FACE RECOGNITION FOR AUTOMATIC CLASS ATTENDANCE SYSTEM

a project report submitted by

PABLO DANIEL AGUADO (IS15EC001)

as a result of his work done as an IAESTE intern

under the supervision of

DR. D. ABRAHAM CHANDY

Universidad Nacional
de San Juan

Karunya
UNIVERSITY

DECEMBER 2015

DECLARATION

I, **PABLO DANIEL AGUADO** hereby declare that this project report, entitled **"Face recognition for automatic class attendance system"**, submitted to Karunya University, is a record of original and independent work done by me during the period 09/2015-11/2015, under the Supervision and guidance of Dr. D. Abraham Chandy, Associate Professor, School of Electrical Sciences, Karunya University. The work contained in this report has not been previously submitted to meet the requirements for a degree or diploma at this or any other higher education institution.

Pablo Daniel Aguado

BONAFIDE CERTIFICATE

Certified that this project report titled **"Face recognition for automatic class attendance system"** is the bonafide work of **Pablo Daniel Aguado**, who carried out the work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other scholar.

Dr. D. Abraham Chandy

SUPERVISOR
Associate Professor
Department of ECE
Karunya University

Countersigned by

Dr. S. Thomas George

Head of Department
Department of ECE
Karunya University

Contents

DECLARATION i

BONAFIDE CERTIFICATE ii

Introduction 1

  Brief Description 1

  Objectives 1

The problem 2

  Overview 2

    Background – Why it is important 2

    Applications 2

  Literature review 2

  Initial constraints 3

Implementation 4

  Proposed System 4

    Overview 4

    Design Decisions 5

    To be defined 6

  Software framework 7

    To be completed 8

    Classes 8

  Globally rotation invariant multi-scale co-occurrence local binary pattern (MCLBP) 10

Results 11

  Databases used 11

  Main program results 11

  Input program results 12

  MCLBP/C 13

Summary 17

Conclusions and future work 18

References 19

Appendix 1: Log 21

Introduction

Brief Description

This report describes the work done by Mr. Pablo Daniel Aguado (IS15EC001), under the guidance of Dr. D. Abraham Chandy, as an IAESTE intern at Karunya University (Offer N°: IN-2015-1110KU). The project lasted from 01/09/2015 to 01/12/2015, and it is expected that it becomes useful for Karunya University.

The project was named “Face recognition based automatic class attendance system”, and an overview of the problem it tried to solve is outlined in the second part, named “The problem”. Some previous work is also discussed in this section. The “Implementation” title deals with the proposed system and the parts that could be effectively done and might be useful for future development. The results of all the tests are presented and analysed in the next part. Finally, some conclusions about the work are considered.

Objectives

Our goal was to develop a system that is able to correctly identify all the students attending a class, without their explicit cooperation and without the teacher's intervention. It should replace the present manual attendance marking system.

Taking attendance is a time-consuming task that can be automated with the help of technology. At Karunya University classes normally have more than 50 students, so a system like the one proposed will surely save a lot of valuable time for education. For example, a teacher has to use the last 15 minutes of class time for taking attendance in a 160 student class.

It was also expected that the results of the project may be expressed in publications, databases and software modules for further research.

The problem

Overview

Background – Why it is important

Taking attendance is a common task performed in various situations, but it is particularly important at working environments and at school and university classes, courses, lectures and seminars. In school and university classes it has a pedagogical meaning, as it tries to increase the involvement of students in the learning process.

Traditionally, there are two methods for taking attendance in education settings. One is unsupervised and consists of students signing a sheet; this is easy but can be compromised without problems. The other is more trustworthy and implies the teacher calling each of the students in a list. However, this implicates additional effort from the teacher, who must make sure to correctly mark attending students, and at the same time wastes a considerable amount of time from the teaching process. Furthermore, it can be much more complicated if one has to deal with large groups of students.

Some modern attendance management systems are RFID card readers, barcode readers, and other involving biometric features as fingerprints, retina, iris and face. The ones that compare biometric features are the hardest to bypass, but may require an important initial investment. Despite all of these can work very well in most job environments or small classes, big groups of people may result in large queues and need the same time than the traditional methods. It is more practical to capture biometric features remotely without the need of each person having direct contact with any equipment. This reduces the possibility of damage to the equipment as well as it makes it more hygienic. Face recognition systems uses the unique features of a human's face in order to identify them in a group. These features can be captured remotely using a camera places in a safe location. The process begins with face detection, where faces from an image or video are detected and passed to the recognition stage, where the detected faces are identified by comparison with an existing database of faces.

Applications

This kind of system can also be very useful in other applications with similar configurations. To name a few:

- Assemblies of any kind, like the ones carried out daily at Karunya University
- In-company meetings
- Conferences, talks at congresses

Literature review

More than 35 journal papers or conference papers regarding face recognition based systems were gathered and reviewed. We think that the general low quality of them (ignoring all the plagiarized works) is related to the nature of the goal: an attendance system is merely another application of a general face recognition framework; the part that can be different in each case is how the overall attendance management is implemented. Another possible cause is that most schools or universities do not take attendance and/or classrooms are small enough so that taking attendance is not a problem. Despite of being published in journals, most papers are of poor quality, as they do not state well the conditions of the experiments and most of the times they do not even write numerical results. Only one of them (Mao et al. 2014) uses published image sets for testing (although they are composed of frontal single-face images). This way, we are not able to do a good comparison between our system and the others, at least on the correct recognition rates.

Some other things found:

- **Methods**: Most papers use Viola-Jones for detection and Eigenfaces (Principal Component Analysis, PCA) for recognition. These are good but not state-of-the-art techniques (especially Eigenfaces).

- **Pre-processing**: Many papers use resizing and simple contrast enhancement after automatic face detection.
- **Hardware**: In 2 of the papers (Priya et al. 2014; Reddy et al. 2015), the authors say that their implementation is in an embedded system or micro-computer. It seems that one Raspberry Pi is enough for a simple recognition system. One of the others uses server-side processing of the images, while the rest do everything in the same computer. There are no details about the cameras used.
- **Blink detection**: Blink detection is used in two papers (Behara & Raghunadh 2013; Chintalapati & Raghunadh 2013) for checking that the student is real. Sensing this or other micro-expressions are very useful anti-spoofing techniques, but they require a higher resolution than the one needed for recognition only.
- **Image / Video**: Most of the systems use a single image for detection and recognition. Other (Chin et al. 2014) is continuously recognizing, and puts a “present” if the recognitions for a certain person are above a pre-set threshold. Other (Mao et al. 2014) uses sets of images of the same person partially recognized (tracklets) for doing a better final recognition.
- **Position**: Some of the systems have ONE camera at the entrance door, so they get a very good frontal image (assuming no occlusion and small doors). Some others have one camera inside, at the front, desk height or above blackboard. However, the sizes of the classrooms are never as big as Karunya’s. Cameras are either fixed or pan/tilt/zoom cameras controlled by the computer.
- **Feedback**: Most of the systems do not give feedback to the student, so as they can be sure they have been recognized. Just 2 of the (proposed) door systems have: one is visual, the other is oral (the computer announces the name of the student recognized).
- **Unknown**: One of the systems sends the unrecognized faces to manual checking in a GUI. The others just mark them absent.
- **Report**: Most of the papers just show the recognized face in the GUI, and propose a future integration with existing Learning Management Systems, or messaging the teacher or the absent students. Others present a sheet with the attendance for the day.
- **Time, Automation, GUI**: All of the systems have a GUI for testing, and most of them also use it as the normal interface that the teacher has to use for starting the attendance marking procedure. Systems are planned to be full automatic, but that part is not implemented. The GUI can also be used for manual error correction.
- **Databases**: One of the systems (Shehu & Dika 2010) use the image of the students’ ID card, but because of aging it works well only for the first year students. The same system proposes updating the database with all the recognized faces, but this would work for good quality faces, i.e. using a door camera. Most of the other papers use only one image of the students, while some use less than 10 of each. Beards, makeup and veils (Balcoh et al. 2012; Dantcheva et al. 2013) can be a problem.

Initial constraints

Each aspect of the problem and the possible solution was analysed and some options were seen. It was decided to leave many of the questions not answered (in particular those related to the attendance management) and to decide about them later on the process. Then, the focus of the work was on the face recognition system. However, some constraints were set:

- Only one camera will be used, per classroom. They will be placed in a high place that permits seeing all the students. The camera is a single lens, wide angle, high resolution surveillance camera.
- The size and shape of the objective classrooms is not quantitatively known. Hence, the camera minimum requirements cannot be set.
- Students will be facing the camera and will not intentionally occlude their faces.
- The face detection will be Viola-Jones cascade object detection (Viola & Jones 2004), as it is a proven industry standard.

- The face recognition part will be based on Eigenfaces (Turk et al. 1991), as this is an easy and classic approach. Later on it can be changed for improving the system.
- Developing and testing will be in Matlab R2014a.

Implementation

Proposed System

Overview

This is an overview diagram of the proposed system:

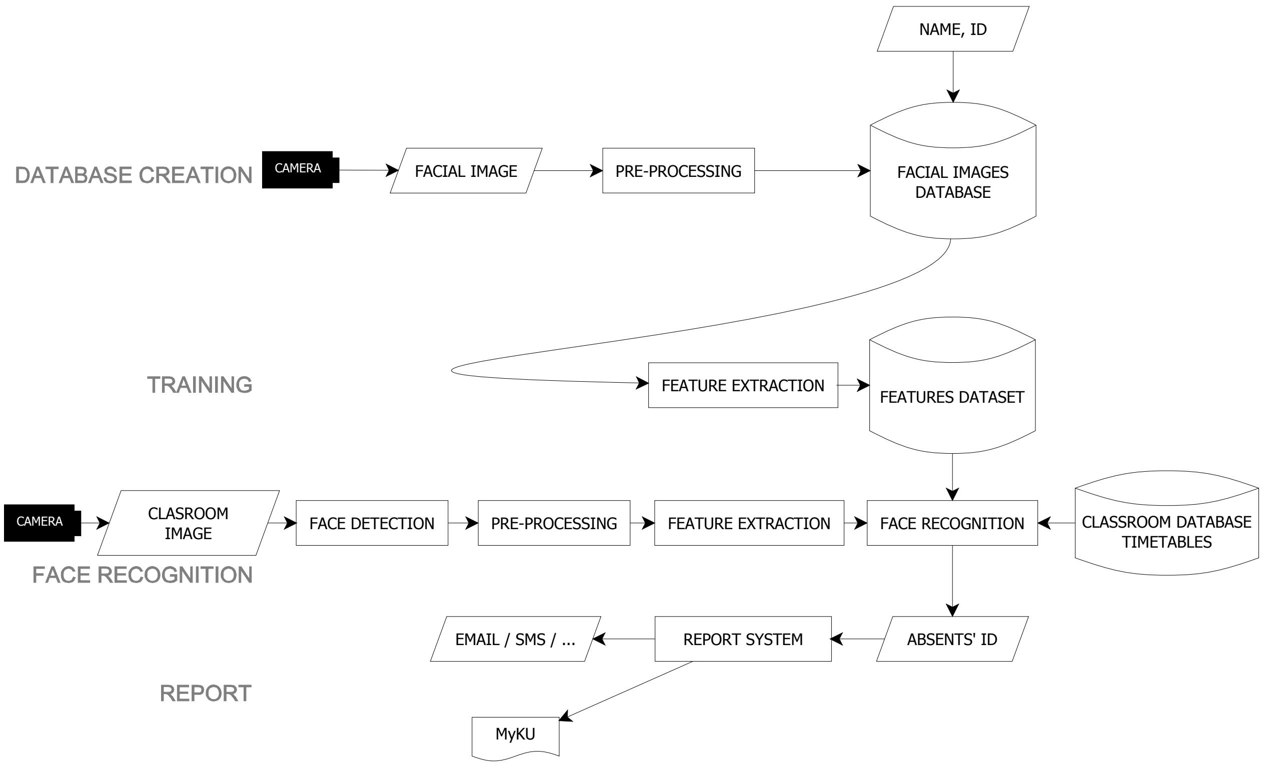

Figure 1: Overview diagram 1

Modules

1. High resolution wide angle camera
2. Pre-processing module
3. Face detection module
4. Feature extraction module
5. Face recognition module
6. Report system
7. Enrolment program
8. Attendance database
    a. Timetables
    b. Students at each class

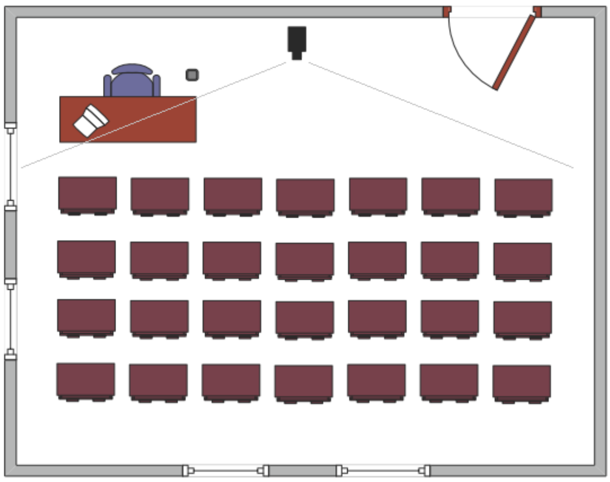

Figure 2: Overview diagram 2

Design Decisions

Object oriented

An object oriented system is proposed, as it allows developing an easy to change modular framework. This is important for testing and upgrading the system with minimal changes, particularly enabling fast changes of the pre-processing, feature extraction and face recognition modules.

Super resolution

Super-resolution is a class of techniques that enhance the resolution of an imaging system, typically by using multiple low resolution images. When applied to faces, it is also called face hallucination, and comprises techniques which take noisy or low-resolution facial images, and convert them into high-resolution images using knowledge about typical facial features. A face recognition based attendance system must have cameras in all classrooms, and then the price (and quality) of the cameras used becomes an important variable. Probably, an average surveillance camera will be used, with little or no magnification at all, and so the quality of the faces obtained will be poor, mainly for the distant subjects.

Hence, super-resolution can be used for enhancing the input image of the face detection module, and face hallucination can be used for enhancing the detected faces. It is very important that all faces in an image are detected, as all the next steps depend on them.

Geometric normalization: Face frontalization by Hassner et al.

The objective of geometric normalization is to get a constant face pose and size, in order to reduce the variance of the face space caused by in-plane rotations, out of plane rotations, occlusions and even expressions – this allows better discrimination. Tal Hassner et al. (Hassner et al. 2014) propose a fast face frontalization algorithm to simplify unconstrained face recognition by reducing it, at least in terms of pose variations, to the simpler, constrained settings. The Matlab implementation by Hassner is available online (Hassner 2015).

Photometric normalization: Tan & Triggs pre-processing

The objective of photometric normalization is to reduce the variations of photometric aspects of the face images before the feature extraction step. This leads to a more robust, illumination invariant system. It was decided to use Xiaoyang Tan and Bill Triggs pre-processing algorithm (Tan & Triggs 2007), as it was seen as one of the best for improving the recognition rates (Han et al. 2013). The implementation by Vitomir Štruc in his PhD toolbox is used (Štruc 2012; Štruc & Pavešic 2010; Štruc & Pavešic 2009).

Database

The database is composed of pictures of all the students of the university. They are ordered in folders identified by the same ID of the students, as these are unique. Each student should have an updated set of photos with different illumination angles and face expressions.

There are available text files for each class. These are identified by a unique name and contain all the students enrolled in that class.

Manual checking

Faces that are detected but that cannot be confidently assigned to an enrolled student are left for manual checking by the teacher. This is achieved by measuring the Euclidean distance between the input feature vector and the mean of all the training feature vectors (belonging to the faces of a student in the database). After matching procedure by the Classification module, if the input vector is farther than a threshold, the matching is considered as not safe enough. The threshold is proportional to the length of the mean training vector.

Classification

Classification is made using the k-Nearest-Neighbours algorithm – this compares the distance between the input face feature vector and the k nearest feature vectors of the training set. Tests were made for different number of neighbours and different distance measures, using the Yale database and the ORL/ATT database. The best results obtained were for Euclidean distance with 5 neighbours.

To be defined

There is a large set of features that were left to be defined after the basic recognition system was working:

Hardware

All the processing can be done in

- the teacher's computer;
- a dedicated computer in the classroom;
- in the cloud / in Karunya's servers;
- in an embedded system built for this purpose.

Note that the used system should be connected to the internet in order to update the attendance.

Automatic or manual start

Attendance taking system can start automatically at the time that the class should start (if the processing is server-side) or the teacher can start it by himself / herself. It should also be defined if the stop is automatic or manual.

Imaging

- One image at start
- One image at end
- One video at start
- One video at end
- Image / video at start and end
- Full class until all students are recognized

Anti-spoofing

A blink detector might not be feasible for very small face images. One option is to do continuous detection and recognition, and marking present to those whose faces were recognized a number of times above a threshold (as proposed in (Chin et al. 2014)). Theoretically, students should not be able to hold a picture of a face during all the needed time, as the teacher could detect it.

Attendance marking

Attendance can be marked:

- if student is present at the start of the class;
- if student is present at the end of the class;
- if student is present at the start and the end;
- if student presence (minutes / detections) is above a threshold.

Feedback

- Nothing
- Light to indicate that the system is detecting / working, or has recognized
- Sound to indicate that the system is detecting / working, or has recognized
- Display to indicate that the system is detecting / working, or has recognized

Graphical User Interface

It should be determined if there are one or more GUIs, and the capabilities they have.

Reports

- Send email with absents
    - Before class finishes
    - After class finishes
- Send SMS with absents
    - Before class finishes
    - After class finishes
- Sheet with absent and present students
- Directly fill in MyKarunya

Database

- First year ID card photo only
- Custom database
    - One photo
    - Multiple photos with different angles and illumination
- Update database
    - With each successful recognition (above a uncertainty threshold)
    - Every year
    - After important changes (shaved, grew beard, moustache, new glasses)

Software framework

The main part of the final (and incomplete) work is able to:

1. Load photos from a database. Each folder corresponds to one subject, and there are multiple photos of each subject. The name of the folder will be the ID of the subject; there are no specifications for the names of the photos.
2. Split the photos into training and test sets. This is for developing only.
3. Pre-process and save all the training images.
4. Create a Class object and load from an external text file all the IDs corresponding to that class.
5. Load the faces for that class, from the pre-processed faces folder, and the feature extractor is trained according to these faces. Please note that it can be also trained with all the subjects and not only the ones belonging to the class in question – in fact, this could be a better approach for most cases. The feature vectors corresponding to the training set are obtained.
6. Create a Classifier object and train it with the training set feature vectors and the corresponding IDs vector.
7. Load the test images corresponding to the students in the class (as if they were previously detected in a classroom image) and pre-process them in the same way than the training images.
8. Classify the pre-processed test images with the previously trained classifier. The closeness is evaluated and each classified test image is tagged as reliable or not.

The input part is able to:

1. Extract sample frames from a video, to be used as input.
2. Detect all the faces in that sequence of frames. All the faces that are in the same place along frames are assigned to the same detected person object. It is supposed that the positions are stationary during that amount of time – the people are not moving. Some or all of the faces could be used for face hallucination before proceeding to the main program.

The code has been delivered to Dr. Abraham Chandy. For further enquiries, the author can be contacted at aguadopd@hotmail.com.

To be completed

At the time of the internship finishing, there are some important parts of the program yet to be completed in order to have a minimum viable product:

Enrolling system

The enrolling module is a very important component of the system. It should take photos of every student, with different illumination angles, pose angles and expressions. At the same time, it should be efficient enough so as to complete the enrolment of the more than 8000 Karunya students in a short time. The conditions should be the same for every student.

Database

The database of pre-processed faces should be designed so as to be easily accessed or copied by each attendance system. Efforts should be made to make it immune to hostile modification attempts.

Skin colour pre-processing

All the face detection systems should also include some skin-colour based filtering after the initial detection, with the aim of removing false positives. This is shown in (Erdem et al. 2011) as better than skin-colour filtering before the detection.

Joining the parts

After completing all the modules, a program joining all the parts should be developed.

Writing an user manual

Classes

Some insight on the classes developed follow. Not all properties or methods are written here. Most classes are derived from the native Handle class. This allows implicit calling of the methods and also is used for making multiple references to the same data, and not copies (important for large arrays of images).

1. **Image**: Basic image class
   a. Properties:
      i. img: a uint8 or double matrix containing the values of the pixels, as read by imread function.
      ii. id: a string with the name of the subject in the image, if known.
   b. Methods:
      i. imshow: overloaded imshow function, for developing purposes.
      ii. reshape: overloaded reshape function. Works for arrays of Images, and it is used in the Eigenfaces training.
2. **TestImage**: Image class used for the test/probe faces.
   a. Properties:
      i. featureVector
      ii. recognized_id: actually a Student object, but should be a string.
      iii. img: Image object.
      iv. probable: Boolean flag indicating if the classification is plausible or not.
   b. Methods:
      i. getFeatures: get the feature vector, using a certain feature extractor object.
3. **FeatureExtractor**: Base class for feature extractor classes.
4. **Eigenfaces**: FeatureExtractor based on Eigenfaces (PCA)
   a. Properties:
      i. eigenfaces: Eigenfaces matrix, used in feature extraction.

ii. meanface: mean face of the training set, used for developing.
iii. p: Percentage of the total variance that wants to be explained with the eigenfaces matrix.
iv. k: Ammount of eigenfaces to be kept, arbitrarily, if p is not used.
v. n: Amount of eigenfaces kept, when using p.

b. Methods:
   i. getFeatures
   ii. train

5. **Classifier**: Classifier based on the knn classifier provided by the Statistics toolbox of Matlab.
   a. Properties:
      i. class: Class object.
      ii. clsf: Matlab's classifier object. In our case, a knn classifier.
      iii. kfoldLoss: the 10 fold loss. This is obtained by splitting the original training set in 10; then one set is used for testing and the other 9 sets are used for training; this is done 10 times, one for each of the 10 sets. The mean loss (proportion of misclassified items) is the kfoldLoss.
      iv. resubLoss: resubstitution loss, proportion of misclassified training faces when they are used again for testing.
      v. testLoss: proportion of misclassified test images when using the Classify2 method.
   b. Methods:
      i. train: train the classifier with the feature vectors and ids vector belonging to the class object.
      ii. Classify2: to classify the test images obtained by the detector.
      iii. checkThreshold: checks the reliability of the classification.
6. **Student**:
   a. Properties:
      i. id
      ii. Photos: set of photos as Image objects.
7. **Class**:
   a. Properties:
      i. name
      ii. list: list (cell) of student names, to be loaded from the Student Database.
      iii. students: array of Student objects.
      iv. featureVectors: array of feature vectors, in same order as students array. Rows are students, columns are features. .
      v. meanFeatureVectors: mean feature vector for each student, used by checkThreshold method of the classifier.
      vi. idsCell: for all the feature vectors.
      vii. ids: for the mean vectors. As list but ordered.
      viii. FE: feature extractor object.
      ix. Clsf: Classifier object
   b. Methods:
      i. train: trains the feature extractor FE, if trainable.
      ii. loadStudents: Loads the Student objects belonging to the Class.
      iii. loadList: Loads the IDs of the students belonging to the class, from a text file.
8. **StudentDatabase**
   a. Properties:
      i. array: array of Student objects.
   b. Methods:
      i. loadPhotos: loads a set of photos from a specified folder.

        ii. splitAndSave: split the photos set in a test set and a training set, for developing purposes.
        iii. preprocessAndSave: pre-process the photos loaded and save them for further use.
9. **InputSequence**: a class containing a video input.
    a. Properties:
        i. frames: cell array of frames from a video input.
        ii. detected_persons: DetectedPerson objects array.
    b. Methods:
        i. process: detect the faces in every frame. All the faces that are in the same place along different frames are assigned to one DetectedPerson object.
10. **DetectedPerson**: Contains all the faces of the same person, detected in each frame.

Globally rotation invariant multi-scale co-occurrence local binary pattern (MCLBP)

During the last two weeks we tried to improve the performance of the feature extractor. For this, Dr. Chandy proposed trying to apply to face recognition a novel LBP based descriptor, the "Globally rotation invariant multi-scale co-occurrence local binary pattern" by Xianbiao Qi et al (Qi et al. 2015). The comparison was intended to be made against the results obtained by Jun Liu et al in their work titled "Variable length dominant Gabor local binary pattern (VLD-GLBP) for face recognition" (Liu et al. 2014). For this, the color FERET database was obtained after the corresponding permission. The grayscale FERET database is the one used for this evaluation, and it is contained in the second DVD of the color FERET.

The steps followed were:

1. All the images were decompressed in Ubuntu using *bunzip2 -f *.tif.bz2*
2. Loaded all the images corresponding to the training gallery set and the *fafb*, *fafc*, *dup1* and *dup2* probe sets.
3. The faces in each image were rotated, cropped and resized according to the eye coordinates provided with the database. No correction to these coordinates was made, even though that very few were found to be erroneous. These sets were saved.
4. The **completed or center gray level MCLPB (MCLBP/C**) features were extracted with the code provided by the original authors. The PCA (Eigenfaces) and simple LBP features were also extracted for comparison. For PCA, the PhD toolbox by Vitomir Štruc was used; for LBP, the code by Marko Heikkilä and Timo Ahonen was used (Heikkilä & Ahonen n.d.).
5. The results were evaluated with the PhD toolbox, obtaining the CMC and ROC curves.

Tests were made for the cropped grayscale images and also for faces with Tan & Triggs pre-processing applied – T&T as implemented by Štruc in his INFace toolbox v2.1 (Štruc & Pavešic 2011; Štruc & Peyre 2012). Despite all the efforts made, the results were worse than the ones specified for PCA (or baseline EigenFaces) in the paper by Liu et al or in the original FERET papers. The *fafc*, *dup1* and *dup2* probes were not tested, as they are more difficult than fafb in terms of expression, illumination and aging. Results are showed under the Results title of this report.

Results

Databases used

When benchmarking an algorithm it is recommendable to use a standard test data set for researchers to be able to directly compare the results. Formal benchmarking was carried out for the MCLBP/C algorithm, using the FERET database. For the main program and input program, databases were used as samples.

ORL/ATT

AT&T "The Database of Faces" (formerly "The ORL Database of Faces"). Ten different images of each of 40 distinct subjects. For some subjects, the images were taken at different times, varying the lighting, facial expressions (open / closed eyes, smiling / not smiling) and facial details (glasses / no glasses). All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position (with tolerance for some side movement). (Samaria & Harter 1994)

Yale

Yale Face Database. Contains 165 grayscale images in GIF format of 15 individuals. There are 11 images per subject, one per different facial expression or configuration: center-light, w/glasses, happy, left-light, w/no glasses, normal, right-light, sad, sleepy, surprised, and wink. (Belhumeur et al. 1997)

FERET

The FERET database was collected in 15 sessions between August 1993 and July 1996. The database contains 1564 sets of images for a total of 14,126 images that includes 1199 individuals and 365 duplicate sets of images. A duplicate set is a second set of images of a person already in the database and was usually taken on a different day. For some individuals, over two years had elapsed between their first and last sittings, with some subjects being photographed multiple times. This time lapse was important because it enabled researchers to study, for the first time, changes in a subject's appearance that occur over a year. **In this case, the grayscale FERET database was used, following the FERET Sep96 evaluation protocol.** (Phillips et al. 2000)

Main program results

Following, some images showing the correctly recognized faces, false positives, false negatives and the probable/not-probable flag.

Figure 3: Correctly recognized face

Figure 4: Incorrectly recognized face

Figure 5: False negative

Some tables were made for comparing different k neighbours, distance measures, amount of eigenfaces used and pre-processing techniques. The following shows the 10 fold mean loss (wrong classified proportion of the total set) for different amount of neighbours in the knn classifier, using the Yale database, 90% of all the Eigenfaces and 9 to 10 training faces for each subject. No photometric normalization is applied.

Table 1: 10 fold mean loss for different k nearest neighbours

Distance	**knn = 1**	**knn = 3**	**knn = 5**	**knn = 7**
Euclidean	0,22857143	0,22857143	0,21428571	0,25
Seuclidean	**0,18571429**	**0,18571429**	0,2	0,19285714
Cosine	0,22142857	0,21428571	0,19285714	0,2

In general, Euclidean or Standardized Euclidean distances produced better results, as well as 5 or less neighbours. Results were better without Tan & Triggs pre-processing. No geometric normalization was applied.

Mean correct prediction rate (1 - loss) was 80% for the Yale database and as high as 94% for the ORL/ATT database.

Input program results

The input program detects all the faces in that sequence of frames. All the faces that are in the same place along frames are assigned to the same detected person object. It is supposed that the positions are stationary during that amount of time – the people are not moving. Some or all of the faces could be used for face hallucination before proceeding to the main program. For this, entropy and information of each image could be analysed so as to use the most similar frames.

Sample video is "St. George's Lower School Chorus - Classroom Musical (One Republic)", taken from YouTube.

Figure 6: Sample frame

Figure 7: Faces extracted belonging to the same person

MCLBP/C

The grayscale FERET database was used for benchmarking the globally rotation invariant multi-scale co-occurrence local binary pattern (completed or center gray level) (MCLBP/C) algorithm, following the FERET Sep96 evaluation protocol.

Tests were made for the cropped grayscale images and also for cropped faces with Tan & Triggs pre-processing applied – all for *fafb* test probe. The *fafc*, *dup1* and *dup2* probes were not tested, as they are more difficult than *fafb* in terms of expression, illumination and aging.

The original MCLBP work by Qi proposes the use of Chi squared distance for comparing feature vectors. Chi squared kernel, used by SVMs, can be defined as:

$$\textbf{Equation 1}: \chi^2 kernel: S(X,Y) = \sum_i \frac{2.X_i.Y_i}{X_i + Y_i}$$

However, in this work we are using KNN. The Chi squared distance between feature vectors X and Y is defined as:

$$\textbf{Equation 2}: \chi^2 distance: D(X,Y) = \sum_i \frac{(X_i - Y_i)^2}{X_i + Y_i}$$

The distances obtained this way were 0 or indefinite even for feature vectors of very different faces, so the Chi squared distance was not used. We used Euclidean distance (L2) or Cityblock distance (L1) instead.

Figure 8 and Figure 9 show ROC and CMC curves for the different feature vectors extracted, with different amount of Eigenfaces kept in PCA (1%, 5% and 50% of all the Eigenfaces, respectively), and using Euclidean distance classifiers (euc) or Cityblock distance classifiers (ctb). Figure 10 shows how the recognition rates increased when Tan & Triggs pre-processing was used (using PCA with 90% of the Eigenfaces, and square, non-stretching cropping).

As it can be observed in Figure 11, Table 3 and Table 4, our PCA results (Table 2) are worse than the ones specified for PCA (or baseline EigenFaces) in (Liu et al. 2014),(Phillips et al. 2000) or (Delac et al. 2005), in spite of having followed relatively the same steps. **We think that some error is being made in the tests**. Moreover, Chi squared distance could not be used, so we cannot conclude that MCLBP/C, as proposed by Qi et al, is better than PCA for face recognition.

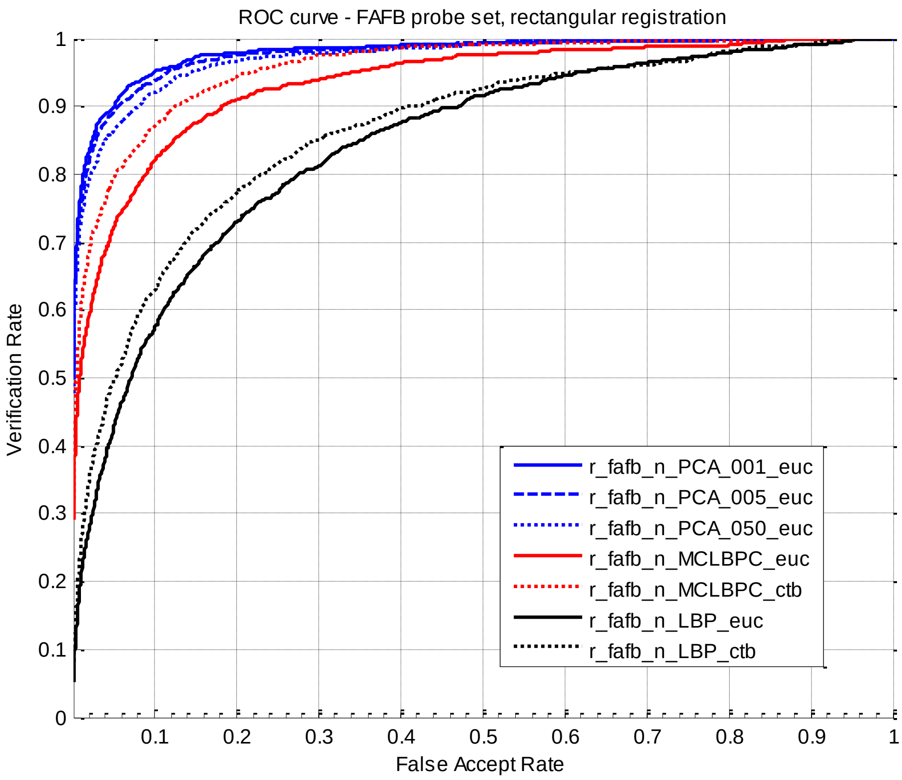

Figure 8: Receiver Operating Characteristic curve for the fafb probe set, using different feature vectors

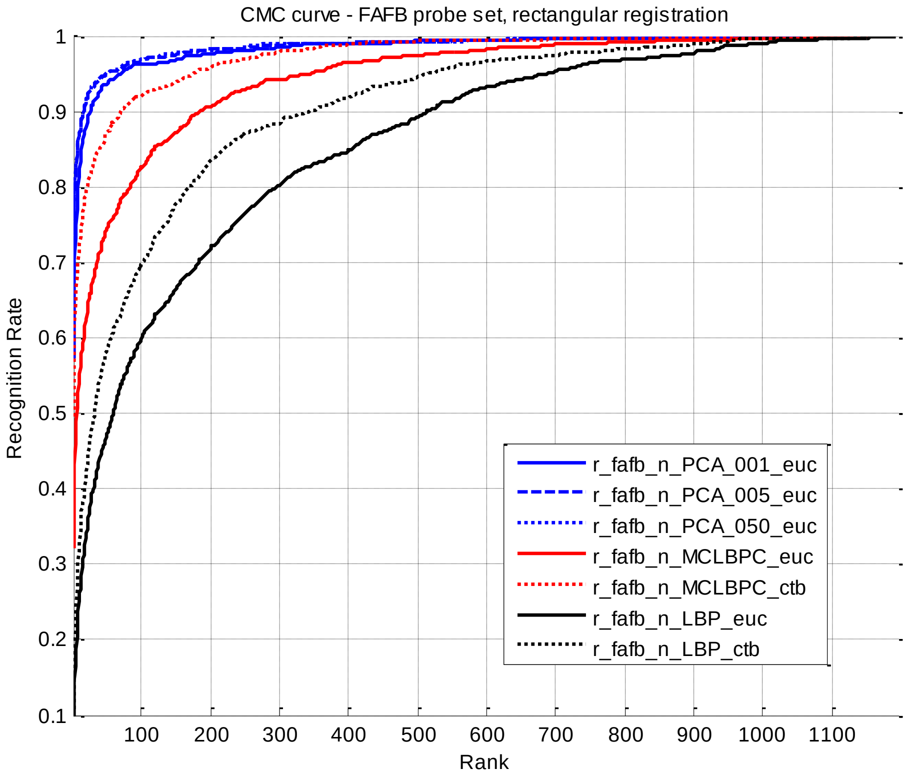

Figure 9: Cumulative Match Curve for the fafb probe set, using different feature vectors

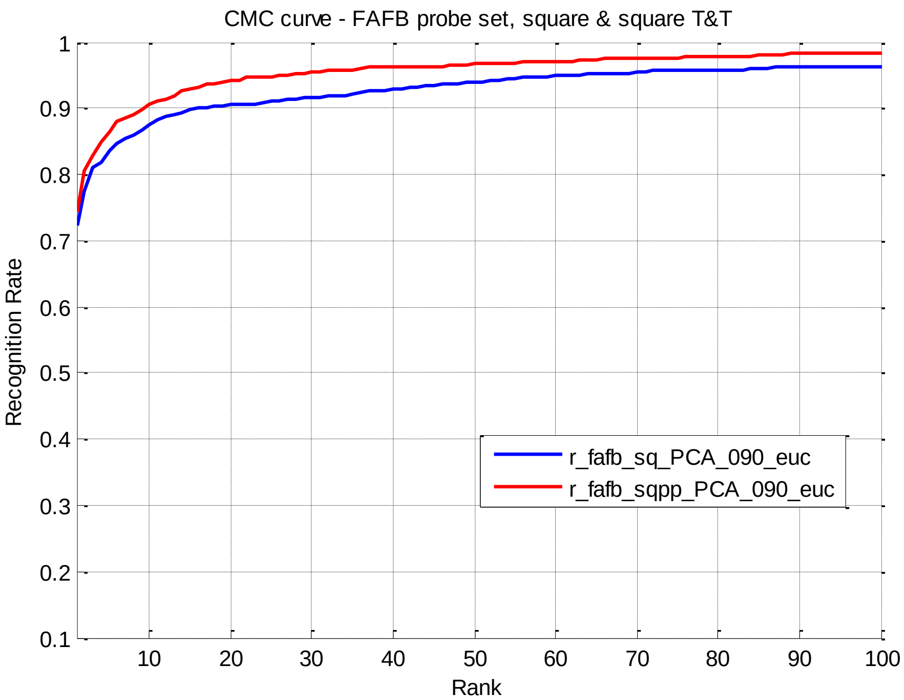

Figure 10: Recognition rates increase with Tan & Triggs pre-processing

Table 2: Rank 1 results obtained by different algorithms implemented in this work for the FAFB probe set

PCA 1%	Euclidean	57
PCA 5%	Euclidean	68
PCA 50%	**Euclidean**	**71**
MCLBP/C	Euclidean	32
MCLBP/C	Cityblock	49
LBP	Euclidean	8
LBP	Cityblock	10

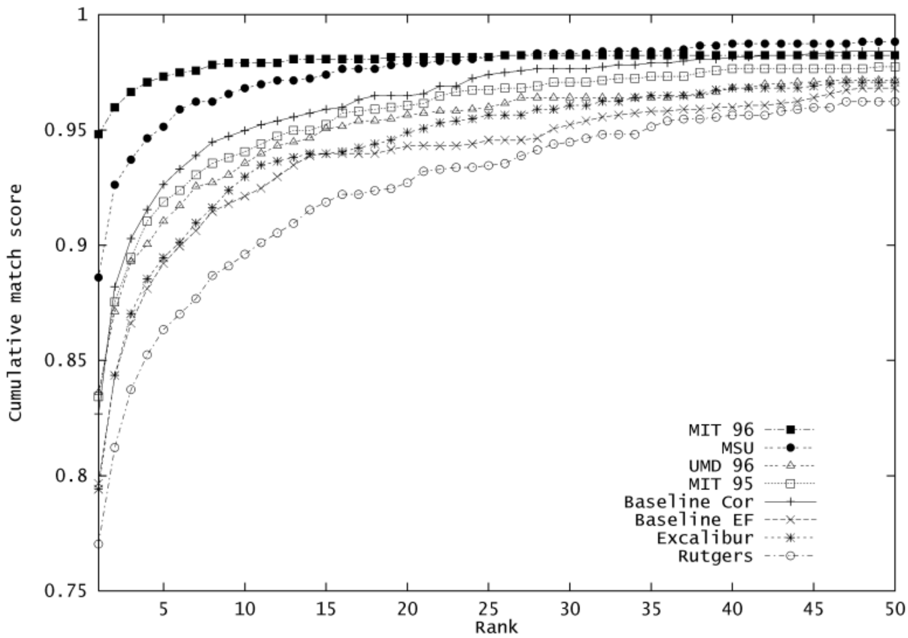

Figure 11: Identification performance in fafb, September 1996 results (Phillips et al. 2000)

Table 3: Rank 1 recognition rates for different algorithms on the FERET database (Liu et al. 2014)

Method	FB	fc	dup. I	dup. II
PCA	0.85	0.65	0.44	0.22
Fisherface	0.94	0.73	0.55	0.31
LBP	0.97	0.79	0.66	0.64
Gabor-M+FLDA	0.9615	0.7629	0.5817	0.3419
LGBP-M [4]	0.98	0.97	0.74	0.71
E-GV-LBP-M [5]	0.9841	0.9897	0.8199	0.8162
VLD-GLBP	**0.9707**	**0.9536**	**0.7230**	**0.7009**

Table 4: Rank 1 recognition rates for different algorithms and metrics on the FERET database (Delac et al. 2005)

	Results at Rank 1 (%)			
	Metric			
Projection	**L1**	**L2**	**MAH**	**COS**
		Fb		
PCA	**82.26**	*82.18*	*64.94*	*81.00*
ICA1	*81.00*	**81.51**	*64.94*	*80.92*
ICA2	*64.94*	*74.31*	*64.94*	**83.85**
LDA	*78.08*	**82.76**	*70.88*	*81.51*
		Fc		
PCA	**55.67**	*25.26*	*32.99*	*18.56*
ICA1	*18.04*	*17.53*	**32.99**	*12.89*
ICA2	*15.98*	*44.85*	*32.99*	**64.95**
LDA	*26.80*	*26.80*	**41.24**	*20.62*
		Dup1		
PCA	**36.29**	*33.52*	*25.62*	*33.52*
ICA1	**32.55**	*31.86*	*25.62*	*32.27*
ICA2	*28.81*	*31.99*	*25.62*	**42.66**
LDA	**34.76**	*32.96*	*27.70*	*33.38*
		Dup2		
PCA	**17.09**	*10.68*	*14.53*	*11.11*
ICA1	*8.97*	*7.69*	**14.53**	*8.97*
ICA2	*16.24*	*19.66*	*14.53*	**28.21**
LDA	*16.24*	*10.26*	**16.67**	*10.68*

Summary

Summarizing what was done:

1. The problem was proposed, and previous work on the matter was read for defining all the things to solve. Decisions were taken about some of the characteristics of the projected solution, and some others were left for later.
2. A main part of the solution was developed. It comprises the feature extraction and subsequent classification. A large testing framework was programmed for this. Samples from ORL/ATT, Yale and grayscale FERET databases were used for this purpose. Correct recognition rates were higher than 80% for multiple face training, without appropriate cropping, geometric normalization or problem-based heuristics.
3. The input part of the program was developed, but with less progress. It is capable of detecting and extracting faces from a sequence of adjacent frames and it groups the faces into persons according to the position in the frames.
4. The performance of the novel MCLBP/C texture descriptor was tested into a face recognition problem, using the grayscale FERET database. The results obtained were not consistent enough so as to extract conclusions from them.

Conclusions and future work

A classroom attendance system based on face recognition, with one frontal camera like the one proposed, is in a constrained environment to a certain extent but has to deal with not so constrained faces, as there can be strong variations in illumination, rotations, expressions and mainly occlusions. For our simple system, a complete, descriptive and updated set of faces for each objective person could be needed. This can be seen in the results for the FERET evaluation, when aged or rotated probes are tested.

However, state-of-the-art systems provide some really promising results regarding unconstrained face recognition. In particular, results of the Labelled Faces in the Wild (LFW) evaluation suggest that this task has already been satisfied (Huang et al. 2007). These novel systems are very complex and each of the parts (normalization, feature extraction, matching) normally involve long previous machine learning with large datasets. As we could see when applying the Tan & Triggs photometric normalization, and when fine-tuning the parameters of the feature extractor and classifier, every step in the process can step-up the recognition rates some points – it may be little but every little is needed.

Regarding the camera, more analysis is necessary. If a normal high resolution, wide angle surveillance camera is not enough even with super resolution techniques (which require more processing and developing), it might be needed to go for a zoom-pan-tilt camera or a multi camera approach. This brings the privacy issue into consideration, as the subjects in observation can feel less comfortable with zooming cameras or multiple cameras.

Some personal comments:

- I think that for determining the camera requirements, the environment settings should be constrained. This also applies to the system capabilities or features, as they do influence the developing.
- If a useful product is meant to be further developed in a reasonable time, all the features discussed in the first part of this report should be reviewed and principally, defined. In my opinion, the objectives of this project surpass the capabilities of one relatively experienced person, for the time given. Only after constraining all the features and evaluating the possible hardware approaches, a system analyst should design all the software modules and, in last term, the program be programmed. Real data should be gathered for proper developing.
- The code written by me can be useful for evaluation purposes, but it has some obsolete testing capabilities that should be rewritten – I think that this report will be more useful than the code itself.
- Some very useful pieces of code by other authors were used in this work. Sharing code is not only valuable for reproducing results, but it also can greatly impulse new research. For example, much time was invested in creating testing frameworks for databases that are already years old.
- As a final remark, I consider that a real 90% accurate system can be more than enough for earning teaching time if it is supplemented by manual checking of those classifications below a pre-set reliability threshold.

References

Balcoh, N.K. et al., 2012. Algorithm for efficient attendance management: Face recognition based approach. *IJCSI International Journal of Computer Science Issues*, 9(4), pp.146–150.

Behara, A. & Raghunadh, M.V., 2013. Real time face recognition system for time and attendance applications.

Belhumeur, P.N., Hespanha, J.P. & Kriegman, D.J., 1997. Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7), pp.711–720. Available at: http://vision.ucsd.edu/content/yale-face-database.

Chin, E.T., Chew, W. & Choong, F., 2014. Automated attendance capture and tracking system. *Journal of Engineering Science and Technology*, , Eureca 2014 Special Issue.

Chintalapati, S. & Raghunadh, M., 2013. Automated attendance management system based on face recognition algorithms. In *Computational Intelligence and Computing Research (ICCIC), 2013 IEEE International Conference on*. IEEE, pp. 1–5.

Dantcheva, A., Chen, C. & Ross, A., 2013. Makeup challenges automated face recognition systems. *SPIE Newsroom*, pp.1–4.

Delac, K., Grgic, M. & Grgic, S., 2005. Independent comparative study of PCA, ICA, and LDA on the FERET data set. *International Journal of Imaging Systems and Technology*, 15(5), p.252.

Erdem, C. et al., 2011. Combining Haar feature and skin color based classifiers for face detection. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, pp. 1497–1500.

Han, H. et al., 2013. A comparative study on illumination preprocessing in face recognition. *Pattern Recognition*, 46(6), pp.1691–1699.

Hassner, T. et al., 2014. Effective face frontalization in unconstrained images. *arXiv preprint arXiv:1411.7964*.

Hassner, T., 2015. Frontalization code v1.2. Available at: http://www.openu.ac.il/home/hassner/projects/frontalize/.

Heikkilä, M. & Ahonen, T., General Local Binary Pattern (LBP) implementation for Matlab, v0.3.3. Available at: http://www.cse.oulu.fi/CMV/Downloads/LBPMatlab.

Huang, G.B. et al., 2007. *Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments*, University of Massachusetts, Amherst. Available at: http://vis-www.cs.umass.edu/lfw/.

Liu, J. et al., 2014. Variable length dominant Gabor local binary pattern (VLD-GLBP) for face recognition. In *Visual Communications and Image Processing Conference, 2014 IEEE*. IEEE, pp. 89–92.

Mao, Y., Li, H. & Yin, Z., 2014. Who missed the class?—Unifying multi-face detection, tracking and recognition in videos. In *Multimedia and Expo (ICME), 2014 IEEE International Conference on*. IEEE, pp. 1–6.

Phillips, P.J. et al., 2000. The FERET evaluation methodology for face-recognition algorithms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(10), pp.1090–1104. Available at: http://www.nist.gov/itl/iad/ig/colorferet.cfm.

Priya, G.L. et al., 2014. Implementation of Attendance Management System using SMART-FR. *International Journal of Advanced Research in Computer and Communication Engineering*, 3.

Qi, X. et al., 2015. Globally rotation invariant multi-scale co-occurrence local binary pattern. *Image and Vision Computing*, 43, pp.16–26.

Reddy, T.M., Prasad, V. & Ramanaiah, N., 2015. Face recognition based attendance management system by using embedded linux. *International Journal of Engineering Research and Science and Technology*, 4.

Samaria, F.S. & Harter, A.C., 1994. Parameterisation of a stochastic model for human face identification. In *Applications of Computer Vision, 1994., Proceedings of the Second IEEE Workshop on*. pp. 138–142. Available at: http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.

Shehu, V. & Dika, A., 2010. Using real time computer vision algorithms in automatic attendance management systems. In *Information Technology Interfaces (ITI), 2010 32nd International Conference on*. IEEE, pp. 397–402.

Štruc, V., 2012. The PhD face recognition toolbox v1.0. Available at: http://luks.fe.uni-lj.si/sl/osebje/vitomir/face_tools/PhDface/index.html.

Štruc, V. & Pavešic, N., 2009. Gabor-Based Kernel Partial-Least-Squares Discrimination Features for Face Recognition. *Informatica (Vilnius)*, 20(1), pp.115–138.

Štruc, V. & Pavešic, N., 2011. Photometric normalization techniques for illumination invariance. In Y. Zhang, ed. *Advances in Face Image Analysis: Techniques and Technologies*. IGI-Global, pp. 279–300.

Štruc, V. & Pavešic, N., 2010. The Complete Gabor-Fisher Classifier for Robust Face Recognition. *EURASIP Advances in Signal Processing*, 2010, p.26.

Štruc, V. & Peyre, G., 2012. The INFace Toolbox v2.1. Available at: http://luks.fe.uni-lj.si/sl/osebje/vitomir/face_tools/INFace/.

Tan, X. & Triggs, B., 2007. Enhanced local texture feature sets for face recognition under difficult lighting conditions. In *Analysis and Modeling of Faces and Gestures*. Springer, pp. 168–182.

Turk, M., Pentland, A.P. & others, 1991. Face recognition using eigenfaces. In *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*. IEEE, pp. 586–591.

Viola, P. & Jones, M.J., 2004. Robust real-time face detection. *International journal of computer vision*, 57(2), pp.137–154.

Appendix 1: Log

Week 1: 31/08 – 05/09

Introduction. Delineation of plan of action. Mindmap of the problem and its parts. Gathering and review of previous work on the matter.

Week 2: 07/09 – 12/09

Literature collection and review. Read about camera specifications. Wrote to the HoD a request for a camera and permission to take pictures in class, with an overview of the proposed system.

Week 3: 14/09 – 16/09

Comparison of what was found in the literature review. Definition about the features of our approach and the things to be done. Started working in Matlab: Tried Simulink but discarded it. Wrote test programs for opening, modifying and saving images and videos.

Week 4: 21/09 – 26/09

Read about system objects and cascade detectors in Matlab. Read about pre-processing before detection. Wrote test programs for detection and extraction of faces in images and videos. Read more about camera specifications and rejected one camera offer.

Week 5: 28/09 – 01/10

Read about face recognition with Eigenfaces and implementations in Matlab. Read about classes in Matlab, decided to use them for abstraction. Started writing Eigenfaces feature extractor. First tests with ORL/ATT database.

Week 6: 05/10 – 09/10

More work on the feature extractor and testing programs. Took parts of PhD toolbox for feature extraction. Read about built-in knn classifiers in Matlab.

Week 7: 12/10 – 14/10

Read about k-fold cross validation and resubstitution error – and related Matlab functions. Classification works. First full-database tests with Mahalanobis and Euclidean distances. Started defining Class and Student classes. Classifier and Feature Extractor belong to the Class class.

Week 8: 19/10 – 22/10

Prof. Chandy tested one camera but the image quality was bad. Created StudentDatabase Class. Worked on test that reads student IDs from file and uses StudentDatabase, Student and Class classes. Tested some classification ideas. Tested Tan & Triggs pre-processing as implemented in PhD toolbox.

Week 9: 26/11 – 31/11

Tried classification with mean feature vector. Implemented parameters as global variables. Automatic splitting and pre-processing of test and training sets. Improved testing framework.

Week 10: 02/11 – 06/11

Completed better pre-processing function, easily modifiable. Implementation of distance threshold with Euclidean distance to mean feature vector of class. Carried out full tests to ORL/ATT and Yale databases, varying the different parameters. Installed openCV binaries for Matlab and Microsoft Windows SDK for being able to compile Zhu Ramanan facial feature detection for Tal Hassner's frontalization. Started working on InputSequence and DetectedPerson classes, for input program.

Week 11: 11/11 – 14/11

Read Xianbiao Qi paper about MCLBP/C and tested the program provided. Read about possible databases and papers to compare with. Wrote email to FERET for getting the database, downloaded color FERET database. Read about how to benchmark with grayscale FERET (Sep96), and started programming test. Decompressed pictures of database.

Week 12: 16/11 – 21/11

Tried eye detection but not robust enough. Used provided eye coordinates to rotate and crop faces. Extracted PCA, MCLBP/C and LBP features and evaluated results.

Week 13: 23/11 – 28/11

Debugging of the bad results – no evident problem found. Review of all the project and report writing.

The developed code has been delivered to Dr. Abraham Chandy. For further enquiries, the author can be contacted at aguadopd@hotmail.com.