Electronics & Computer Science
Faculty of Physical and Applied Sciences
University of Southampton

Tim Brooks

December 9, 2013

# Face Recognition in the Wild

First Examiner: Dr Paul Lewis

Second Examiner: Dr Mark Nixon

A Project Progress Report
Submitted For The Award Of:
MEng Computer Science

**Abstract**

OpenIMAJ is an award winning set of libraries and tools for image manipulation. Currently, a pipeline for Face Recognition in the Wild has not been implemented in OpenIMAJ, this project is implementing a Face Recognition in the Wild pipeline, and investigating its effectiveness against other methods. This report is to detail the progress made so far on the project of Face Recognition in the Wild. In this report, four main areas are examined of the project progress. From these, conclusions have been drawn detailing how the project has progressed. So far, the Evaluation methods have been completed, ready for the development of the pipeline. It can be seen that the project has not reached the expected stage initially set out at the beginning, and considerations from this have been made to create an outline of remaining work to be completed.

# Contents

# 1 Project Introduction and Goals

## 1.1 Introduction

Faces are used by humans as a way of recognising individuals, over the past few decades, algorithms have been developed that can recognise faces. These algorithms are constantly improving, as more and more technology is beginning to use Face Recognition. Face Recognition in the wild is the process of detecting and recognising faces that are not fully aligned to the camera, from more natural pictures. Face Recognition is a more trivial use of Face Detection, it incorporates recognition into detecting and recognising who the face belongs to. Currently, a Face Recognition pipeline has been implemented in OpenIMAJ, but it has not been very well tested, nor does it make use of the most effective method for Face Recognition.

## 1.2 Goals

The aim for this project is to investigate and implement the method of Face Recognition used in [6] in OpenIMAJ. This method is a pipeline of many state of the art features in Face Recognition, some already used in OpenIMAJ, this project will investigate those and use them in the final implementation. When the method has been implemented into OpenIMAJ, the project will explore the performance of the method against the expected results, and the results against other methods, including the state of the art High Dimensionality Local Binary Patterns (High-LBP) [1], and others, time permitting. An aim of the project is to design it such that it is similar to the design ideology of OpenIMAJ, having very high cohesion between objects.

# 2 Background and report of literature search

The paper Fisher Vector faces in the Wild [6] shows that when Fisher vectors are used alongside densely sampled SIFT features, the resulting face recognition system is capable of achieving state-of-the-art face verification performance on the Labelled Faces in the Wild [3] benchmark. The paper proceeds to discover how a compact descriptor can be learnt from the fisher vectors using discriminative metric learning, since fisher vectors are highly dimensional. The conclusion of this paper is supported by a set of tables and graphs, detailing the results of tests on the Labelled faces in the wild dataset. It shows that when outside training data is used (unrestricted setting), the pipeline has the second highest mean at 0.9303 with a slightly smaller error margin. This is second to High-Dimensionality LBP at 0.9318 accuracy. In a Restricted setting, where there is no outside training data, the method yielded the highest accuracy of 0.8747, meaning it is the new state-of-the-art in that particular setting. The paper describes the steps it took in order to achieve its results, which uses many advanced facial recognition methods. In this literature review, the methods used will be examined, along with work related to Face Recognition.

In [6], there is a description of the processing pipeline used to achieve the high standard of facial recognition. This consists of the following parts:

- Facial landmark detection
- Align and crop face
- Dense Scale Invariant Feature Transformation, Gaussian Mixture Modelling and Fisher Vector Encoding
- Discriminative Dimensionality Reduction
- Compact Face Representation

## 2.1 OpenIMAJ

OpenIMAJ is an Open Intelligent Multimedia Analysis toolkit for Java [2]. It consists of a set of libraries and tools used for multimedia content analysis and content generation. OpenIMAJ already utilises several components needed for the Fisher Vector face recognition implementation. It contains the necessary code for Facial Landmark Detection, Align and Crop, Dense SIFT, Principal Component Analysis. For Facial Landmark Detection, there is a class in OpenIMAJ that can be used, the FKEFaceDetector (Front Keypoint

Enriched Face Detector). There is also the AffineAligner that can be used in order to perform the Align and Crop face.

## 2.2 Dense SIFT

Dense SIFT as described in the Fisher Vector Faces in the Wild paper is adapted is the process of extracting distinctive invariant features [4]. It works by performing 4 stages to transform image data into scale-invariant coordinates relative to local features on the face.

1. Scale-space extreme detection. This searches all scales and image locations to identify points that may be of interest that are invariant to scale and orientation. This is a preliminary step to find out which points would be suitable for feature matching.

2. Keypoint localisation. At the points found from step 1, a detailed model is fit to determine their location and scale. Keypoints are then chosen based on their measure of stability.

3. Orientation assignment. In order to provide invariance to the transformations about to happen in 4., one or more orientations are assigned to each keypoint location basied on the local image gradient directions. Any future operations will be performed on image data relative to the assigned orientation, scale and location for each feature.

4. Keypoint descriptor. By measuring the local image gradients at the selected scale in the region around the keypoint, they can be transformed to allow for significant levels of local shape distortion and change in illumination. This allows us to determine the image features.

With the above steps completed, the image features are extracted from the set of reference images and stored in a database ready for further processing. SIFT points are extremely useful due to being highly distinctive. These features are also known as descriptors.

## 2.3 Fisher Vector Encoding

Fisher Vectors describe a set of local features in a single vector. Fisher Vector encoding is the process of combining a large set of vectors into a high dimensional vector representation. By using a Gaussian Mixture Model (GMM) to the vector elements, the derivatives of the log-likelihood of the model can be encoded with respect to its parameters. By using the set of local features from Dense SIFT, a highly dimensional Fisher vector can be created. The

GMM is seen as a face model, and if plotted on the image, the mean and variance of GMMs are represented by ellipses of all the features. We can create a representation between the features and the GMM centres by training a GMM with diagonal covariances, then calculate the derivatives with respect to the Gaussian mean and variances. As shown in [5], L2 Normalisation can improve the application of the Fisher Vector

## 2.4 Large-Margin Dimensionality Reduction

Large-Margin Dimensionality Reduction is the step in which the Fisher Vectors are compressed to small discriminative representation. This stage is where the calculation is made that determines if the two given faces are the same or not. In [1] it details the method, explaining how a high dimensional fisher vector can be projected to a low dimensional vector, in order for the squared euclidian distance can be calculated. If the squared euclidian distance between the two images are smaller than a threshold, then they are the same person in the image.

## 2.5 Labelled Faces in the Wild Dataset

The Labelled Faces in the Wild Dataset is a database of images collected for the purpose of testing face recognition systems. The main problem it is used for is such that, Given two pictures, each of which contains a face, decide whether the two people pictured represent the same individual [3]. The database contains 13,233 images, of 5,749 different individuals. 1680 individuals have two or more images in the database, whilst 4,069 have just one. The database contains a wide variety of faces, varying on many factors including pose, lighting, expression, background, race, ethnicity, age, gender, clothing, hairstyles, camera quality, colour saturation and focus. This give a wide range of samples to use in order to test the face recognition system.

The way that this dataset works is by using the pre-defined views (sets of images, some positively matching and others negatively matching). The first is for algorithm development, and the second is for performance reporting. Using these two different views helps ensure that the algorithm will not be skewed for the final results.

# 3 Report on Technical Progress

## 3.1 Analysis

### 3.1.1 Requirements

A requirements specification has been created for an implementation of an algorithm, using research discussed in section 1. Functional Requirements describe what the project will do. Non-functional requirements explain how the project will do so.

**Functional Requirements**

- The project will use the Fisher Vector encoding face recognition pipeline used in [6].

- The project will compare two images and output if they are the same or not. The ability to assess if two images are the same is a basic requirement of face recognition.

- The project will use the Labelled Faces in the Wild image dataset - The Labelled Faces in the Wild dataset is a very common dataset used to test the effectiveness of face recognition algorithms. By using this dataset, I am able to compare to published results of other face recognition methods - including the results achieved by [6].

- The project will be able to evaluate the results by creating a Receiving Operating Characteristic Equal Error Rate (ROC-EER) graph. A ROC-EER graph is created by plotting the proportion of true positives out of the total actual positives, against the fraction of false positives out of the actual total negatives, this is useful as it will allow comparison between the implementation and other face recognition methods.

- The project will be able to evaluate by outputting a classification accuracy - the percentage of image pairs correctly classified. This will also help evaluate the implementation against the other face recognition methods.

**Non-functional**

- The project will be written in Java and be able to work with OpenIMAJ - The main aim of this project is to implement the face recognition in OpenIMAJ, a Java library.

- Use existing classes in OpenIMAJ where possible. OpenIMAJ has many classes ready to be used, the implementation of this project will be in such a way that if there is an already existing method, it will be tested to ensure its effectiveness, and used.
- Create each step of the pipeline individually, so they can be used standalone in the future. Such to the nature of OpenIMAJ, by ensuring that each algorithm used is implemented in its own class, it will be able to be used in other OpenIMAJ projects, not just this one.

### 3.1.2 Risks

The table in Appendix A.1 outlines the possible problems that would affect the project. Each problem has a an explanation, along with how much it would impact the project (in terms of the amount of time needed to overcome the problem). Along with that, a risk factor is given, determining how much of a risk the problem would be. As you can see, the highest risk is a hardware failure on the development machine, or data corruption. For each risk, a contingency plan has been created, detailing what would happen in each eventuality. A contingency plan has been created, as shown in Appendix A.2. A contingency plan is very useful to have created, as it will not only help overcome and problems in the project if they happen, but will help prevent those problems in the first place.

## 3.2 Design

### 3.2.1 Proposed Final Design

This project has been split down into several sections in order to help structure the implementation of it. Each section is as follows:

- Evaluation Methods
- Facial Landmark Detection
- Dense SIFT
- Gaussian Mixture Model
- Fisher Vector Encoding
- Discriminative Dimensionality Reduction
- Compact Face Representation

- Pipeline implementation

The way that the implementations have been planned allow a clear design structure to be concluded. Each implementation will be created as an individual class or set of classes. This will result in a highly cohesive project, helping with project management. In the long term it will encourage the reuse of code in other OpenIMAJ projects. The evaluation methods have been created first in order to allow for testing to take place throughout the project. Below is a class diagram, based on work completed and plans for future work. [INSERT CLASS DIAGRAMS] The project will run from command line to output the results for the evaluation. If time allows, this project may create a Graphical User Interface to run the evaluation tools, however, this is not a requirement for project success.

### 3.2.2 Tools

This project will be developed using Eclipse as a development environment. Eclipse allows for easy cross platform integration, allowing the opportunity to work on Mac, Linux and Windows with ease. Eclipse has an advanced set of debugging tools available to be used, which will be of great help if the project comes in to any difficulties.
OpenIMAJ, whilst is the focus of this project, is going to be used in development. OpenIMAJ contains several tools needed for implementing the face recognition method, and they will be tested to ensure suitability and used accordingly.
LaTeX will be used to create and present the Project Brief, Progress Report and the Final Report. LaTeX is a document preparation and typesetting program to output a report. This will give the project more flexibilty over the content of the reports, and allows the author present a professional report each time. To write the LaTeX, the program TexMaker will be used. This program is a productive way to write LaTeX, with an built in compiler, it outputs the document in the same window, allowing for fast writing and formatting.
BibTeX will be used to handle references in the LaTeX document. This allows for fast and easy referencing in the reports. A separate .bib file is created in a text editor and populated with the references in the BibTeX format. They can then be easily used in the LaTeX document.
To ensure that no work is lost, Git version control will be used in conjunction with a remote repository hosted privately on Github. Git is useful for several reasons. It is a way of backing up to a remote location, and so will reduce the effects of a large loss of work from a local workstation, as it will

be available to access from Github. Being able to use version control is great for development, as new branches allow work to be carried out without affecting the master copy. This will be especially useful in the stages of the project where the face recognition method is being optimised to achieve the best results possible. Git will also be used to store and version control the Project Reports.

## 3.3 Work completed to date

So far in this project, work has been spent on research and first stages of development. At the beginning of this project, a gantt chart was created Appendix B which shows the breakdown of each task (Work before Christmas holiday B.1, and work after Christmas holiday B.2. No work was planned during the Christmas holidays due to January exams and other coursework) and the time taken to complete them. At this stage of the project, everything up until the Implementation of Fisher Vector Encoding should be completed. However, due to a variety of reasons, this is not the case. The author has found that due to the topic of the project, several stages took far longer than originally planned. The background reading and research taking longer than planned, this was due to the reading into face recognition, and time was being spent trying to fully understand the methods employed by [6]. Approximately 50 hours has been spent in total on Background Research and Background Reading, with larger amount of time being spent on research and into OpenIMAJ, including learning how to use it and research into already available packages. The literature review also took longer than planned, at about 25 hours. To this date, the work completed is up to the Evaluation Methods, with them taking approximately 20 hours, as opposed to the 8 hours.

Within the Gantt Chart, I included other Semester 1 courseworks, this is vital to help with project management, as it allows the project to be looked at from a wider scale of the semester, and if internal deadlines set are likely to be met. Whilst these are setbacks, they are setting a better basis for the rest of the project. More time will be allocated over Christmas to make up the time that has been lost. From experience and knowledge learnt from Semester 1, a new Gantt Chart has been created to reflect the work already completed, and the plan of the work to be completed. [Appendix New Gantt Chart - to insert]

# 4 Plan of remaining work

Work has been split into wider sections, the implementation and the evaluation. By creating deadlines for both of these sections, deadlines internal to the sections can be created and utilised. This will give the best opportunity possible for the project to be kept on track and finished within good time. For the implementation, the following needs to be completed

- Dense SIFT
- Gaussian Mixture Model
- Fisher Vector Encoding
- Discriminative Dimensionality Reduction
- Compact Face Representation
- Pipeline implementation
- Testing

Some of this work will be completed over the Christmas holidays, although time needs to be considered for the authors 4 January exams and January coursework deadline. For the evaluation, the following needs to be completed

- Evaluation using the Labelled Faces in the Wild dataset
- Evaluation of other methods using LFW
- Comparison of different methods and datasets
- Final Report

From what has been learnt from the original Gantt Chart, the updated Gantt Chart reflects the work to be completed. The project aims to be completed towards the end of March (18th-25th March), with further work to be carried out on the Final Report up until the deadline. A deadline of the 14th April has been set in advance of the course deadline to ensure the project is completed in good time.

# Appendices

## A Appendix A

### A.1 Risk Table

Table 1 – *Risks*

| Risk ID | Risk | Outcome(s) | Impact | Risk Factor |
|---|---|---|---|---|
| 1 | Hardware failure on the main development machine | System would need rebuilding or hardware would need replacing. Would cause large delays | 5 | High |
| 2 | Data corruption | Work would have to be restarted from the most recent backup. Depending on when the most recent backup was, this would cause large delays. | 5 | High |
| 3 | Authors personal issues (e.g. illness) | Could cause long term delays to the project development | 4 | High |
| 4 | Misjudging time needed for the project | The whole project wouldn't be completed to it's full potential, this would affect the final outcomes as set out in the requirements | 3 | Medium |
| 5 | Dataset no longer available | This would cause a number of issues and slight delays due to the evaluation method being based around the Labelled Faces in the Wild dataset. | 3 | Medium |

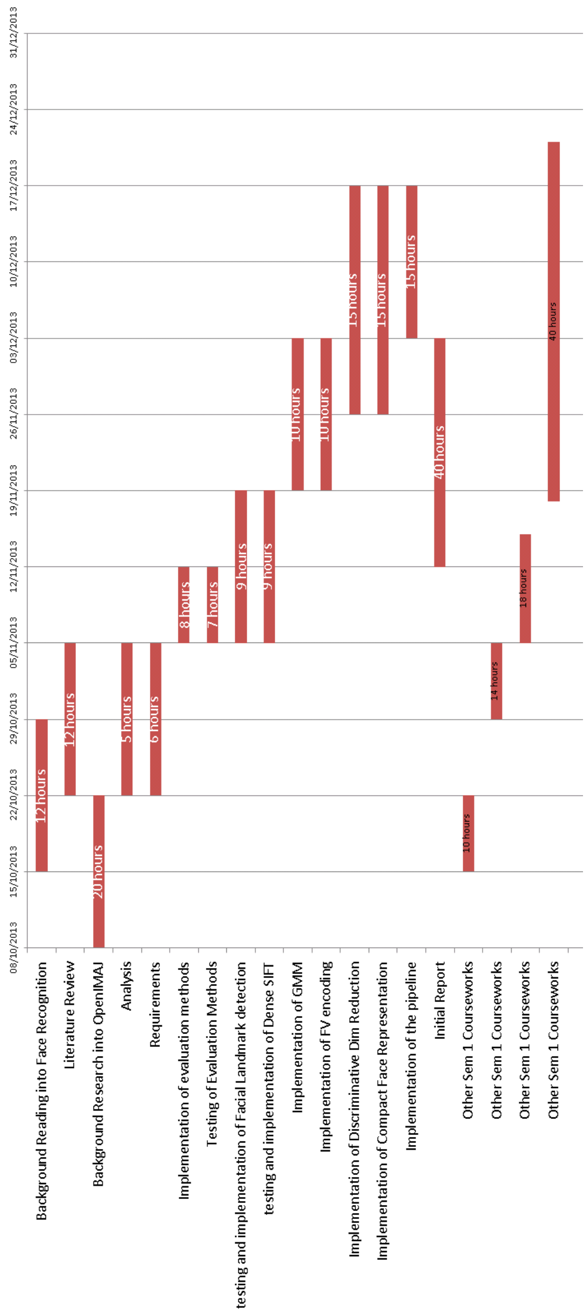
## A.2 Contingency Plan Table

Table 2 – *Contingency Plan Table*

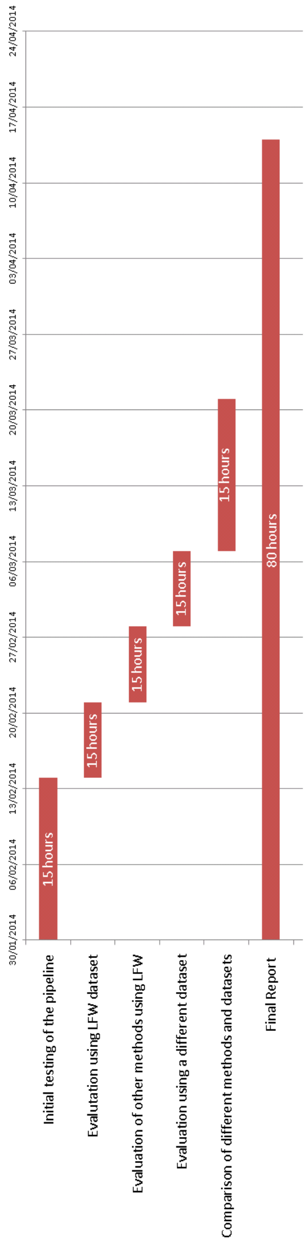
| **Risk ID** | **Contingency Plan** |
|---|---|
| 1 | In the event of a hardware failure on the main development machine, the plan of action. If this happened a few days before a major deadline, a new machine would be used to continue the work, provided there is a very recent backup. If it happened further from the deadline, or no backups have been made of the recent work, attempts to fix the machine would take place. |
| 2 | In the event of data corruption, the most recent copy will be downloaded from the version control repository. A main reason why Git is being used at an external repository is for the backup functionality, hopefully preventing any data loss. This is only a successful plan if the author is consistent at using Git |
| 3 | In the event of the author having personal issues, advice would be sought after from the project supervisor and personal tutor, Dr. Paul Lewis and Dr. Enrico Costanza, and if necessary, further action will be taken to make the Faculty aware of the issues. |
| 4 | In the event of there not being enough time to complete the project due to poor time allocation, efforts would be focussed on completing as much as possible to achieve the minimum set out in the requirements. |
| 5 | The dataset has been downloaded, so in the event of the dataset no longer being online to download, the project will not be affected. |

# B Appendix B

## B.1 Preliminary Gantt Chart, work before Christmas



## B.2 Preliminary Gantt Chart, work after Christmas



# References

[1] D. Chen, X. Cao, F. Wen, and J. Sun. Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification. *Int. J. Comput. Vision*, 2004.

[2] J. S. Hare, S. Samangooei, and D. P. Dupplaw. Openimaj and imageterrier: Java libraries and tools for scalable multimedia analysis and indexing of images. In *Proceedings of the 19th ACM international conference on Multimedia*, MM '11, pages 691–694, New York, NY, USA, 2011. ACM.

[3] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller. E.: Labeled faces in the wild: A database for studying face recognition in unconstrained environments. 2007.

[4] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, nov 2004.

[5] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *Computer Vision–ECCV 2010*, pages 143–156. Springer, 2010.

[6] K. Simonyan, O. M. Parkhi, A. Vedaldi, and A. Zisserman. Fisher vector faces in the wild. *IJCV*, 60(2):91–110, 2004.