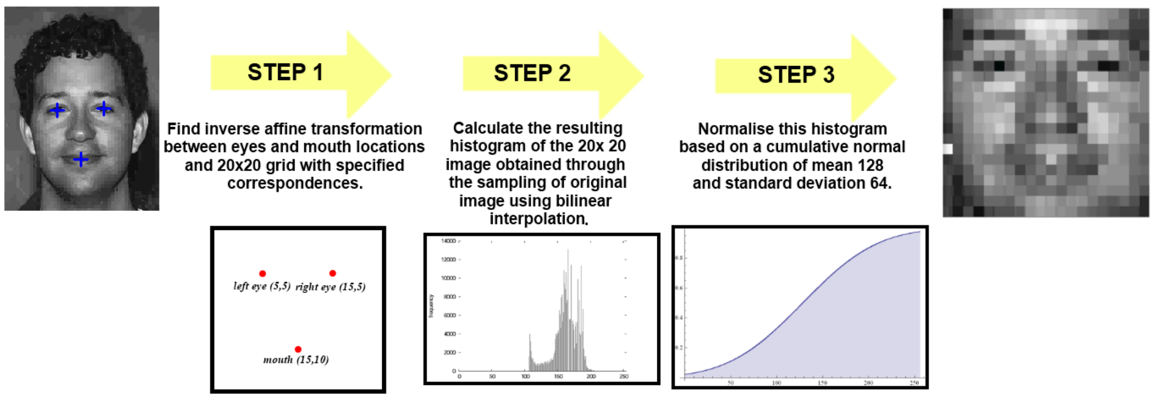

STEP 1
Find inverse affine transformation between eyes and mouth locations and 20x20 grid with specified correspondences.
STEP 2
Calculate the resulting histogram of the 20x 20 image obtained through the sampling of original image using bilinear interpolation.
STEP 3
Normalise this histogram based on a cumulative normal distribution of mean 128 and standard deviation 64.
left eye (5,5)
right eye (15,5)
mouth (15,10)

Figure 3.5: Pre-processing steps for each of the face images.

3.4 Classifier Training

3.4.1 AdaBoost

To train our classifier, the boosting algorithm as outlined by Viola and Jones [5] will be used. The AdaBoost algorithm (short for "Adaptive Boosting") was first introduced by Freund and Schapire in 1995 [35]. It is an iterative algorithm, which works by selecting a weak or base classifier repeatedly at each stage and giving it an associated weight. The algorithm is adaptive in that at each step the weights of incorrectly classified samples are increased. This results in the new classifier being forced to focus on those "hard" samples. In this way, it is a greedy learner. The algorithm, if successful, outputs a strong classifier, which is a thresholded linear function of the selected weak classifiers. This function uses as weights for each of the weak classifiers the associated weights that were determined through the iterations of the algorithm. The outline of the algorithm as used in [5], a discrete variant of AdaBoost, is shown in Algorithm 1.

The many advantages of the algorithm is that it is fast and is relatively straightforward. There does not exist any parameters to be tuned by the user (apart from the number of weak classifiers to be used, T). It is a meta-algorithm in that it requires no prior knowledge of the weak classifiers to be able to combine them in the algorithm. Its power lies in the fact that it is possible to efficiently convert weak classifiers into a strong classifier. One of the defining advantages, according to Freund and Schapire

Algorithm 1 AdaBoost

Input: Samples $(x_1, y_1) \dots (x_n, y_n)$ where x_i are the images and $y_i = 1$ for the female and 0 for male samples.

Initialise weights $w_{1,i} = 0.5/\mathrm{F}$, $0.5/\mathrm{M}$ for $y_i = 0, 1$ respectively, where F and M are the number of female and male samples.

For $t = 1, \dots, T$ (maximum number of weak classifiers to use):

1. Normalise the weights $w_{t,i}$ such that $\sum_i w_{t,i} = 1.0$
2. For each weak classifier, C_j, see how well it predicts the classification. Measure the error with respect to the weights w_t:

   $$\epsilon_t = \sum_i w_{t,i} \left| C_j(x_i) - y_i \right|$$

3. Choose the weak classifier (denoted C_t) with the lowest ϵ_t.
4. Update the weights:

   if example is classified incorrectly:

   $$w_{t+1,i} = w_{t,i}$$

   else

   $$w_{t+1,i} = w_{t,i} B_t$$

   where

   $$B_t = \frac{\epsilon_t}{1 - \epsilon_t}$$

The result of the strong classifier is

$$S(x) = \begin{cases} 1 : & \sum_{t=1}^{T} \log(\frac{1}{B_t}) * C_t(x) \geq 0.5 \sum_{t=1}^{T} \log(\frac{1}{B_t}) \\ 0 : & \text{otherwise} \end{cases}$$

[36], is that it comes with a set of theoretical guarantees given sufficient data and a weak learner that can reliably provide only moderately weak hypotheses. This provides a different way of looking at the problem of classification. Instead of attempting to design a classifier that is accurate over an entire testing space, the algorithm allows one to focus only on finding weak classifiers that need only perform better than random.

The algorithm does come with restrictions however. The performance of the algorithm is reliant upon the data that is input, and if not given sufficient data may not perform well. It is also reliant upon the weak classifiers that are used. If such classifiers are too complex or are too weak (do not perform better than random enough), the performance will suffer. Another, restriction of the algorithm is that it is susceptible to noise.

As previously mentioned, the AdaBoost algorithm focuses on samples that are incorrectly classified by weak classifiers. In effect, this gives the algorithm the ability to identify outliers (i.e. samples that are either mislabelled or inherently ambiguous). If the number of such outliers is large in a sample set, the performance of the algorithm can be adversely affected. To mitigate this potential problem, Friedman et al. [37] suggested a variant known as "Gentle AdaBoost". This variation affords less emphasis on the misclassified samples in the training schedule of the algorithm. A variation of the AdaBoost algorithm that refined the setting of the weights of the weak classifiers was also introduced by Schapire and Singer [38], in a variant called "Real AdaBoost".

It has been mentioned, that the performance of the algorithm is reliant upon the weak classifiers used. The types of classifiers used by Viola and Jones [5] is that of Haar-like features. These are image patches of alternate white and black regions that are placed at various locations across an image, and the sum of the pixel values under each of the regions are subtracted from each other providing a value to be used as the weak classifier. These image patches are Haar-like features due to their resemblance to Haar basis functions. A subset of some of the features can be seen in Figure 3.6.

These features can be calculated efficiently using a method known as the "Integral Image". This method sums up the pixel values in an image for each pixel in the previous rows and columns (i.e. the top-left-most rectangle) up to that pixel and thus allows for the calculation of sum of the pixel values under a rectangular region to be computed using only three binary operations. This is illustrated, in Figure 3.7, which appears in [5].

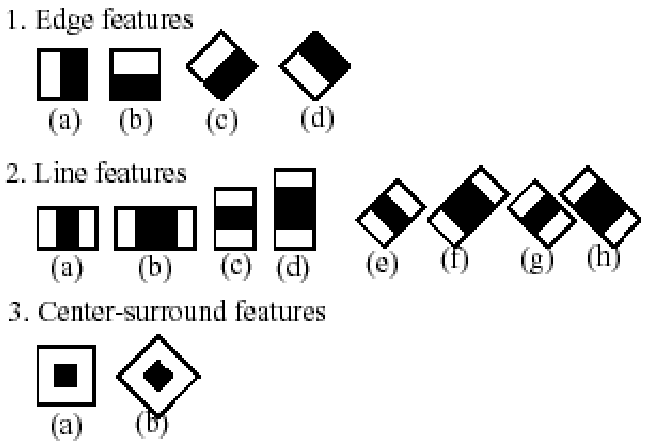


Figure 3.6: Haar-like features.

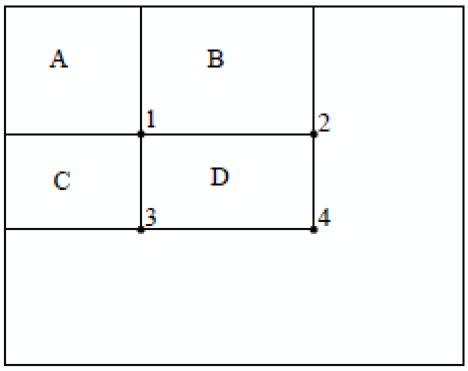


Figure 3.7: An integral image.

In Figure 3.7, the value of the sum of the pixels in rectangle D is found by four memory references and three binary operations. The value at location 1 is the sum of the pixel values in rectangle A. Similarly, at 2 it is $A + B$, and at 3 it is $A + C$. Thus the sum within D can be computed as $4 + 1 - (2 + 3)$.

For the purposes of the training of our classifier, Haar-like features will be used for the reasons that they can be calculated conveniently as outlined. It should be noted that this method is different to that of Baluja and Rowley [18], who used pixel comparisons as weak classifiers.

3.4.2 Training

To train the classifier, the OpenCV module *haartraining* was used. As inputs, this module requires text files listing the positive and negative sample filenames. These lists had been prepared through the processing performed in Section 3.3. Some useful tutorials for the usage of this module can be found at [39] and [40].

As part of the training, the parameters that were used in the training were a minimum hit rate of 0.999 and maximum false alarm of 0.5. The number of positive (female) samples that were used in the training was 990. The number of negative (male) samples used in the training was 1402. It is stated on [40], that a ratio of negative to positive samples of 2 is preferable for good quality results. In the case of our training it was restricted to 1.42 due to the limited number of face samples available.

As weak classifiers, the full set of Haar-like features available under the module was used. A classifier was also trained using the alternative Local Binary Pattern ("LBP") weak classifiers.

Classifiers were also trained using different versions of AdaBoost that are available under the module, namely Discrete AdaBoost and Gentle AdaBoost.

Chapter 4

Evaluation of the Classifier

In order to measure the performance of the trained classifiers it is necessary to determine some specific measurements that we can use to compare the classifiers. It is also necessary that rigorous statistical testing be used to ensure that the testing is consistent across the classifiers.

In this chapter, we will outline the statistical set up of our testing methodology, we will provide the results of our testing under this methodology, and we will provide commentary on these results.

4.1 5-Fold Cross-Validation

It has been noted that previous research has tested their classifiers on samples that were used in the training of the classifiers [12]. This method of testing does not give a representation of the performance of a classifier on an independent dataset. It may be the case that the classifier itself may suffer from over-fitting and hence perform well in classifying these samples, whilst when tested on an independent dataset of samples, the performance may vary significantly.

A technique that is commonly used to verify the performance of a classification scheme on an independent dataset is cross-validation. This method involves the partitioning of a dataset into two distinct sets of training and testing samples, training a classifier using only the training set and testing the classifier on the testing set. This method may be repeated several times with different partitions to get an average representation of the performance of the classifier. Such techniques are called k-fold cross-validation. In this case the dataset is partitioned into k subsets. For each of the k stages, $(k-1)$ of the subsets are used as the training set, with the classifier tested on the outstanding subset. Each subset is used only once as the testing set under the k stages. The performance of the classifier can then be averaged across the results of the performance for the k trained classifiers.

This technique is commonly used in the validation of gender classification schemes [18, 17]. The value for k that is most used is 5. For the purposes of assessing the performance of our trained classifiers this is the technique that shall be used.

In order to perform this, it was necessary to split our sample set into 5 separate subsets. A C++ application was built that would read the text files listing the positive and negative samples file locations and names. From these lists a set of 5 subsets of the samples was derived randomly. These subsets were combined 5 times in such a way that each subset was partitioned as the training set only once. Text files were subsequently generated for each of the training and respective testing sets.

It was necessary to train each of the classifiers 5 times using each of the different partitions of training samples. The training was performed as listed in Section 3.4.2. This was performed for each of the Gentle AdaBoost, Discrete AdaBoost and LBP weak classifiers variations of the classifier.

For each of the 5 trained classifiers (under each variation), the classifier attempted to classify the samples as listed under the testing sets' file lists. In a separate file, it was recorded whether the classification result was a true positive, true negative, false positive or false negative.

It should be noted that the testing samples are images of faces pre-processed in line with the methods outlined in Section 3.3. The reason for using the pre-processed images is that we are looking to ascertain the performance of the gender classifier solely. The techniques that were used in this section, such as the rapid object classifiers for faces, eyes and mouth, will have their own errors and we wish to eliminate the effect of such errors so that we can focus on the performance of the gender classifier. This method is in line with that used in much of the literature [18, 21, 17].

4.2 Evaluation Parameters

In order to assess the performance of the classifier we must define some parameters under which to measure it. In the following definitions we shall denote true positives, true negatives, false positives and false negatives as TP, TN, FP and FN, respectively. We shall also denote cardinality as $|\cdot|$.

An obvious parameter in assessing the performance of a classifier is that of its *accuracy*. This is a measure of the classifier's closeness of correctness to the actual correctness. It is more formally defined by the following equation.

$$\text{accuracy} = \frac{|TP| + |TN|}{|TP| + |FP| + |TN| + |FN|} \tag{4.1}$$

The *precision* of a classifier (or *positive predictive value*) is the proportion of samples which are correctly classified as positive as a proportion of all samples that are classified as positive.

$$\text{precision} = \frac{|TP|}{|TP| + |FP|} \tag{4.2}$$

The *sensitivity* of a classifier measures the proportion of actual positives that were correctly classified as such.

$$\text{sensitivity} = \frac{|TP|}{|TP| + |FN|} \tag{4.3}$$

Conversely, the *specificity* of a classifier measures the proportion of actual negatives that were correctly classified as such.

$$\text{specificity} = \frac{|TN|}{|TN| + |FP|} \tag{4.4}$$

It can easily be seen that,

$$\text{accuracy} = \text{sensitivity} + \text{specificity} \tag{4.5}$$

We shall assess the performance of each of the trained classifiers under these parameters.

4.3 Evaluation

Having calculated the parameters for each of the 5 trained classifiers under each configuration, the results were averaged across the 5 folds. It was found that the best classification scheme was that of Gentle AdaBoost using Haar-like features. It achieved an accuracy of 90.8% and a precision of 88.2% under cross validation. The Discrete AdaBoost classifier's performance was very close, with 90.7% accuracy and 88.1% precision. It was found however, that using Gentle AdaBoost and LBP weak classifier did not yield as good results, with an accuracy of 80.4%. The results are presented graphically in Figure 4.1.

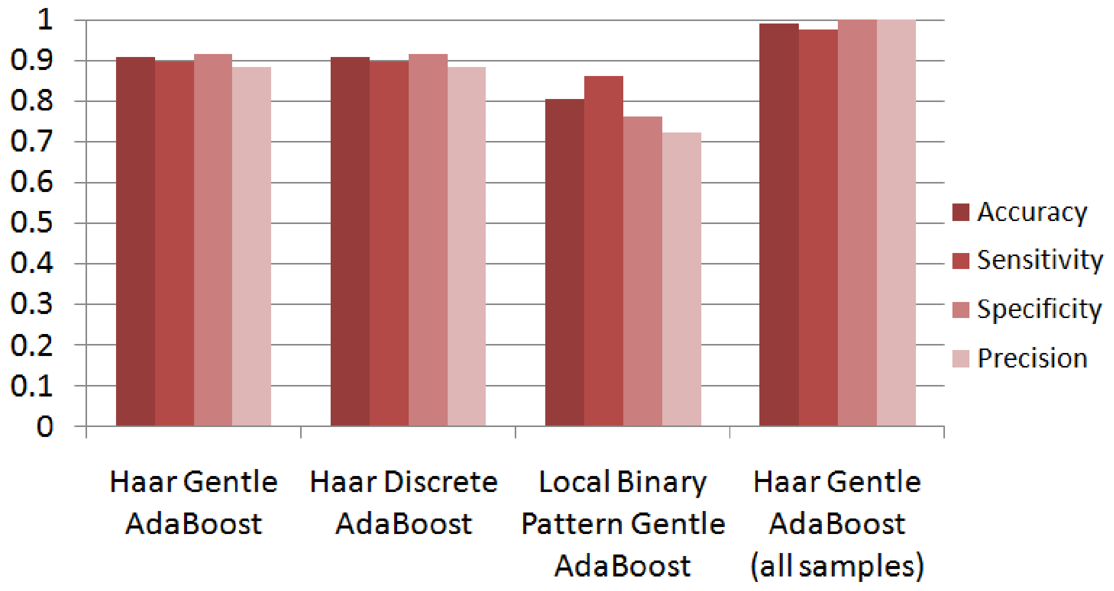


Figure 4.1: Classification Results.

When all of the samples were included in the training of the classifier, and the classifier was tested on the same samples, an accuracy of 99% was achieved. The data independence condition is not preserved here as it is under cross-validation. This information is useful however, in comparing the results with those achieved by Moghaddam and Yang [12]. It should also be noted that the fully trained classifier is the one that we be utilised as part of a dynamic advertising application.

In comparison to other methods, we find that our best classifier achieved an accuracy of 90.8% compared to that of 94.4% achieved by Baluja and Rowley [18] upon which our training methodology is based. The reasons for the differences are not immediately clear and may warrant further investigation. One of the differences between the methods was that the one presented here utilised Haar-like features as weak classifiers, whilst the one presented by Baluja and Rowley utilised pixel comparisons. It may be the case that pixel comparisons may offer stronger granularity in the weak classifiers as individual pixels can be compared as opposed to regions of pixels using Haar-like features. Pixel comparisons may also allow for comparisons between non-adjacent pixels which may also strengthen the weak classifiers. It was found however, that when the samples were mixed across the training and testing sets that the classifier presented here achieved an accuracy of 99%, whilst the one presented by Baluja and Rowley achieved a maximum accuracy of 96.6%. Similarly, the reasons for this remain unclear and may warrant further investigation. In comparison with the results obtained by Moghaddam and Yang [12], where samples were also mixed across the training and testing sets, the classifier presented here outperforms their classifier, with the classifier presented here achieving 99% accuracy, compared to 96.4% achieved using their method.

4.4 Remarks

The gender classifier that has been trained has shown to have a high degree of accuracy (90.8%). There exists potential for increasing this accuracy through different methods. As the training of the classifier was based upon the methodology outlined in [12], it may be worthwhile investigating if the classifier can achieve greater accuracy through the use of pixel comparisons as weak classifiers rather than Haar-like features. Intuitively, the sample images are quite small (20×20) and it may be worth investigating how using larger samples would affect the performance of the classifier. It was found previously that when using SVMs [12], the difference in classification performance with low resolution (21×12 pixels) versus higher resolution (84×48 pixels) was 1%, although it would be interesting to determine if this remains true for AdaBoost based classifiers.

Finally, there have since emerged other candidates for the weak classifiers, such as Dense SIFT and Gabor features [23], that can be used as part of an AdaBoost algorithm to train a classifier for gender classification. It may be worth investigating how the performance of the classifier would be affected if these features were to be used as weak classifiers in the training of the classifier.

Chapter 5

Application Design

The purpose of the training of the gender classifier is to allow it to be used as part of a wider dynamic advertising application. The trained gender classifier should enable a system to be built that can target advertisements more appropriately towards the viewer(s) of the advertisement. We will now describe the requirements of such a system, the design of the system, including the components and set-up, and some of the technologies that will be used in implementing the system.

5.1 Requirements

The system that will provide such functionality will encompass certain hardware as well as a computer application that will provide the functionality to the system. The application that is intended to be built needs to be able to perform certain specific tasks. These tasks can be broken down as follows:

1. **It must be able to detect that a person (viewer) of an advertisement is present.** This function needs to be able to work when a viewer may be some distance away ($> 3m$), or indeed much closer ($< 1m$). This function needs to be performed at close to real-time rates (i.e. > 10 frames per second) to allow for other functionality of the system to be performed smoothly.

2. **It must be able to determine the gender of such persons.** This requirement means that once a viewer is detected, the system needs to be able to

determine the gender of such a viewer. The temporal requirements for this stage are not as stringent as in the previous requirement, i.e. the system is permitted some time to classify the viewer's gender accurately (< 3 seconds). The distance requirement is also not as stringent as before ($< 3m$), but it is desirable that the system can maximise the distance at which it can determine the viewer's gender.

3. **It must be able to dynamically change an advertisement being displayed based upon the classified genders of the viewers of an advertisement.** The application should be able to change the advertisement being displayed once a viewer's gender has been determined to better target the advertisement towards that viewer. This must be performed within 1 second of determining a viewer's gender for the system to satisfy any useful purpose.

4. **It must be extendable to be used with other classification schemes (age, glasses, ethnicity etc.).** In implementing the application, it should be able to function equally well if other classification schemes were to be used in addition. This requirement enforces certain software engineering paradigms to be used. Namely, object-oriented design shall be used to allow for this

Along with the above requirements, there are additional desirable features that the application could provide.

1. **It can determine the number of viewers of the advertisement.** This is desirable as it can provide to advertisers information regarding how popular or viewed an advertisement is. This may be helpful in determining which locations are better suited for advertisements to be placed in.

2. **It can track the viewers of an advertisement as they move past it.** This feature can allow for different additional features to be implemented. For example, the application could determine how long a person viewed an advertisement for, and provide statistics relating to which advertisements attract the viewers' attention more. It would also allow the application to not have to update the gender of a viewer once it has been determined. It could also be possible to allow for the application to become interactive. For example, the display could react to stimulus such as the location or movement of the viewer.

3. **It can provide a model for how advertisements should be displayed when there are viewers of different genders simultaneously viewing the advertisement.** This model should allow for the case where if there is a strong majority of one particular gender of viewers that it may be possible to display the advertisement most appropriate for that gender.

In designing the system it is necessary to satisfy the above requirements. The desirable features will also be attempted to be implemented.

5.2 System Set-Up

The system we intend to implement will be composed of a number of hardware components. Each of these are necessary to be able to satisfy the requirements listed in Section 5.1.

Firstly, a camera is necessary to receive images of the scene in which the viewers will appear. This camera will provide an image feed to an attached computer which will run the application which performs the tasks of classifying the viewers. Finally, a screen attached to the computer will display the advertisement to the viewer. The camera will be placed directly above the advertisement facing in the same direction in which the advertisement is facing.

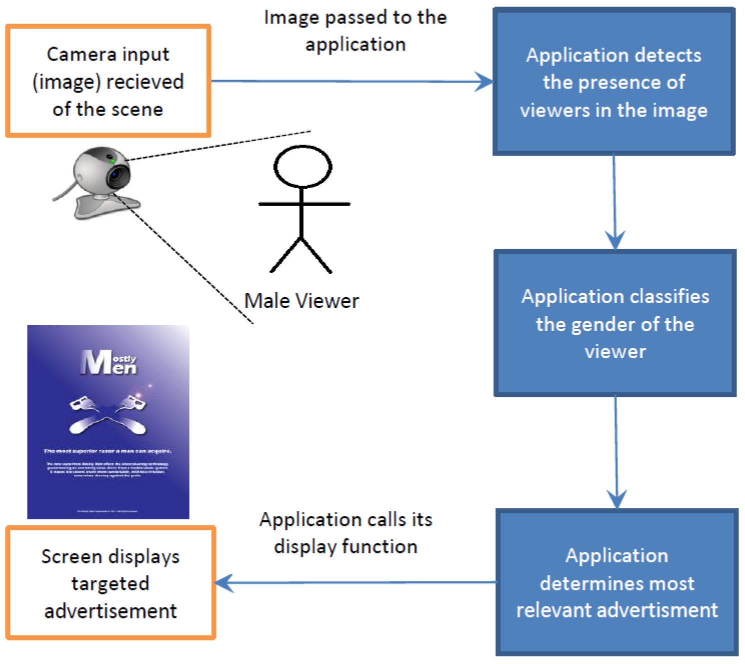
A visual summary of the set-up is shown in Figure 5.1

We can see from Figure 5.1, that the application has several functions (or its own components) that need to be implemented. If however, we are to implement the desirable features listed in Section 5.1, it is necessary to modify this design somewhat. A modification to this setup will help illustrate the components that will be needed. Such a modified set-up can be seen in Figure 5.2

With this modified set-up there are additional components of the application that will need to be implemented. We shall now detail these components.

5.3 Application Components

From the previous section, we are able to identify some necessary components for the application. These components are as follows:



Camera input (image) recieved of the scene
Image passed to the application
Application detects the presence of viewers in the image
Male Viewer
Application classifies the gender of the viewer
Mostly Men
Application calls its display function
Screen displays targeted advertisement
Application determines most relevant advertisment

Figure 5.1: The set-up of the system.

1. **Pre-processor.** The original image that is received from the camera will need to undergo some pre-processing before we perform some of the intended functions. Such pre-processing may include background determination and subtraction.

2. **Viewer Detector.** A component will need to be implemented that can detect the presence of a viewer in the image. Of vital importance in the implementation of this component is that it must perform at an optimal speed and accuracy.

3. **Viewer Tracker.** A component will need to be implemented that can track viewers between frames (images) received from the camera. This should be able to distinguish between viewers that are present and also be able to communicate with the viewer detector to determine if newly detected viewers are already being tracked.

4. **Gender Detector.** A component must be implemented that can determine the gender of a tracked viewer. It will take as inputs the tracked viewers and for each return a result of the likely gender.

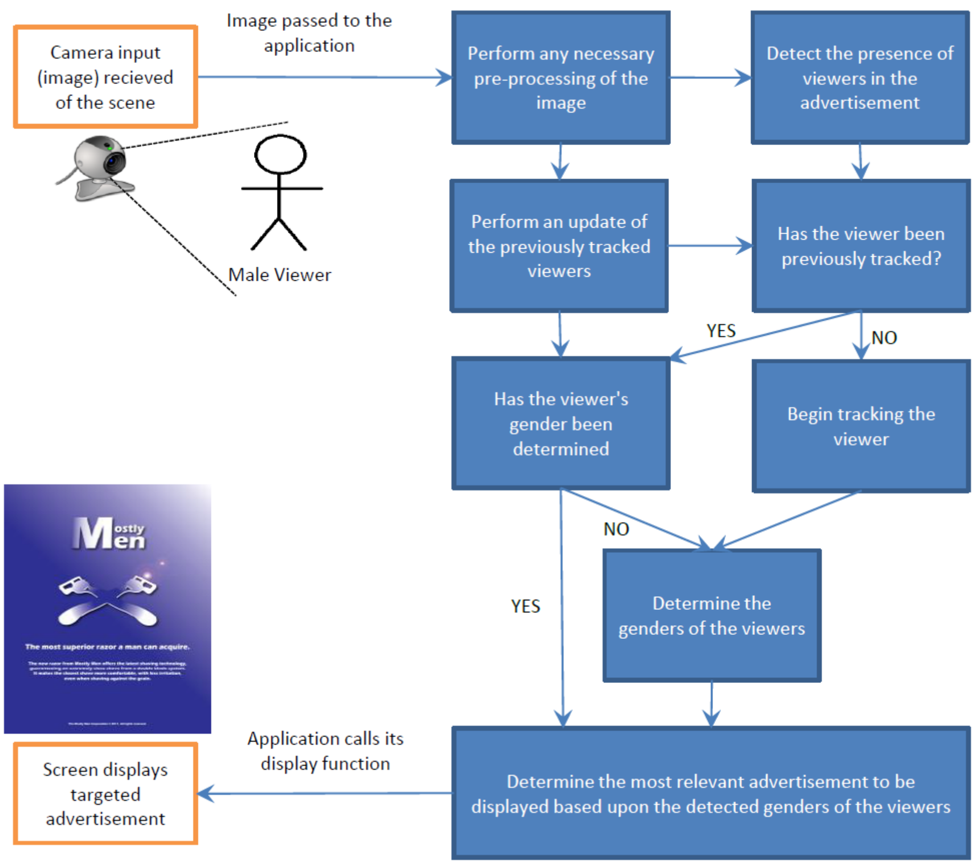

Camera input (image) recieved of the scene
Image passed to the application
Perform any necessary pre-processing of the image
Detect the presence of viewers in the advertisement
Male Viewer
Perform an update of the previously tracked viewers
Has the viewer been previously tracked?
YES
NO
Has the viewer's gender been determined
Begin tracking the viewer
NO
YES
Determine the genders of the viewers
Mostly Men
The most superior razor a man can acquire.
Application calls its display function
Screen displays targeted advertisement
Determine the most relevant advertisement to be displayed based upon the detected genders of the viewers

Figure 5.2: The modified set-up of the system.

5. **Advertisement display.** A component must be implemented that can display the most relevant advertisement. It will take as an input the likely genders of the tracked viewers and subsequently display an advertisement based on the result of some processing of this information.

We shall discuss how each of these components is implemented in the following chapter.

5.4 Technology

In order to build the application it is necessary that we select some technologies to be used. It was decided that the application will be built using C++ due to the requirement that we utilise object-oriented design patterns (as mentioned in Section 5.1).

It is also necessary that we make use of some code libraries in implementing the application. It was decided that OpenCV satisfied a number of our needs in relation to how to implement the application. The reasons for this are similar to those detailed in Section 3.1.2. It provides a number of useful functions, especially in relation to receiving input from a camera and being able to perform image processing. As our classifier has been previously trained using OpenCV, for specific use in OpenCV, it was also a requirement that at least some aspects of OpenCV be utilised.

As we are looking to build an application that will provide a display, it may be worthwhile utilising a library to perform some Graphical User Interface ("GUI") functionality. Qt [41] is a cross-platform application framework that provides many useful tools and functions for the construction of GUI interfaces. It is implemented in C++ and provides much useful functionality such as multithreading. Versions of the framework are available under Open Source licenses.

There exists an Open Source project, qt-opencv-multithreaded [42], that combines the functionality of the above libraries in an application that performs image processing on images received from a camera input. The application utilises the multithreading functionality of Qt to have threads that separately handle an image buffer from the camera input and the processing of the images. It is available under MIT licence, which means that permission is granted for persons to use, modify, publish, licence and sell copies of the software.

As this application provides much of the groundwork in setting up a GUI application which can perform image processing on images received from a camera, it is our intention to extend this software to be able to perform the tasks of a Dynamic Advertising Application. The software as distributed performs only basic image processing tasks (edge detection, filtering) but it is coded using object oriented design to allow for users to extend the software to perform their own intended processing.

Chapter 6

Application Implementation

In this chapter, we will detail the implementation of the different components that make up the Dynamic Advertising application. We have previously detailed these components in Section 5.3. We will introduce some concepts and techniques that will enable us to implement these components as well as providing detailed descriptions of the properties of these components and of how they are suited to the implementation of the relevant components.

6.1 Background Determination

One of the components that we implement is that of the pre-processor. This component is responsible for performing the initial processing of a frame received from the camera to allow it to be processed more easily by the later components such as the viewer detector. One of the tasks that we wish for this component to perform is to highlight moving objects in an image to allow the viewer detector to search in these regions for a potential viewer. This is performed through a method known as background detection.

Various methods of detection of people also utilise background determination [3, 7, 8] to be able to segment regions of an image for the detection of people in such regions. Their approaches to the determination of a background image vary from static background images [3], median filtering [7] to running averages [8]. The method we chose to implement was that of stable values. This method is outlined in Algorithm 2.

Algorithm 2 Stable Values Background Determination

For the initial background image, $B(\mathbf{x})$, use the first available frame, $I_1(\mathbf{x})$, where $I_t(\mathbf{x})$ denotes the t^{th} frame. The proposed new background is denoted $B_p(\mathbf{x})$, and the proposed count for a pixel $\mathbf{x}_i$ is denoted by $c(\mathbf{x}_i)$.

```
for Each received frame, $I_t(\mathbf{x})$ do
    for Every pixel $\mathbf{x}_i$ in the current background image, $B(\mathbf{x})$ do
        if $c(\mathbf{x}_i) = 0$ and $|B(\mathbf{x}_i) - I_t(\mathbf{x}_i)| > L_{\text{prop}}$, where $L_{\text{prop}}$ is a specified threshold
        above which a change has deemed to occurred in the background then
            Propose a new background pixel
            $B_p(\mathbf{x}_i) = I_t(\mathbf{x}_i)$
            $c(\mathbf{x}_i) = 1$
        end if
        if $|B(\mathbf{x}_i) - I_t(\mathbf{x}_i)| > L_{\text{prop}}$ and $|B_p(\mathbf{x}_i) - I_t(\mathbf{x}_i)| < L_{\text{tol}}$, where $L_{\text{tol}}$ is the tolerance
        about the proposed pixel value then
            Increment the count for that pixel
            $c(\mathbf{x}_i) = c(\mathbf{x}_i) + 1$
        else
            Reset the accepted proposed pixel counter, $c(\mathbf{x}_i) = 0$
        end if
        if $c(\mathbf{x}_i) \geq L_{\text{n}}$, where $L_{\text{n}}$ is the number of frames that a proposal needs to be
        accepted for. then
            Accept the proposed pixel value in the background image
            $B(\mathbf{x}_i) = B_p(\mathbf{x}_i)$
            $c(\mathbf{x}_i) = 0$
        end if
    end for
end for
```

In this algorithm the initial background is set to be the first frame. Basically, the algorithm works by proposing a new background pixel if the corresponding pixel in the current image is outside a given threshold of this pixel L_{prop}. If for a subsequent number of frames (L_{n}) the value of this proposed pixel is within a certain tolerance L_{tol} of the corresponding pixel value in each of the frames, then the proposed background pixel is promoted to being a background pixel. The threshold values here were tuned by experimentation.

The advantages of this method is that it deals well with changes in lighting in a scene or when an object is placed in or removed from a scene. Static background images can handle neither of these cases well. With running averages, it has an undesirable artifact of a blurring effect in the background as an object moves across the scene. This is also observable in median filtering. A review of common background subtraction techniques can be found in [43].

In order to detect moving objects, we subtract the current frame from the background image, and if the absolute value is above a certain threshold we indicate that this pixel is a foreground object in a mask image. If we denote $B(\mathbf{x})$ as the background image for all pixels $\mathbf{x}$, the current frame by $I_t(\mathbf{x})$, and the threshold by L_t then the mask image $F(\mathbf{x})$ is given by the following equation.

$$F(\mathbf{x}) = \begin{cases} 1: & |I_t(\mathbf{x}) - B(\mathbf{x})| \geq L_t \\ 0: & \text{otherwise} \end{cases} \tag{6.1}$$

Once again, the value of the threshold was tuned by experimentation in the implementation. It was found that there was a trade-off between using a high value and using a low value for this threshold. It was found that at a low value, there existed regions in this mask image that should not be highlighted as foreground objects, whilst at high values certain key regions that should be highlighted were not present. A way to counteract this was to utilise some morphological operations to remove some of the small regions that existed in the mask image when a lower threshold was used. The mask image was closed (a dilation followed by an erosion) to fill in small holes in the foreground objects. Subsequently, the mask image was opened (an erosion followed by a dilation) to remove small regions that may exist in the mask image. As part of this component, we were also able to return the location and dimension of the minimum bounding upright rectangle of the non-zero values of the mask-image. This rectangle will be used in the following viewer detection component. Sample results of this background subtraction stage can be seen in Figure 6.1.

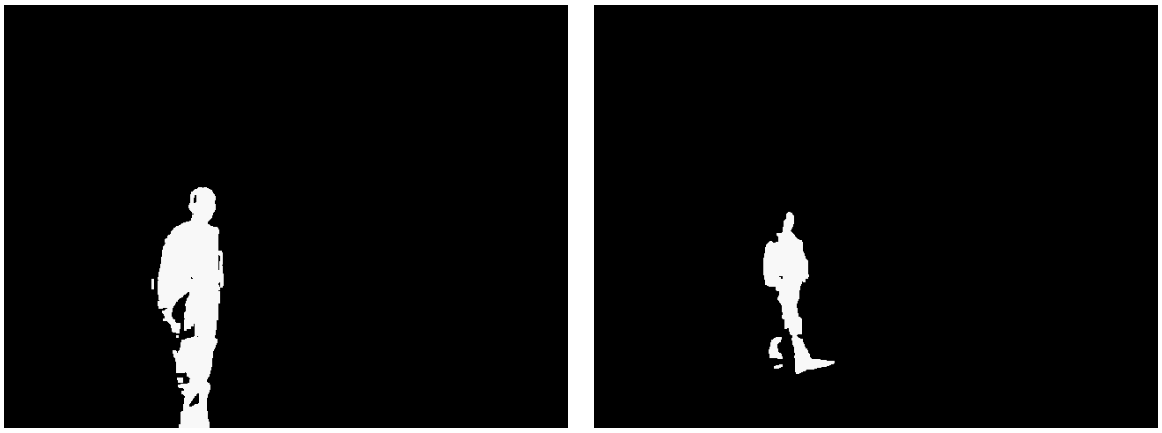


Figure 6.1: Some mask image results of the background detection.

6.2 Face Detection

As part of the requirements of the Dynamic Advertising application, we must be able to identify the presence of viewers of an application. It is reasonable to define a viewer as someone who is directly facing an advertisement. In this way, we can reestablish our viewer detection component as a face detection component.

To achieve the task of identifying the face, we utilise the method described previously in Section 3.3.1. This method utilises the pre-trained classifier for frontal faces as distributed with OpenCV. This detection routine satisfies the requirements of the viewer detector component in that it is accurate and fast when performed on a full 640×480 image. It does represent however a large portion of the processing time available to the application (c. 60ms). As part of our implementation, we wished to maximise this performance through a number of optimisations.

Using the bounded rectangle that is returned from the background detection routine, we concentrate the detection of the faces within this region as this is where a viewer is likely to be present. The rectangle itself is expanded slightly horizontally to account for any small offsets where a face may be present. The rectangle is expanded upwards by an amount also, to ensure the full region of a face is included. It is likely that the first row of the original rectangle will begin at the top of a viewers head so this method of expanding upwards will better include the viewers' faces. If the current mask image is empty, indicating that there are no moving objects within the scene, the entire image is processed using the classifier.

As a result of this, generally only a small fraction of the image is ever needed to be classified for the presence of faces. This greatly decreased the time requirements of the classification. It was found that it was possible to increase the accuracy of the classifier to be able to detect viewers at even greater distances by taking the region of the image that was being passed to the classifier and resizing it to a larger scale. This resizing was performed using bilinear interpolation which provides fast but good quality results. The scaling factor was a ratio of the area of the full image (640×480) to the area of region of interest, up to a maximum limit of 2.0. At this level, a minimum speed of the classification stage was maintained, whilst the distance at which a viewer could be recognised was increased significantly.

6.3 Face Feature Detection

In order to be able to later perform the gender classification, it was necessary to perform some processing of any detected faces. The face detector component returns the locations of the rectangle that enclose the detected face. It was necessary to identify features of a viewer's face within that rectangle. These features were the viewer's left and right eyes and mouth. To perform this, a methodology in line with Section 3.3.1 is used. It is important that we have this consistency across the processing of the samples used in the training of the classifier and in the Dynamic Advertising application. The gender classifier has been trained on images where each of the eyes and mouth was found using pre-trained rapid object classifiers (trained using the AdaBoost algorithm). If we are to utilise this gender classifier to classify viewers' faces in a real-environment application, performing the same detection of the face features will likely give us the best results as it will mean that there will exist consistency across the features' locations in the training and real-environment faces.

Similarly to before, we restrict the detection of the face features to their most relevant region of the face rectangle. The viewer's left eye is only searched for in the top-left quarter of the face rectangle and conversely for the right eye. The viewer's mouth is only searched for in the bottom half of the face rectangle. This provides a definitive speed-up over searching the entire face rectangle for each feature. The parameters used in the classifier such as the scale factor can thus be increased due to the time afforded by this saving, which will mean that the classifiers for the face features will have greater accuracy.

It was found in initial tests however, that although these classifiers achieved a good level of accuracy, they sometimes did not find the location of the features to a good enough precision. It was necessary to enforce a validation step once these features have been detected. This validation step involves checking three conditions relating to the distances and angles between the features. Figure 6.2 illustrates the distances and angles that are involved in this check.

We call this validation the "T Test" for reasons that should be obvious from Figure 6.2. The three tests that the detected face features are required to satisfy in order for them to be considered valid are:

1. The angle between between the line joining the two eye positions, $L1$, and the horizontal, θ, must satisfy the following:

   $$-\frac{\pi}{6} \leq \theta \leq \frac{\pi}{6} \tag{6.2}$$

   The reason for this condition is that we would like the face to be reasonably well aligned. This test returns an invalid result if the face is too rotated about the axis coming out of the face (i.e. rolled) to expect an affine transformation to provide an accurate representation of a face when transformed to a 20×20 image.

2. The angle between the line joining the mouth position to the midpoint of $L1$, and the perpendicular bisector of $L1$, ϕ, must satisfy the following:

   $$-\frac{\pi}{8} \leq \phi \leq \frac{\pi}{8} \tag{6.3}$$

   The reason for this condition is also to ensure that the face is reasonably well aligned. This test returns an invalid result if the face is too rotated about the

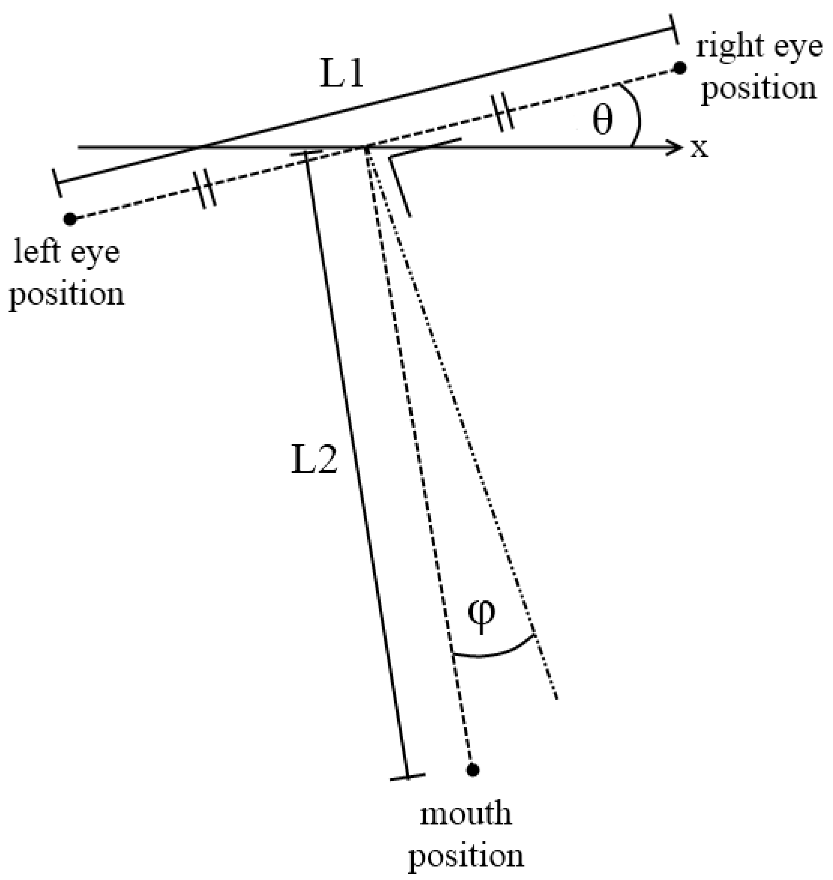


Figure 6.2: The T condition.

vertical axis extending out of the top of a viewer's head (i.e. turned) to expect an affine transformation to provide an accurate representation of a face when transformed to a 20×20 image.

3. The lines $L1$ and $L2$ must satisfy the following:

$$\|L1\| \leq \|L2\| \tag{6.4}$$

This condition is the final check that the face is reasonably well aligned. This test returns an invalid result if the face is too rotated about the horizontal axis extending out of the side of a viewer's head (i.e. tilted) to expect an affine transformation to provide an accurate representation of a face when transformed to a 20×20 image.

Once the face feature locations have been validated, this allows us to perform a registration to a $20{\times}20$ image thumbnail, in the same was as was performed in Section 3.3.2. The intensities of pixels in the thumbnail images are subsequently normalised in line with the methods outlined in Section 3.3.3. This provides us with a standardised image that can now be used as the basis of a gender classification of the viewer.

6.4 Gender Classification

The gender classifier component takes as an input a 20×20, transformed, normalised image of a viewers face. The trained classifier is run on this image and returns a positive result if it has detected the presence of a female face and a negative result otherwise. It was found in Section 4.3, that the classifier is 90.8% accurate. Thus, over time one would expect a number of misclassifications of viewers' genders.

In order to deal with this situation, a probability model for a viewer's gender is constructed. This model uses a weighted sum of the previous classification results and the current classification result. If we have a viewer V say, we denote its classification as a female at a time t as $P_t(V)$. This can take either the value 1 (female) or 0 (male). Thus, we can determine the probability of a certain gender over a number of classifications. We denote the probability of a viewer's gender at a time t by $P_{sum,t}(V)$. It is initially set to $\frac{1}{2}$ for $t = 0$. It is subsequently computed using the following equation:

$$P_{sum,t}(V) = w * P_{sum,t-1}(V) + (1 - w) * P_{t-1}(V) \tag{6.5}$$

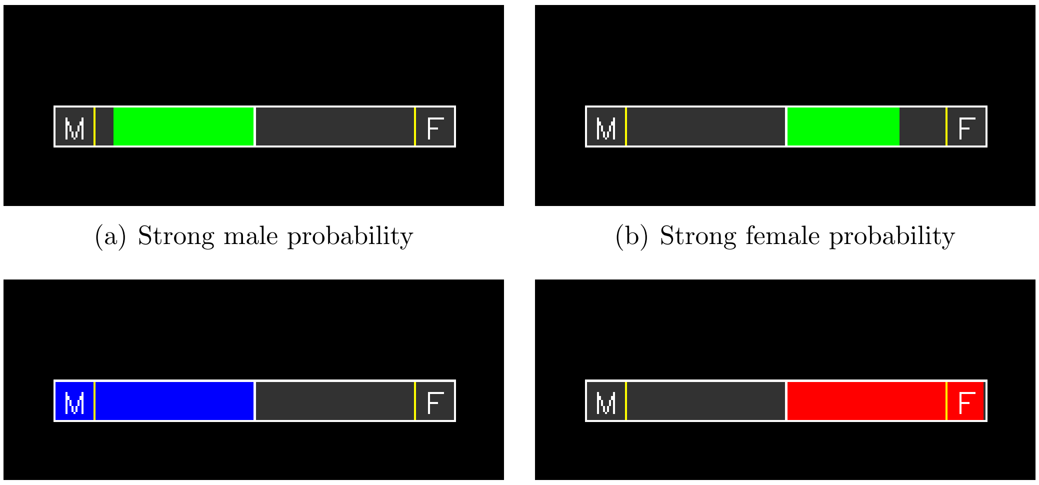
In this way, if a misclassification of the viewer occurs in line with the classifier's accuracy (i.e. 9.2% of the classifications), the probability will still converge on the correct answer.

It should be noted that in order for this probability model to be effective, it is necessary that we can identify the same viewer in multiple frames. The process for achieving this is covered in the next section.

In implementing the probability model, the weight, w was tuned based on experimentation. A level at which it converged relatively rapidly whilst did not give too much weighting to any misclassifications was found.

As part of the application, once we have determined the gender of a viewer to a certain confidence level, we wish to lock in this classification and no longer update the value for $P_{sum,t}(V)$. If the value was within a certain absolute difference of zero or one the gender was permanently selected as male or female respectively.

To display the results of this classification on screen as part of a debugging process or otherwise, a visual representation of the results was devised. The value is represented as a "probability bar", where the extension of a green bar towards an extemum left (male) or right (female) indicates the current probability. If the probability breaches the confidence level at which it is permanently classified for that viewer, the probability bar changes colour to blue for male and red for female, and is no longer updated. This gives a succinct visual representation of the gender classification results when observing it being performed on-screen. Examples of the probability bar are shown in Figure 6.3.



(a) Strong male probability

(b) Strong female probability

(c) Permanently classified male probability

(d) Permanently classified female probability

Figure 6.3: Examples of how the "probability bar" visually represents a viewer's probable gender.

As we have mentioned previously, the ability to use this probability model is predicated on the fact that we can track a viewer's face between frames. We will now deal with how we tackle this issue.

6.5 Face Tracking

The tracking of viewers across frames is a task upon which certain aspects of components of the Dynamic Advertising application rely. This task is a commonly tackled problem in the area of Computer Vision. The approach that we implemented combines aspects of different solutions to provide a more generalised and powerful method of tracking viewers' faces. We will now list some of the concepts used in the tracking of a viewer's face, including the properties that make them suitable for this task, and how they fit in to a larger face tracking component.

6.5.1 Mahalanobis Distance

A primitive, albeit effective, method of tracking faces is that of distance measurement. If one knows the location of a viewers' face in one frame and the location of the viewers' face in the subsequent frame, the faces can be tracked by measuring the distance between the faces in the first frame and the faces in the next frame. It can be reasoned that the minimal distances are between the faces in each frame that correspond to the same viewer in the two frames. This is based on the assumption that the viewer has a limited velocity at which they can move at between frames. Another assumption that is made here is that we can accurately find each viewer's face in each frame. Generally, this is not the case for the type of face detection method we use. The classifier that is used to detect faces can only classify well-aligned reasonably frontal faces. It is also possible that the classifier may simply not find the face (a false negative). Should a viewer's face either become obstructed or rotated in any way, it is quite possible that the classifier will not be able to locate the face.

There exists numerous distance measures that can be used for determining the distance between two face locations. The most widely used is that of Euclidean distance. Given a point $\mathbf{x} = (x_1, y_1)$ and a point $\mathbf{y} = (x_2, y_2)$, the Euclidean distance between point $\mathbf{x}$ and point $\mathbf{y}$, $d_{\mathrm{E}}(\mathbf{x}, \mathbf{y})$, is given by:

$$d_{\mathrm{E}}(\mathbf{x}, \mathbf{y}) = |\mathbf{x} - \mathbf{y}| = \left((x_1 - x_2)^2 + (y_1 - y_2)^2\right)^{\frac{1}{2}} \tag{6.6}$$

If we were to have a group of points $\mathbf{X} = (\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n})$, and a group of points $\mathbf{Y} = (\mathbf{y_1}, \mathbf{y_2}, \ldots, \mathbf{y_n})$, a common way of measuring the distance between the groups of points is to calculate the mean point of the set $\mathbf{X}$, $\bar{\mathbf{x}}$, and the mean point of the set $\mathbf{Y}$, $\bar{\mathbf{y}}$, and calculating the Euclidean distance between these two means.

Another distance measure, known as quadratic distance or *Mahalanobis distance* can measure the separation of two groups of points. Supposing as before, we have two groups with means $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$, the Mahalanobis distance is given by the following equation:

$$d_{\mathrm{M}}(\mathbf{X},\mathbf{Y}) = \Big((\bar{\mathbf{x}} - \bar{\mathbf{y}})^T \, \mathbf{S}^{-1} \, (\bar{\mathbf{x}} - \bar{\mathbf{y}}) \Big)^{\frac{1}{2}} \tag{6.7}$$

where $\mathbf{S}$ is the covariance matrix between the two groups.

The Mahalanobis distance has some desirable properties for the purposes of being able to measure the distance between groups of points. It takes into account correlations of the data set. This is desirable if we are trying to determine the matching between two sets of points that may be present in face regions between two frames, as we can reason that the points should be highly correlated. It differs from the Euclidean distance in that it is also scale invariant. This is a highly desirable property in matching points in face regions between two frames. If a face is moving towards or away from a camera the distribution of points will be scaling and hence the Mahalanobis distance measure will be able to account for this and return a measure which is not affected by this scaling.

In our implementation, we calculated the covariance matrix for a set of points and calculated the covariance matrix between the two sets of points via pooled covariance. In calculating the covariance matrix for a set of points it was necessary to first centre the points on the mean of the set. The covariance matrix was inverted by explicit 2×2 matrix inversion. This then allowed us to calculate the Mahalanobis distance between two sets of points.

6.5.2 Speeded Up Robust Features

As part of the Face Tracking component we may wish to find some interest points within a face region to allow for them to be tracked. In order to do this we must select the type of features that we wish to track. A type of feature points that have many desirable properties are Speeded Up Robust Features ("SURF") [44]. These feature points can be calculated for an image very efficiently, making use of an integral image (Section 3.4.1) for image convolutions. The calculation of the feature points makes use of a Hessian matrix (which is calculated by the convolution of the Gaussian second order derivative with an image). In the description of the feature points the method makes use of differences in the image patches surrounding a detected feature to provide sub-pixel accuracy and orientation for the feature.

SURF features offer many desirable characteristics for use. They can be calculated very efficiently. This is very desirable considering the fact that the Dynamic Advertising application needs to be able to operate at close to real-time rates. They are invariant to scale, which as mentioned previously is something that will necessarily be encountered when a face region is moving towards or away from a camera. The features also have other desirable characteristics such as robustness, repeatability and ease of matching.

As part of the OpenCV distribution, a SURF feature detector is packaged. For our implementation, we perform this feature point detection in a region of the detected faces. We ensure that the features are within the actual face region by taking the rectangle that has been found to contain a face. We determine the minimum bounded ellipse that will fit inside this rectangle. We subsequently scale down this ellipse. When performing the feature detection we pass a mask for the rectangle where each pixel is non-zero if it within this scaled ellipse or zero when it is without. In this way we ensure that no SURF features for parts of the background that may be present in the rectangle are classified as features to track, as this would likely cause problems as the background would not move with the face.

6.5.3 Pyramidal Lucas Kanade

In order to be able to match feature points between faces an algorithm that was utilised was that of Pyramidal Implementation of the Lucas-Kanade Feature Tracker [45]. This method is a differential method for optical flow estimation. It makes use of least square calculations in a 3×3window about a point. The implementation here also utilises images pyramids to locate the points at successive levels for greater accuracy. One of the advantages of the method is that it is less sensitive to noise than point wise methods.

The method however, makes the assumption that the displacement of points in the neighbourhood of a feature point is the same. For feature points of a face, this may not necessarily be the case, as there can exist internal flows in a face region, such as a changing expression of a viewer. As part of our implementation, we set a threshold below which if the number of tracked feature points dropped, a new set of feature points would be generated to track. The algorithm also assumes that the feature points will not be undergoing any great changes in velocity. This may not necessarily be the case where a viewer may suddenly change velocity. In such a case, it is also necessary to find new feature points if the number that were tracked goes below a certain threshold number.

6.5.4 Mutual Information

Mutual information gives a measure of the shared information between two probability distributions. In the case here we can think of images as probability distributions. Mutual Information has been used in the image processing domain for registration of images [46]. However, it has also been used as a method of tracking faces of people [3]. In our implementation, we use it for this purpose.

Mutual information comes from a branch of science known as Information Theory. The mutual information between two discrete random variables, X and Y, can be calculated via their entropies, $H(X)$ and $H(Y)$, and their joint entropy $H(X,Y)$, by the following equation.

$$I(X;Y) = H(X) + H(Y) - H(X,Y) \tag{6.8}$$

where $I(X;Y)$ denotes the mutual information. For the purposes of using mutual information as a similarity metric, a variation known as normalised mutual information, $NI(X,Y)$ can be used, which satisfies the criteria for a metric.

$$NI(X,Y) = \frac{I(X;Y)}{max(H(X),H(Y))} \tag{6.9}$$

Further details regarding information theory and the derivations of the above identities can be found in Appendix A.

We calculate the mutual information between thumbnail (20×20) face images to allow for the matching of faces to be performed. We use a modified metric of mutual information, known as normalised mutual information, as mentioned above, to ascertain which faces have the greatest similarity. It was found during the implementation that often a viewer may be too far away for enough feature (SURF) points to be generated and tracked. It was also at such large distances that the measure of Mahalanobis distance of faces between frames may be outside the maximum threshold due to the fact that the viewer appears to move more quickly in the two-dimensional space in which this is measured, when they are further away. What were available to allow for viewer tracking to be performed however, were the thumbnail images. This allowed us to track faces at much greater distances. It was also found that a viewer may often turn away momentarily or that their face may become occluded for a time period. By being able to compare the thumbnail faces for a detected face with previously identified (and gender classified) faces, we were able to re-track this face. This enables us to satisfy the requirements of the application to be able to count unique viewers of an advertisement, as without this method, new viewers would have been counted, when they were in actual fact already counted.

To calculate the mutual information between two thumbnail images, we construct the joint entropy histogram. This histogram is 2-dimensional, with each dimension corresponding to each of the images. We go through each of the pixel locations and increment the relevant site of the histogram based on the pixel value in the two images. Once this is performed, we normalise this histogram so that it represents a probability distribution function. We calculate the marginal entropies by taking the sum of the probabilities of each row, $p(x)$ and substituting it into the equation for entropy.

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x) \tag{6.10}$$

We similarly do this for the columns to calculate the marginal entropy for the other image, $H(Y)$. Finally, we perform a similar calculation using all of the histogram sites to find the joint entropy, $H(X, Y)$. Substituting these values in to Equation 6.8, gives us a value for the mutual information. We can then use this value to find the normalised mutual information between the two images, $NI(X, Y)$, using Equation 6.9.

We implemented the above calculations using different varieties of bin sizes to help speed up the processing. In Figure 6.4, we can see the results of the calculation of the joint entropy histogram between two thumbnail face images of the same viewer. It should be noted that for visualisation purposes, the intensities have been scaled so that the histogram site with the largest value is pure white.

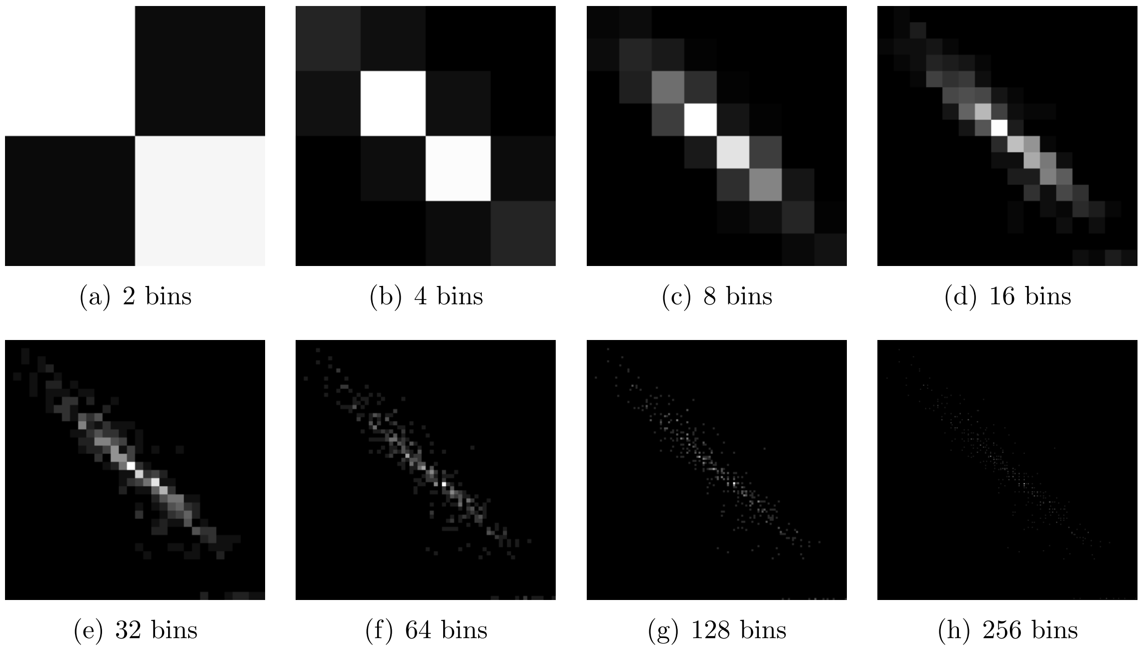


Figure 6.4: Sample Joint Entropy histograms for two thumbnail images of the same viewer for various different bin sizes. Note: the origin is the top-left corner.

6.5.5 Schedule for tracking

Now that the different ways in which the application tracks the viewer have been described, it is necessary to define how the schedule of the tracking methods is implemented. The schedule is broken down into three separate stages.

The first method by which we attempt to track a viewer is via the Mahalanobis distance. This is the least expensive way to attempt to match faces and so is attempted first. We calculate the distances between the current SURF features points for each face and the vertices of the rectangles of faces that have been detected using the face detector. We seek to match the faces that return the minimum distances by checking to see if this distance is below a minimum threshold level. If it is, we update the face with the new rectangle location and then use Pyramidal Lucas Kanade to attempt to locate the SURF features in the new rectangle. If the number of found features goes below a certain threshold we run the SURF feature detection again on the new face rectangle and update the face with these features.

If there remains any faces that have been detected in the new frame and faces that have been previously found, we attempt to match these faces by finding the match that maximises the normalised mutual information between the thumbnail faces. This value must be above a certain value for a match to be accepted. We also calculate the distance between such a match and if it is above a certain threshold value (different to the threshold value above), we reject the match. As above, once a match has been found we update the locations of the feature points.

Finally, we attempt to match the faces via the Pyramidal Lucas Kanade approach. If a face has been detected in the previous frame but cannot be matched to a face in the current frame, this implies we will need to track the face. To do this we use the SURF feature points of the face in the Pyramidal Lucas Kanade algorithm. This attempts to locate the positions of these features in the new frame. If a sufficient number of features are found in the new frame, and each of these features is not an outlier, we track and update the face by expanding the face rectangle to encompass all of the tracked features. We test for outliers using a Euclidean distance measure between its previous and new locations and reject those that are above a certain threshold. The tracking usually only ever reaches this stage if a viewer's face has been tilted or obscured and the face detector was unable to find it in the frame. This may also happen if the viewer stands extremely still for a period of time and becomes part of the background in the background detector. If this happens, however this method allows us to deal with such occurrences.

If a face that has been detected in the current frame was not matched to a previous frame it is tracked, and SURF Features are generated for the face, and it is added to the list of currently tracked faces.

If a face was previously found but was not tracked under any of the three stages above, a counter for it is incremented. If the face does not get tracked again within a certain number of frames (via maximum mutual information), the face will be discarded from the list of tracked faces.

The combination of these three stages means that the application uses the cheapest approaches first to track the viewers faces before trying to locate it by tracking the individual feature points. This enables the system to handle very difficult situations, such as when a viewer may turn or be obscured momentarily. It also allows, via the mutual information stage, to track viewers at reasonably large distances ($> 5m$). It has been observed that sometimes if the viewer is moving at too great a speed, the feature points cannot be tracked whatsoever. In, this case, the mutual information tracking stage also enables this to be rectified by being able to match images of the viewers' faces.

Please refer to Appendix B, for a video showing the tracking in progress.

6.6 Advertisement Display

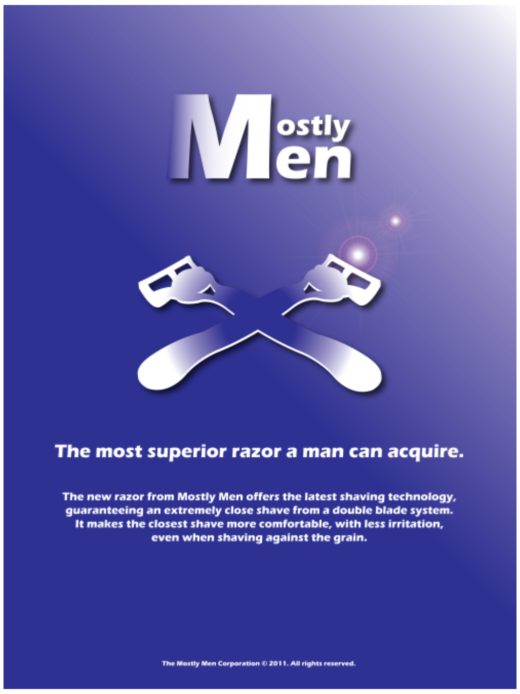
One of the requirements of the Dynamic Advertising application is that it can display the advertisement that is best targeted to the viewers of an advertisement. In order to perform this task, a probability based model is used to determine which advertisement to display. The probability model is similar to that used in Section 6.4. The average gender probability for the viewers of the advertisement that are currently being tracked is passed to the component that handles the displaying of the advertisement. We denote the current average gender result of the viewers by $f_t(\bar{V})$, where $\bar{V}$ denotes the average of the viewers. We denote the current value of detected genders by the display component by $f_{sum,t}(\bar{V})$ It is initially set to $\frac{1}{2}$ for $t = 0$. It is subsequently computed using the following equation:

$$f_{sum,t}(\bar{V}) = w_{ad} * f_{sum,t-1}(\bar{V}) + (1 - w_{ad}) * f_{t-1}(\bar{V}) \tag{6.11}$$

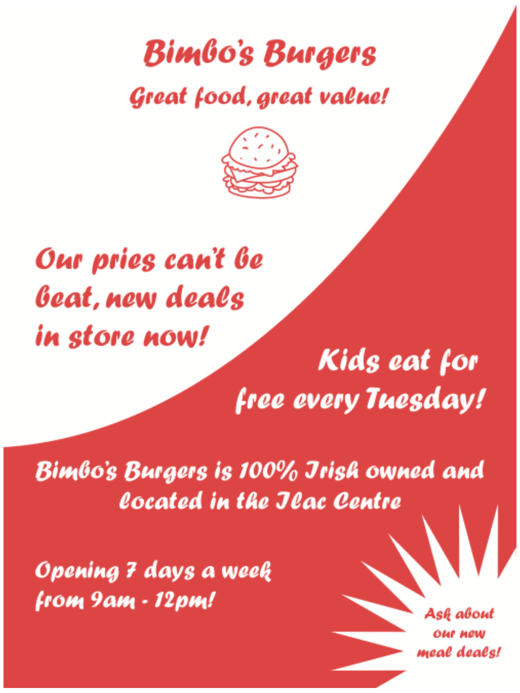
The weight, w_{ad}, was tuned manually so that the display responds to changes in the genders of viewers in a short time ($< 1s$ generally), whilst there is minimal flickering between advertisements over time.

Mock-up advertisements were drawn up for each of a male audience, a female audience and a mixed or default audience. These advertisements can be seen in Figure 6.5

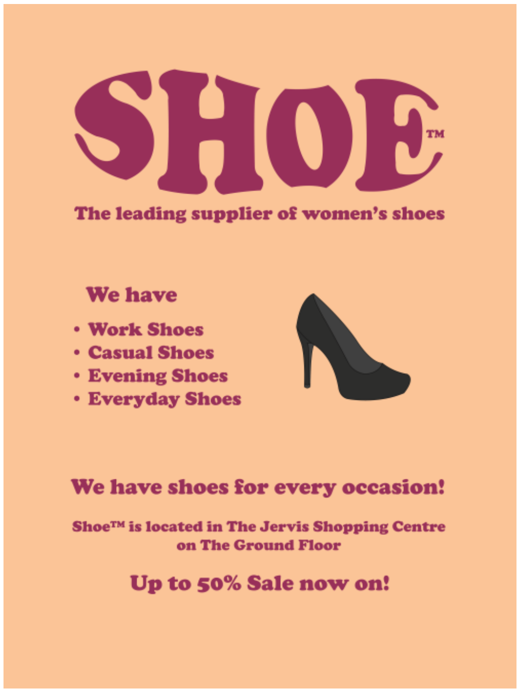
Once the value of $f_{sum,t}(\bar{V})$ reached below a certain limit above zero, or above a certain limit less than one. The advertisement that is being displayed is changed. These limits were tuned through experimentation. In this way, the advertisements are targeted towards whichever the most likely gender of viewers that are currently viewing the advertisement.


Mostly Men
The most superior razor a man can acquire.
The new razor from Mostly Men offers the latest shaving technology, guaranteeing an extremely close shave from a double blade system. It makes the closest shave more comfortable, with less irritation, even when shaving against the grain.
The Mostly Men Corporation © 2011. All rights reserved.

(a) Male targeted advertisement


Bimbo's Burgers
Great food, great value!
Our pries can't be beat, new deals in store now!
Kids eat for free every Tuesday!
Bimbo's Burgers is 100% Irish owned and located in the Ilac Centre
Opening 7 days a week from 9am - 12pm!
Ask about our new meal deals!

(b) Generic/default advertisement


SHOE™
The leading supplier of women's shoes
We have
• Work Shoes
• Casual Shoes
• Evening Shoes
• Everyday Shoes
We have shoes for every occasion!
Shoe™ is located in The Jervis Shopping Centre on The Ground Floor
Up to 50% Sale now on!

(c) Female targeted advertisement

Figure 6.5: The mock-up advertisements for display.

Chapter 7

Application Evaluation

As part of the evaluation of the Dynamic Advertising Application we will refer to the requirements of the application as set out in Section 5.1. We will attempt to measure how well the application performed against these requirements and provide some commentary on how the application could be improved to overcome any problems that may have been identified.

7.1 Required Components

We provide an evaluation of the Dynamic Advertising application under the requirements of the application. The application was tested on a desktop PC with Intel Xeon CPU (2.67GHz) (4 cores), with 4GB of RAM, running Windows 7 SP1 (32-bit). The webcam that was used was the Microsoft LifeCam Cinema H5D. The following results are those that have been gathered in initial tests of the application. If this project were to be extended, it would be advisable to gather more testing results to allow for more accurate representations of the following observations to be made. Such an analysis would include being able to test the system on a standardised set of videos so that the effect of the changing of certain parameters can be measured more precisely.

It must be able to detect that a person (viewer) of an advertisement is present.

This task was performed by the viewer detector component. A pre-trained frontal face classifier (trained using AdaBoost) was utilised to perform this task. The detection performs well in relation to the ability to detect a viewer when they are some distance away. It was found that faces could be recognised as far away as $10m$. The results at this stage were not consistent however, and the distance at which the detector could detect a face consistently was found to be $7m$. This satisfies and greatly exceeds the distance requirement of $> 3m$. A screenshot of the application detected a face at such a distance is shown in Figure 7.1.

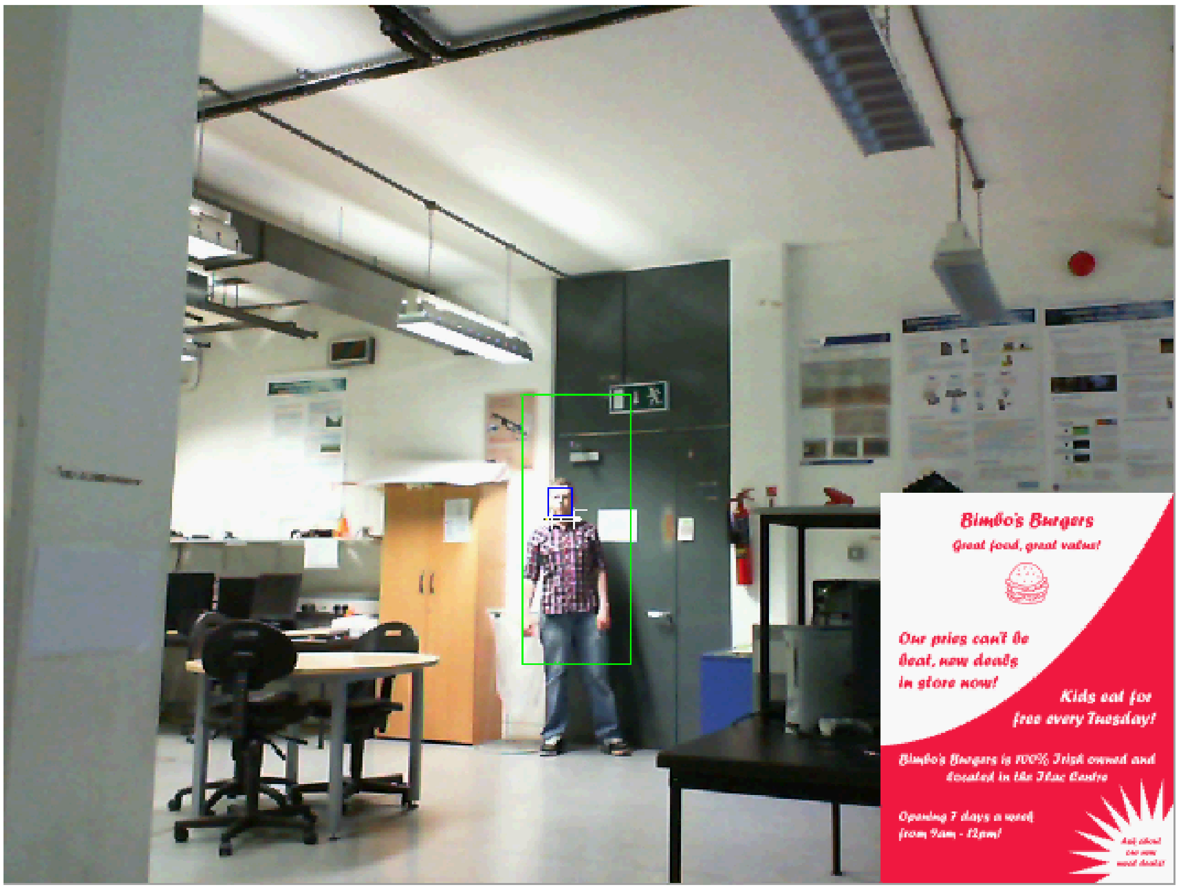

Bimbo's Burgers
Great food, great value!
Our pries can't be beat, new deals in store now!
Kids eat for free every Tuesday!
Bimbo's Burgers is 100% Irish owned and located in the Ilac Centre
Opening 7 days a week from 9am - 12pm!

Figure 7.1: A screen shot showing face detection at a maximum distance. Note the green rectangle denotes the area within which the face detection is being performed.

An aspect of the implementation that helped achieve this performance was that of the background detection component. This allowed for image scaling, and for the detector to be searched at more scales than without the background detection component. It was found that without the background detection being performed, a viewer could be detected consistently at distances of $4m$. It may be worthwhile investigating whether even better results could be achieved by using different background detection methods, such as those listed in [43].

It was found that the viewer could also be consistently detected at distances as close as $0.5m$ to the camera, which exceeds the requirement of $< 1m$.

One assumption that was made was that the definition of a viewer is a person who is directly facing the advertisement. This assumption is reasonable, but it would be desirable if it could be relaxed somewhat. Because, a frontal face detection method was used, the faces that were detected using this module were only ever frontal. Similarly trained classifiers exist for profile faces in the OpenCV distribution and it may be interesting to see if one could use these in addition to the frontal face classifier. There also exists methods in the literature that can perform multi-view face detection [47].

The application runs smoothly for the purposes of face detection (as well as for the other purposes). It has been observed to consistently achieve frame rates > 10 frames per second. Previously, the application had been under-performing with rates of 3-4 frames per second. It was found that after the OpenCV library was updated to the latest version (2.3.1), the performance increase was quite significant. In the release notes for the latest version, Willow-Garage state that the performance of the sparse Lucas-Kanade optical flow has been greatly improved since the previous version. They state that on a 4-core machine, it will run nine times as fast as the previous version. As this component represented a significant bottleneck for the application, once the new version of OpenCV was used, the frame rate increased dramatically. This also allowed for some of the optimisations such as the maximising of the distance of the detection of viewers to be introduced.

The application now runs much more smoothly since the installation of this new version of OpenCV and we will see how much of the other problems encountered are mitigated once the frame rate is increased.

It must be able to determine the gender of such persons.

This requirement related to once a viewer is detected, the system needs to be able to determine their gender. This aspect is reliant on the ability of the face features (eyes and mouth) to be detected accurately. Once these are detected and found to be valid, the face image can be processed for classifying by the gender classifier. It was found that the maximum distance at which these face features could be recognised was $7m$. The features were consistently detectable (when valid) at distances $< 5m$. Some of the performance uplift afforded to the application by the speeded up OpenCV functions allowed for this distance to be maximised to the point where a smooth frame rate was still achievable. A screenshot of the application having determined the gender of a viewer at such a distance is shown in Figure 7.2.

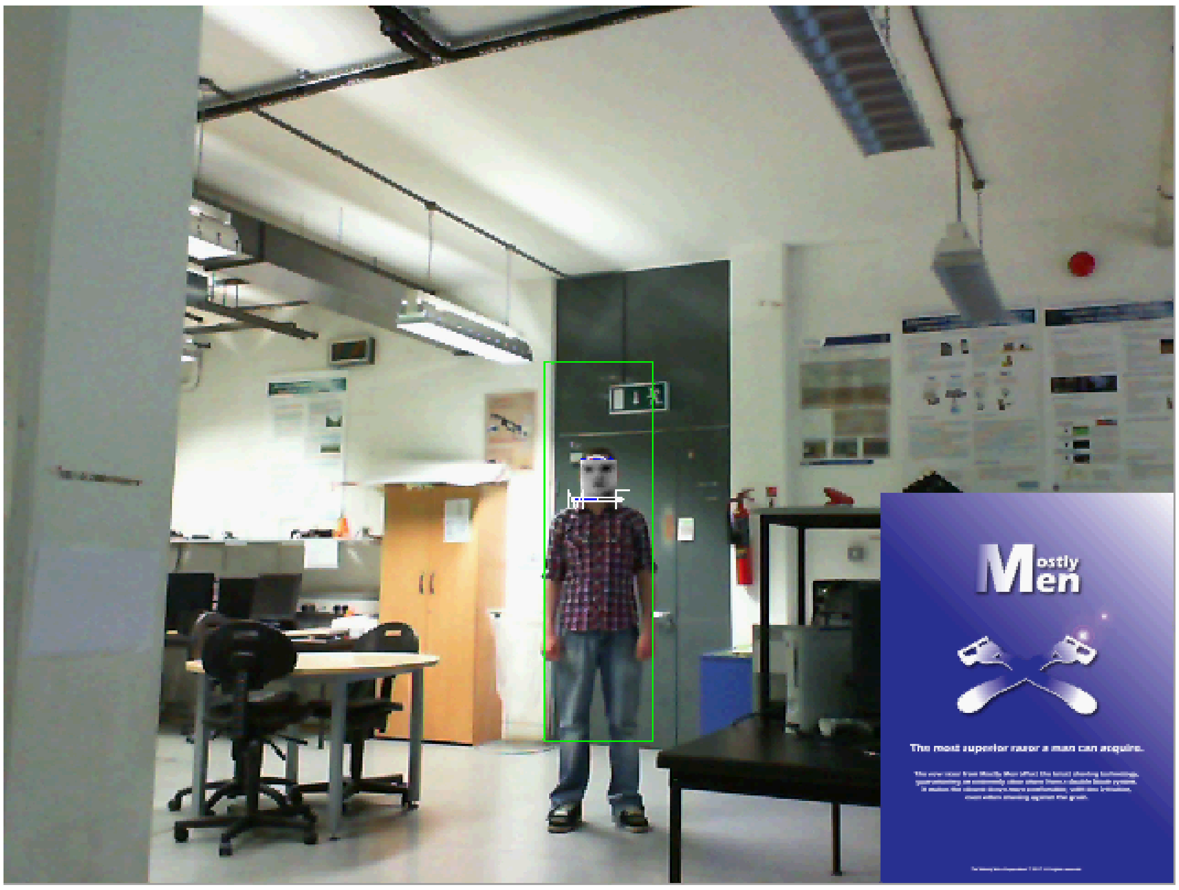


Figure 7.2: A screen shot showing gender classification at a maximum distance.

It was found that the gender classification generally worked immediately once the features were detected. Depending on the number of misclassifications encountered (which were generally low), the viewers gender was determined within $1 - 2s$. It should be noted that the weights afforded to the gender update affect this immensely and these can be tuned to further speed this up. The downside to this is that it may increase the risk of a viewer being misclassified as a particular gender if they are misclassified for a number of consecutive individual frames.

In initial testing the system has yet to be observed completely misclassifying a viewer when the test conditions are followed (viewer is facing forwards towards the display/camera). There have been observations of the gender classifier misclassifying when a viewers face is not properly aligned, i.e. tilted backwards. This is likely due to the affine transformation to the 20×20 thumbnail proceeding when it should really be observed to be invalid in such cases. Potential methods to ensure that these invalid transformations are not performed include adding additional checks to the validity step of the face features (eyes and mouth) detector (Section 6.3) in addition to the three conditions that exist in the current application iteration.

It may be worth investigating this more quantitatively, with numerous test sample videos of subjects to be able to extrapolate any underlying reason for this. This would also allow us to quantify the accuracy of the gender classifier when used as part of the Dynamic Advertising application.

It must be able to dynamically change an advertisement being displayed based upon the classified genders of the viewers of an advertisement.

The application satisfied the requirements under this. The probability model suited the display of the advertisement well and it was possible to adjust its update weights in order to ensure that it was able to display a targeted advertisement within $1s$ of determining a viewer's gender. The trade-off here, is that the more weighted towards the current viewer's gender the update is, the more flickering of changing advertisements will be observed as viewers enter and exit the scene.

It must be extendable to be used with other classification schemes (age, glasses, ethnicity etc.).

For the implementation of the application, we utilised Object-Oriented Design patterns. This allows for the application to be open for extension. The particular classifier used here (gender) could very feasibly be replaced by another classifier for characteristics such as age, ethnicity, etc. The application can also allow for such classifiers to be used in addition to the gender classifier.

7.2 Additional Features

We now evaluate the performance of the implementation of the additional features that were listed in Section 5.1. As above, the results under this section are based upon initial testing of the system. It would be advisable that if this application was to be extended that a more quantitative approach to the testing of these features would be needed.

It can determine the number of viewers of the advertisement.

This was satisfied by the application. The application was able to determine the number of viewers of the advertisement by counting the number of viewers that it tracked. In initial testing this counting seemed to be accurate. Specifically, the application did not suffer from double counting of viewers when a viewer may temporarily turn away from the advertisement. This was achieved through the tracking via the maximisation of mutual information. As mentioned above, the new OpenCV distribution that helped speed up the application, made this counting process more accurate. Previously, when the application was achieving much lower frame rates, a viewer could have moved significantly between the two frames and hence, they may have been classified as a new viewer. With the smoother frame rate, this is much less of an issue as the application can better track the viewer at close to real-time rates.

It can track the viewers of an advertisement as they move past it.

As we have discussed above, the application provides this additional functionality. Under the previous version of OpenCV, whenever the SURF features were tracked across faces, it was necessary to utilise the Lucas Kanade Optical Flow method. Since the performance of this method has been improved dramatically, we have seen that the application can now track viewers at close to real-time rates. A video showing the tracking of users at these rates can be found in Appendix B.

One of the novel methods employed in our tracking methodology, is the use of the Mutual Information to determine matches between viewers' face across frames. This methodology allows for the viewer to turn away from the advertisement, and as long as viewer turns back to the advertisement within a certain time frame, they will continue to be tracked.

There exists many methods for tracking users in video and the method implemented in this application sought to make best use of the information that was already available under the gender classification scheme, such as the thumbnail images and the face regions in the image. It may be worth investigating other methods of tracking viewers in a video. For example, it may be interesting to see if Kernel-Based Object Tracking [48] could perform such a task as well.

It can provide a model for how advertisements should be displayed when there are viewers of different genders simultaneously viewing the advertisement.

This was satisfied as has been described in Section 7.1. The application itself has not been tested with anything > 3 viewers, however if the application is to be extended it may be worthwhile investigating how well the application can handle larger numbers of viewers.

7.3 Remarks

We have discussed how the application has satisfied the requirements that were set out for it. In certain areas such as viewer detector distance, the application has been observed to perform extremely well. If the application was to be extended however, it would be worthwhile performing a fuller quantitative analysis of certain of these aspects. As part of the additional features, the application can perform tracking of a viewer. It may also be worth investigating alternative methods to perform this task to allow for a comparison with the implemented tracking methodology.

Chapter 8

Conclusions and Further Work

8.1 Conclusions

Under this dissertation we wished to investigate methods that would allow for the targeting of display advertisements towards a viewer more effectively. Current display advertising technology is not very intelligent. There exists no analogue to the targeting of, say, online advertising, in the outdoor advertising market. Secondly, the communication of the advertisement is unidirectional. A person may view an advertisement, but the entity placing the advertisement has no automated means of determining if the advertisement is drawing attention or even being viewed at all. Should a system be set-up whereby a camera is placed atop an advertising display, and the advertisement is allowed to be changed based on the processing of the images received through the camera, this may provide a framework to allow for these tasks to be achieved.

Based on such a supposed set-up, there would need to be certain tasks to be performed in the operating on the images received by the camera. One such task is being able to identify the presence of a viewer of an advertisement. Other tasks relate to being able to provide a description of the viewer of the advertisement. Such a description of a viewer, may include their gender, their age, their ethnicity, whether they wear glasses, etc.

In order for us to perform this task, it was necessary to identify previously published methods that have performed such tasks. The task of people detection is an active research problem to this day. Various methods have been proposed to fulfil this task, through methods such as background determination and subtraction, to rapid object classifiers. The task of being able to classify the gender of an individual has also received a lot of attention in the research community. The task is usually more directly put as being able to classify the gender of a subject based on an image of their face. Methods that have been employed, with significant successes, to solve this task include the training of classifiers using the AdaBoost algorithm, with pixel comparisons used as weak classifiers, to combinations of dimension reducing methods such as Principle Component Analysis with feature space classification methods such as Support Vector Machines. These methods, and variations on these methods have also been employed to perform the task of determining the gender of a subject based on a facial image, again with some successes.

For the purposes of our research, it was decided that we would focus on the task of being able to construct an accurate gender classifier, and being able to use such a classifier as part of a wider real-environment application that would be able to display different advertisements dynamically based on the classifications of the viewers of the advertisement.

In order to train a classifier, a methodology similar to that of Baluja and Rowley [18] was followed. The reasons for following this methodology was that in their testing of this AdaBoost-trained classifier compared to other methods, it was found that this classifier was least sensitive to image rotations, scales and translations. The robustness of the classifier to these effects suggested that it may be the best-suited methodology for use in the real-world environment.

Prior to the training of the classifier, it was necessary to perform some pre-processing on the dataset that we intended to use. This dataset was made up of images taken from the FERET image database. The dataset consisted of 990 female samples and 1402 male samples. The preprocessing that was needed to be performed on these images was that they needed to undergo a rigid transformation to a template size of 20×20, and once this was performed the images needed to have their intensities normalised. The correspondences for the rigid transformation were found in the sample images by using pre-trained classifiers (trained using AdaBoost) for face features of eyes and mouth.

Once the data had been prepared, the AdaBoost algorithm was used to train a gender classifier using Haar-like features as weak classifiers. The trained classifiers were tested using the 5-fold cross-validation technique. Of the trained classifiers, it was found that the Gentle AdaBoost variant performed best, achieving 90.8% accuracy under 5-fold cross-validation. This compares well with previous results obtained in the literature, such as 94.4% achieved in [18]. The reasons for the lower accuracy achieved for our classifier may be down to the weak classifiers used in the AdaBoost training schedule. Our method utilised Haar-like features while pixel comparisons were used in [18]. It may be the case that pixel comparisons offer better granularity for picking up some of the differences that may exist between a set of male and female faces. The fully trained classifier (i.e. not cross-validated) achieved an accuracy of 99% which outperforms much of the methods tested similarly (96.6% for Moghaddam and Yang [12]).

Now that a relatively high accuracy classifier had been trained, it was to be used as part of a wider display advertising application. In order to detect the presence of viewers of an advertisement a pre-trained frontal face classifier was used. Faces of viewers were tracked between frames received from a camera input using a bespoke methodology that combined Mahalanobis distance measure, SURF feature Points, Pyramidal Lucas Kanade optical flow estimation and mutual information. For each identified viewer, their face image was processed using the same methods that were used to preprocess the images that were used in the training of the classifier. Once these images were processed, the gender of a viewer could be classified. We modelled the gender of the viewer using probabilities to mitigate against the possibility of any misclassifications. A similar display probability model was used to determine what advertisement to display at particular time (e.g. male targeted, female targeted, generic).

The application performs well in initial testing and achieves the requirements that were set out for it. The application also provides additional functionality in that it can determine the number of viewers that an advertisement has had. Further testing remains to be performed to quantify the accuracy of the classifier in the real-world environment. The application can be extended to include other classification schemes such as age, ethnicity etc.

8.2 Further Work

Further work lies in identifying other potential methods for training a classifier to achieve even greater accuracy. Other classification schemes have emerged that purport to achieve higher accuracy through the use of Scale Invariant Feature Transforms (SIFT) and Gabor transforms at dense grid pixels [23] as weak classifiers in an AdaBoost trained classifier. Another method [21], has a desirable property of capturing discriminative information at different scales and orientations using centralised Gabor Gradient Histogram (CGGH) and a Centre-Based Nearest Neighbour (CNN) measure to perform classification, achieving an accuracy of 96.6%.

We did not make use of any dimension reducing methods in the training of the classifier. Techniques such as Principle Component Analysis have recently been combined with Local Linear Embedding and Support Vector Machines to perform gender classification and achieve accuracies up to 95% [22]. It may be interesting to see if such dimensional reducing methods can be combined with other classification schemes or even perhaps be used as weak classifiers under an AdaBoost training schedule to perform the task of gender classification.

It is worth investigating if an adaptation of such methods or a new classification scheme could be constructed that may be better suited to a real-world environment. It may also warrant investigation to see if such methods could be applied to other classification problems such as age and ethnicity.

Further testing of the Dynamic Advertising application could provide more quantitative results of how well the classifier performs on real-world data. Certain aspects of the application may be altered to achieve greater performance. The application utilises a background detection stage and it may be interesting to see if other background detection methods could increase the performance of the application [43]. The method of detecting the viewer's face in the application is dependent upon the viewer facing towards the camera/screen (i.e. reasonably well-aligned face). It may be worthwhile investigating proposed methods to be able to determine the presence of viewers that may not have well-aligned faces [47].

The methodology for the tracking of a viewer was implemented to best make use of the information that was already available. It may be interesting to see if the tracking could be better performed using other tracking methodologies such as kernel-based object tracking [48].

Finally, we would intend to investigate methods of making an advertising application that is interactive. The application could respond to stimuli such as the presence and locations of viewers to make an advertisement seem more appealing.

Appendix A

Information Theory

Information theory is a branch of science which deals with mathematical details of how information is received and transmitted. It was developed in order to answer two fundamental questions regarding the communication of information. The first of these questions: "What is the ultimate data compression?" yielded the answer of the entropy H. The second question:"What is the ultimate transmission rate of communication?" yielded the answer the channel capacity C. It is the former question that is of concern here. Information theory has numerous applications and, conversely, contributions relating to a very diverse range of academic fields. These include electrical engineering, computer science, statistics, physics and mathematics.

The fundamental concepts of information theory are generally attributed to Claude Shannon, an American electrical engineer and mathematician. In his 1948 classic work "A Mathematical Theory of Information" [49], he introduces the concepts of information entropy and channel capacity. In this paper he addresses the problems stated above, particularly "What is the ultimate data compression?" by introducing the concept of the entropy of an information source.

Much greater detail about the theories forthcoming can be found in [50], where much of the mathematical definitions presented here can be found.

A.1 Shannon's Entropy

Let X be a discrete random variable, i.e. a mathematical description of a random process that takes values from discrete set. It can take any value from the set $\mathcal{X}$, a discrete set, also called X's alphabet.

During a trial, X takes a value $x \in \mathcal{X}$. Associated with each outcome is a probability $p_X(x)$. So in a trial, X has probability $p_X(x)$ of taking value x.

Note that for the probability mass function, summing over all possible outcomes yields unity.

$$\sum_{x \in \mathcal{X}} p_X(x) = 1 \tag{A.1}$$

Onwards, we may denote the probability mass function by $p(x)$ rather than $p_X(x)$ for convenience. Also note that $p(x)$ and $p(y)$ refer to two different random variables and are in fact two different probability mass functions, $p_X(x)$ and $p_Y(y)$, respectively.

Definition The *entropy* $H(X)$ of a discrete random variable X is defined by

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x) \tag{A.2}$$

We can also write $H(p)$ for the above. This alternative notation is used because entropy does not depend on the object in $\mathcal{X}$, but rather on the probability distribution, $p(X)$, i.e. entropy is a functional of the distribution of X. The log is to the base 2 and entropy is expressed in bits. For a succinct example we can use that of a fair coin toss. Assuming $\mathcal{X}$ has two elements, heads and tails, each of which has a probability of occurring of 0.5. Then the entropy of X, a fair coin toss, is 1 bit. It is convention to use $0 \log 0 = 0$, which is verified by continuity since $x \log x \to 0$ as $x \to 0$. Thus terms of zero probability will have no affect upon the entropy of a discrete random variables.

Definition The *expectation value* E, of a function $f(x)$ is given by

$$Ef(X) = \sum_{x \in \mathcal{X}} f(x)p(x) \tag{A.3}$$

Thus we can see that the entropy of X can also be interpreted as the expected value of the random variable $\log \frac{1}{p(X)}$, where X is drawn according to the probability distribution

$p(x)$. Thus

$$H(X) = E \log \frac{1}{p(X)} \tag{A.4}$$

We shall now proceed to prove various properties of entropy.

Lemma A.1.1

$$H(X) \geq 0$$

Proof

$$0 \leq p(x) \leq 0 \;\Rightarrow\; log\frac{1}{p(x)} \geq 0 \qquad ▮$$

Lemma A.1.2 *Say*

$$H_b(X) = -\sum_{x \in \mathcal{X}} p(x) \log_b p(x)$$

Then

$$H_b(X) = \log_b 2H(X)$$

Proof

$$\log_b p = \log_b 2 \log_2 p$$

by the law of logarithms. ▮

This property allows the change of base in the definition of entropy.

A.2 Joint Entropy and Conditional Entropy

In this section, the concepts of joint entropy and conditional entropy shall be introduced. Firstly, we must make some definitions.

Definition Consider X and Y, two random variables with X taking values in $\mathcal{X}$ and Y in $\mathcal{Y}$. The *joint probability* is

$$p_{X,Y}(x, y) = \text{probability that } X = x \text{ and } Y = y$$

Using this definition, we can now define the joint entropy.

Definition The *joint entropy*, $H(X,Y)$ of a pair of discrete random variables (X,Y), with a joint probability $p(x,y)$, is defined as

$$H(X,Y) = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p_{X,Y}(x,y)\log p_{X,Y}(x,y) \tag{A.5}$$

We must also define probability distributions for the marginal and conditional cases.

Definition From the definition of the joint probability we can define the *marginal distributions*

$$\begin{aligned} p_X(x) &= \text{probability that } X = x \text{ irrespective of what } Y \text{ is} \\ p_Y(y) &= \text{probability that } Y = y \text{ irrespective of what } X \text{ is} \end{aligned}$$

and, it follows that

$$\begin{aligned} p_X(x) &= \sum_{y\in\mathcal{Y}} p_{X,Y}(x,y) \\ p_Y(y) &= \sum_{x\in\mathcal{X}} p_{X,Y}(x,y) \end{aligned} \tag{A.6}$$

We must now also define the conditional probabilities.

Definition We define the *conditional probabilities* as

$$\begin{aligned} p_{X|Y}(x|y) &= \text{probability that } X = x \text{ if } Y = y \\ p_{Y|X}(y|x) &= \text{probability that } Y = y \text{ if } X = x \end{aligned}$$

We now use a property of probabilities known as Bayes' rule. Bayes' rule states that the probability of $X = x$ and $Y = y$ is the probability of $X = x$ multiplied by the probability of $Y = y$ given that $X = x$:

$$p_{X,Y}(x,y) = p_X(x)p_{Y|X}(y|x) \tag{A.7}$$

and, similarly

$$p_{X,Y}(x,y) = p_Y(y)p_{X|Y}(x|y) \tag{A.8}$$

These equations can be re-expressed as

$$\begin{aligned} p_{Y|X}(y|x) &= \frac{p_{X,Y}(x,y)}{p_X(x)} \\ p_{X|Y}(x|y) &= \frac{p_{X,Y}(x,y)}{p_Y(y)} \end{aligned} \tag{A.9}$$

Since $p_{X,Y}(x,y) = p_{Y,X}(y,x)$, we can write Bayes' rule in it's general form,

$$p_{Y|X}(y|x) = \frac{p_Y(y)p_{X|Y}(x|y)}{p_X(x)} \tag{A.10}$$

This leads us to our next definition.

Definition If we have a joint distribution $p_{X,Y}(x,y)$, then for a given value of X, $X = x$, say, the conditional distribution can be used to give an entropy. The *conditional entropy* is the average of this entropy over all values of X:

$$H(Y|X) = \sum_{x\in\mathcal{X}} p_X(x) \sum_{y\in\mathcal{Y}} p_{Y|X}(y|x) \log p_{Y|X}(y|x) \tag{A.11}$$

and, by Bayes' rule,

$$H(Y|X) = \sum_{(y,x)\in\mathcal{Y}\times\mathcal{X}} p_{Y,X}(y,x) \log p_{Y|X}(y|x) \tag{A.12}$$

We shall now use these definitions to prove a useful property of the joint entropy.

Theorem A.2.1 *(Chain Rule)*

$$H(X,Y) = H(X) + H(Y|X) \tag{A.13}$$

Proof

$$H(X,Y) = -\sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} p_{X,Y}(x,y) \log p_{X,Y}(x,y) \tag{A.14}$$

By Bayes' rule

$$H(X,Y) = -\sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} p_{X,Y}(x,y) \log p_X(x)p_{Y|X}(y|x) \tag{A.15}$$

$$H(X,Y) = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p_{X,Y}(x,y)\log p_X(x) - \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p_{X,Y}(x,y)\log p_{Y|X}(y|x) \quad (A.16)$$

From the definition of marginal distributions, we get

$$H(X,Y) = -\sum_{x\in\mathcal{X}} p_X(x)\log p_X(x) - \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p_{X,Y}(x,y)\log p_{Y|X}(y|x) \quad (A.17)$$

Now, from the definitions of entropy and conditional entropy,

$$H(X,Y) = H(X) + H(Y|X) \quad (A.18)$$

as required. ▮

A.3 Relative Entropy

As has been discussed, the entropy of a random variable is a measure of the uncertainty of the random variable, i.e. the amount of information required on average to describe the random variable. In this section, the concept of relative entropy shall be introduced.

The relative entropy can be taught of as a measure of the distance between two putative probability distributions. The relative entropy $D(p||q)$ is a measure of the inefficiency of assuming that the probability distribution is q, when the true distribution is p. In this example, knowing the true distribution, p, of the random variable allows us to construct a code with average description length $H(p)$. However, should we wish to use the code for a distribution q, we would need $H(p) + D(p||q)$ bits on average to describe the random variable.

Definition The *relative entropy* or *Kullbach-Liebler distance* between two distributions $p_X(x)$ and $q_X(x)$ for some random variable X is

$$D(p_X(x)||q_X(x)) = \sum_{x\in\mathcal{X}} p_X(x)\log\frac{p_X(x)}{q_X(x)} \quad (A.19)$$

In this definition, we use the conventions that $0\log\frac{0}{0} = 0$, $0\log\frac{0}{q} = 0$ and $p\log\frac{p}{0} = 0$. Thus for all $x \in \mathcal{X}$ such that $p_X(x) > 0$ and $q_X(x) = 0$, then $D(p_X(x)||q_X(x)) = \infty$.

We observe that the relative entropy is zero if $p_X(x) = q_X(x)$. However, we note that it is not symmetric in $p_X(x)$ and $q_X(x)$.

A.4 Mutual Information

Mutual information is a quantity that measures the mutual dependence of two variables. Expanding on this, it is a measure of the amount of information that one random variable contains about another random variable. Mutual information can also be stated conversely as being the reduction in the uncertainty of one random variable due to the knowledge of the other.

Definition Consider two random variables X and Y with a joint distribution $p_{X,Y}(x,y)$ and marginal probability distributions $p_X(x)$ and $p_Y(y)$. The *mutual information* $I(X;Y)$, is the relative entropy between the joint distribution and the multiple of the marginal distributions.

$$I(X;Y) = \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p_{X,Y}(x,y)\log\frac{p_{X,Y}(x,y)}{p_X(x)p_Y(y)} \tag{A.20}$$

Using relative entropy, this can be expressed as

$$I(X;Y) = D(p_{X,Y}(x,y)||p_X(x)p_Y(y)) \tag{A.21}$$

We shall exploit the relations between mutual information and entropy in the next section to prove some useful identities.

A.5 Relationship Between Entropy and Mutual Information

We shall now prove the key result for this section. The proof is due to [50] and will allows the swift calculation of mutual information between two images.

Theorem A.5.1 *(Mutual Information and entropy)*

$$I(X;Y) = H(X) - H(X|Y) \tag{A.22}$$

$$I(X;Y) = H(Y) - H(Y|X) \tag{A.23}$$

$$I(X;Y) = H(X) + H(Y) - H(X,Y) \tag{A.24}$$

$$I(X;Y) = I(Y;X) \tag{A.25}$$

$$I(X;X) = H(X) \tag{A.26}$$

Proof From the previous section, the definition of mutual information is

$$I(X;Y) = \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p_{X,Y}(x,y)\log\frac{p_{X,Y}(x,y)}{p_X(x)p_Y(y)} \tag{A.27}$$

By Equation A.9, we have

$$I(X;Y) = \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p_{X,Y}(x,y)\log\frac{p_{X|Y}(x|y)}{p_X(x)} \tag{A.28}$$

Now, expanding the log gives

$$I(X;Y) = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p_{X,Y}(x,y)\log p_X(x) + \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p_{X,Y}(x,y)\log p_{X|Y}(x|y) \tag{A.29}$$

Using Equation A.6, we get

$$I(X;Y) = -\sum_{x\in\mathcal{X}} p_X(x)\log p_X(x) - \left(-\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p_{X,Y}(x,y)\log p_{X|Y}(x|y)\right) \tag{A.30}$$

From our definitions of entropy A.2 and conditional entropy A.12 we have

$$I(X;Y) = H(X) - H(X|Y) \tag{A.31}$$

Thus the mutual information $I(X;Y)$ can be defined as the reduction in the uncertainty of X due to the knowledge of Y.

By symmetry, it follows that

$$I(X;Y) = H(Y) - H(Y|X) \tag{A.32}$$

This means that X and Y are mutually informative, i.e. X says as much about Y as

Y says about X.

Since $H(X,Y) = H(X) + H(Y|X)$, as shown by Theorem A.2.1, we have

$$I(X;Y) = H(X) + H(Y) - H(X,Y) \tag{A.33}$$

Finally, we observe that

$$I(X;X) = H(X) - H(X|X) = H(X) \tag{A.34}$$

Thus, the entropy of a random variable is equivalent to the mutual information of a random variable with itself. ∎

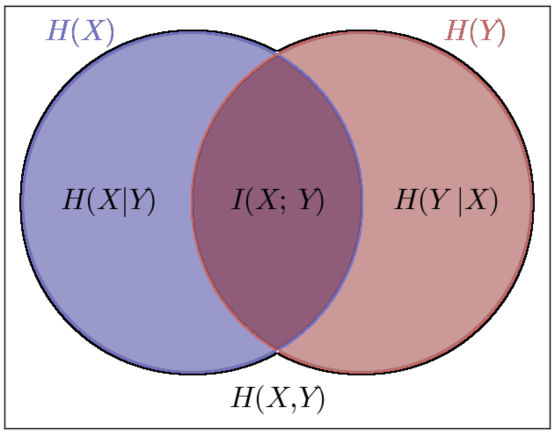


Figure A.1: Relationship between entropy and mutual information.

Figure A.1, represents the relationship between $H(X)$, $H(Y)$, $H(X,Y)$, $H(X|Y)$, $H(Y|X)$ and $I(X;Y)$ expressed in a Venn diagram. $I(X;Y)$ corresponds to the intersection of the information in$X$ and the information in Y.

A.6 Mutual Information as a Metric

Mutual information is deemed to have certain qualities that would be useful in a metric. It does not however qualify as a metric as it does not satisfy the triangle inequality.

According to [51], a metric $d(x, y)$ should satisfy the following conditions:

1. Nonnegativity
$$d(x, y) \geq 0 \tag{A.35}$$

2. Identity of indiscernibles
$$d(x, y) = 0 \Leftrightarrow x = y \tag{A.36}$$

3. Symmetry
$$d(x, y) = d(y, x) \tag{A.37}$$

4. Triangle inequality
$$d(x, z) = 0 \Leftarrow x = y \tag{A.38}$$

Another desirable property for a metric is that it is normalised. One mutual information normalisation is defined as

$$NI(X, Y) = \frac{I(X; Y)}{max(H(X), H(Y))} \tag{A.39}$$

This quantity is symmetrical, satisfies positive definiteness (nonnegativity and identity of indiscernibles) as well as the triangle inequality, therefore it is a metric.

Appendix B

Resources

Please see attached DVD-R for the following resources:

1. Source code, included the trained gender classifier, and compilation instructions for the Dynamic Advertising Application.

2. Video of the Dynamic Advertising Application in use, illustrating its features.

Bibliography

[1] J. Plunkett, "Plunkett's Advertising and Branding Industry Almanac 2010: Advertising and Branding Industry Market Research," *Statistics, Trends and Leading Companies.*

[2] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: real-time tracking of the human body," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 780–785, July 1997.

[3] I. Haritaoglu, D. Harwood, and L. S. Davis, "Hydra: Multiple people detection and tracking using silhouettes," *IEEE Workshop on Visual Surveillance*, p. 6, 1999.

[4] C. Papageorgiou and T. Poggio, "A trainable system for object detection," *International Journal of Computer Vision*, vol. 38, pp. 15–33, 2000.

[5] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'01)*, vol. 1, p. 511, 2001.

[6] Y. Freund and R. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," in *Computational learning theory*, pp. 23–37, Springer, 1995.

[7] M. Ekinci and E. Gedikli, "Background estimation based people detection and tracking for video surveillance," in *Computer and Information Sciences (ISCIS'03)*, vol. 2869 of *Lecture Notes in Computer Science*, pp. 421–429, Springer Berlin / Heidelberg, 2003.

[8] J. Zhou and J. Hoang, "Real time robust human detection and tracking system," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW'05)*, p. 149, June 2005.

[9] V. Fernandez-Carbajales, M. Garcia, and J. Martinez, "Robust people detection by fusion of evidence from multiple methods," in *International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS'08)*, pp. 55–58, May 2008.

[10] R. Brunelli and T. Poggio, "Hyperbf networks for gender classification," in *Proceedings of the DARPA Image Understanding Workshop*, vol. 314, San Diego: CA, 1992.

[11] M. Lyons, J. Budynek, and S. Akamatsu, "Automatic classification of single facial images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, pp. 1357–1362, Dec. 1999.

[12] B. Moghaddam and M. Yang, "Learning gender with support faces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 707–711, 2002.

[13] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, "The FERET evaluation methodology for face-recognition algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 1090–1104, 2000.

[14] G. Shakhnarovich, P. A. Viola, and B. Moghaddam, "A unified learning framework for real time face detection and classification," in *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, FGR '02, (Washington, DC, USA), p. 16, IEEE Computer Society, 2002.

[15] B. Wu, H. Ai, and C. Huang, "Lut-based adaboost for gender classification," in *Proceedings of the 4th international conference on Audio- and video-based biometric person authentication*, AVBPA'03, (Berlin, Heidelberg), pp. 104–110, Springer-Verlag, 2003.

[16] M. Castrillón-Santana, J. Lorenzo-Navarro, D. Hernández-Sosa, and Y. Rodríguez-Domínguez, "An Analysis of Facial Description in Static Images and Video Streams," *Pattern Recognition and Image Analysis*, pp. 461–468, 2005.

[17] Z. Yang, M. Li, and H. Ai, "An experimental study on automatic face gender classification," in *International Conference on Pattern Recognition (ICPR'06)*, vol. 3, pp. 1099–1102, 2006.

[18] S. Baluja and H. Rowley, "Boosting sex identification performance," *International Journal of Computer Vision*, vol. 71, no. 1, pp. 111–119, 2007.

[19] A. Jain, J. Huang, and S. Fang, "Gender identification using frontal facial images," in *IEEE International Conference on Multimedia and Expo (ICME'05)*, p. 4 pp., July 2005.

[20] H. Lin, H. Lu, and L. Zhang, "A new automatic recognition system of gender, age and ethnicity," in *World Congress on Intelligent Control and Automation (WCICA'06)*, vol. 2, pp. 9988–9991, 2006.

[21] X. Fu, G. Dai, C. Wang, and L. Zhang, "Centralized gabor gradient histogram for facial gender recognition," in *International Conference on Natural Computation (ICNC'10)*, vol. 4, pp. 2070–2074, Aug. 2010.

[22] A. Graf and F. Wichmann, "Gender classification of human faces," in *Biologically Motivated Computer Vision*, pp. 1–18, Springer, 2010.

[23] J.-G. Wang, J. Li, C. Y. Lee, and W.-Y. Yau, "Dense sift and gabor descriptors-based face representation with applications to gender recognition," in *International Conference on Control Automation Robotics Vision (ICARCV'10)*, pp. 1860–1864, Dec. 2010.

[24] J. Hayashi, M. Yasumoto, H. Ito, and H. Koshimizu, "Method for estimating and modeling age and gender using facial image processing," in *International Conference on Virtual Systems and Multimedia (VSMM'01)*, pp. 439–448, 2001.

[25] X. Geng, Z.-H. Zhou, and K. Smith-Miles, "Automatic age estimation based on facial aging patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, pp. 2234–2240, Dec. 2007.

[26] G. Guo, Y. Fu, C. Dyer, and T. Huang, "A probabilistic fusion approach to human age prediction," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW'08)*, pp. 1–6, June 2008.

[27] G. Guo, G. Mu, Y. Fu, C. Dyer, and T. Huang, "A study on automatic age estimation using a large database," in *IEEE International Conference on Computer Vision*, pp. 1986–1991, Oct. 2009.

[28] F. Gao and H. Ai, "Face age classification on consumer images with gabor feature and fuzzy lda method," in *Advances in Biometrics*, vol. 5558 of *Lecture Notes in Computer Science*, pp. 132–141, Springer Berlin / Heidelberg, 2009.

[29] "The FG-NET Aging Database." `http://www.fgnet.rsunit.com/`. Accessed September 10, 2011.

[30] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, June 1998.

[31] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[32] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly, 2008.

[33] R. Laganière, *OpenCV 2 Computer Vision Application Programming Cookbook*. Packt Publishing, 2011.

[34] R. Szeliski, "Image alignment and stitching: A tutorial," *Foundations and Trends in Computer Graphics and Vision*, vol. 2, no. 1, pp. 1–104, 2006.

[35] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," in *Proceedings of the Second European Conference on Computational Learning Theory*, (London, UK), pp. 23–37, Springer-Verlag, 1995.

[36] Y. Freund and R. Schapire, "A short introduction to boosting," *Journal of Japanese Society for Artificial Intelligence*, vol. 14, no. 5, pp. 771–780, 1999.

[37] J. Friedman, T. Hastie, and R. Tibshirani, "Additive Logistic Regression: a Statistical View of Boosting," *Annals of Statistics*, vol. 28, 1998.

[38] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Machine Learning*, vol. 37, pp. 297–336, 1999.

[39] "Tutorial: OpenCV haartraining (Rapid Object Detection With A Cascade of Boosted Classifiers Based on Haar-like Features)." `http://note.sonots.com/SciSoftware/haartraining.html`. Accessed September 10, 2011.

[40] "FAQ: OpenCV Haartraining." `http://www.computer-vision-software.com/blog/2009/11/faq-opencv-haartraining/`. Accessed September 10, 2011.

[41] "Qt." `http://qt.nokia.com/products/`. Accessed September 10, 2011.

[42] "qt-opencv-multithreaded: A multithreaded OpenCV application using the Qt framework." `http://code.google.com/p/qt-opencv-multithreaded/`. Accessed September 10, 2011.

[43] M. Piccardi, "Background subtraction techniques: a review," in *IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, pp. 3099–3104, 2004.

[44] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Computer Vision ECCV 2006*, vol. 3951 of *Lecture Notes in Computer Science*, pp. 404–417, Springer Berlin / Heidelberg, 2006.

[45] J. yves Bouguet, "Pyramidal implementation of the lucas kanade feature tracker," *Intel Corporation, Microprocessor Research Labs*, 2000.

[46] A. Dame and E. Marchand, "Accurate real-time tracking using mutual information," in *Mixed and Augmented Reality (ISMAR), 2010 9th IEEE International Symposium on*, pp. 47–56, Oct. 2010.

[47] C. Huang, H. Al, B. Wu, and S. Lao, "Boosting nested cascade detector for multi-view face detection," in *Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04)*, vol. 2, pp. 415–418, Aug. 2004.

[48] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 564–577, May 2003.

[49] C. E. Shannon, "A mathematical theory of communication," *The Bell system technical journal*, vol. 27, pp. 379–423, July 1948.

[50] T. M. Cover and J. A. Thomas, *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing).* Wiley-Interscience, 2006.

[51] F. Escolano, P. Suau, and B. Bonev, *Information Theory in Computer Vision and Pattern Recognition.* Springer London, 2009.