Towards a Facial Recognition Workstation

With Accompanying Examples

Matthew Shane Kingon
Supervisor: Prof EP Wentworth

Submitted in partial fulfilment of the requirements for the degree of BACHELOR OF SCIENCE (HONOURS) of Rhodes University

RHODES UNIVERSITY
Where leaders learn

Computer Science Department
Rhodes University
Grahamstown, South Africa

Abstract

Abstracty things

ACM Classification

classificationy things

Acknowledgements

I would like to acknowledge the financial and technical support of Telkom SA, Tellabs, Genband, Easttel, Bright Ideas 39, THRIP and NRF SA (UID 75107) through the Telkom Centre of Excellence in the Department of Computer Science at Rhodes University.

Contents

1 Introduction **1**

2 Research of background Literature **3**

2.1 Introduction to Literature Review 3
2.2 Specific Fields 4
2.2.1 Making a Toolbox 4
2.2.2 Background context 4
2.3 Image representation overview 5
2.4 Eigenface Algorithm 6
2.4.1 Intuitive description 6
2.4.2 Training 7
2.4.3 Prediction 10
2.4.4 Summary of EigenFaces 11
2.5 Normalization 11
2.5.1 Cropping 11
2.5.2 Lighting 12
2.5.3 Alignment and scaling 12
2.6 Conclusion 13

3 Illumination **15**

3.1 The Issue of Illumination 15
3.2 Comparing Two Methods 16
3.2.1 Histogram Equalization 16
3.2.2 Summary of Histogram Equalization 17
3.2.3 Mean Illumination Estimation 18
3.2.4 Summary of Mean Illumination Estimation 21

3.2.5 Validation of MIE implementation . . . 22
3.3 Reading The Graphs . . . 24
3.4 Experiment 01: Setup . . . 24
3.4.1 Experiment 01: Results . . . 25
3.4.2 Experiment 01: Discussion . . . 27
3.5 Experiment 02: Setup . . . 27
3.5.1 Experiment 02: Results . . . 28
3.5.2 Experiment 02: Discussion . . . 31
3.6 Summary . . . 32

4 Assignment Problem **33**
4.1 The Problem . . . 33
4.2 Hungarian Algorithm . . . 34
4.3 Summary . . . 37

5 Conclusion **39**
5.1 Summing up . . . 39
5.1.1 Working Example . . . 39
5.1.2 Theory of future work . . . 40
5.2 Looking ahead . . . 40

A Appendages **41**

References **41**

Chapter 1

Introduction

Chapter 2

Research of background Literature

2.1 Introduction to Literature Review

This work investigates the feasibility of using facial recognition as a means to track classroom attendance. It's worthy to note that there have already been many attempts to do so, some having been more successful than others. However, this work differs from the others as we hope to design a system that can be easily extended or modified to test out new ideas to improve recognition. Not only do we wish to offer a system you can easily plug new normalization techniques into but also able to make use of additional information that pertains to a specific problem a user may be facing.

Possible additional information that could be exploited include the knowledge that students tend to sit in the same area each day often varying their position by little more than a seat or two. This knowledge could be used to strengthen accuracy ratings should an individual known to sit at that location is identified. Another aspect could be useful to such a system is the knowledge that prior to lectures it is already known who should be there. Thus an attempt can be made to optimize the solution between two sets, namely a set of present faces, and the class-list set.

Facial recognition is a complex field and has been well researched over the past decades even so, it is far from being a fully understood or solved problem. An aspect clearly portrayed by the fact that there are many variations in methods and techniques out there to solve this problem. As such this work attempts to create a tool-kit platform for facial detection and recognition. This platform will act as scaffolding for the addition of any feature related to facial recognition, be it pre-processing or actual facial recognition algorithms.

The work provides an illustrated application of this platform by implementing facial recognition for lecture attendance tracking. This work focuses on extending the pre-processing side of the tool-kit using the already provided OpenCV implementation of Egienfaces to do the actual recognition. One notable extension being that of the Mean Illumination Estimation algorithm. Some more concepts that are added include; image cropping, orientation correction and plane alteration. All of the above concepts describe various aspects of image normalization.

2.2 Specific Fields

2.2.1 Making a Toolbox

One of the constantly developing, key aspects to the research project is to create a face recognition tool-kit. The idea is to selectively add relevant image manipulation techniques or other such features to the code base, thus allowing the client to mix and match them and after application get a report stating how successful the combinations chosen are. Some features would be cropping the faces out from the background noise, others would aim to control lighting. Hence this section could get very lengthy as each aspect is researched.

The toolbox is developed using Python and the OpenCV library in conjunction with the mathematical, Numpy library. However, to limit the scope of the project from getting to ambitious the system this work implements will, at least initially, be console based.

2.2.2 Background context

Despite modern day technology many school environments still struggle with the problem of class/lecture attendance tracking. Some may ask, why do we need such a tool to track attendance? Tracking attendance has many useful benefits for schools and universities the obvious one is that many students try skip lectures to avoid work. Thus tracking their attendance would help in identifying such students. This would, hopefully, result in larger attendance of such classes/lectures.

The standard solution to this problem has varied slightly but for the most part has either been a simple piece of paper passed around the class letting the students sign/tick their names off (mostly used in universities), or a roll call at the start of the class by teachers in lower level class room environments (primary/high-school).

Thus it shouldn't come as a surprise that there have been many attempts to solve the problem of lecture attendance tracking and hence remove some issues. some of the main ideas put forward are: fingerprint scanning systems, iris scans, card readers, voice recognizers etc. The problem with these systems is that they are still all rather intrusive workarounds, requiring students to take an active part in their attendance tracking, this results in either lines outside of lecture room venues as students wait to verify that they are in attendance, or alternatively, a rather distracting procedure to do while they could be listening to the lecture.

Many past papers on this topic have addressed the existence of these issues in some context or another. [1] [2] [3] Now as many agree facial recognition has the potential to be a very simple, and non-intrusive means of tracking attendance, as in the ideal case it would simply need a camera at the front of the class and as the lecture goes on it identifies all students present. However, the technology available today is still not robust enough, hence the need for further research, development and refinement in this field. Some points to consider are lighting as it is a very big problem that has had many attempts at a solution most are not satisfactory as they degrade the image too extensively. A more hardware sided issue would be camera quality.

It should be noted that facial recognition isn't a perfect science to start with. Many solutions don't even take into account that they are attempting to recognize a face. These algorithms could be more accurately described as object recognizers, some rather popular examples of this type of system include Egienfaces, Fisherfaces. However, there do exist systems that can achieve accuracy close to that of a human. This work takes into account many of these issues and also attempts to use outside knowledge to recognize students (seating patterns, clothing colour etc.)

2.3 Image representation overview

The OpenCV library was chosen as it provides many useful image manipulation and computer vision techniques. However, this means a solid understanding of how OpenCV represents these images is required to best make use of the provided functionality.

OpenCV has already overloaded many mathematical operations to take their representation into account. Hence it is possible to simply take two images imported via "cv.imread(...)" or other such methods and add or subtract them with a ("+" or "-"). However, this is implemented only for basic mathematical operations. When one wishes to perform more complex arithmetic procedures one needs to take into account the representation of an image.

Little more than grey scaled images are required for many computer vision techniques including ones this work makes use of. Thus the matrix representation that describes the images this work makes use of is that of a simple 2D array or, mathematically, a 2D matrix. This comprises the core of an image class. However, there are many other headers that are provided by an image class, these include headers that describe the width and height of the image, the mathematical representation of the values inside the matrix (i.e. 8,16,32 bit numbers weather or not they are floats etc), name of the image, how many channels it has (Red, Blue, Green) and weather or not it has an alpha channel (transparency). These comprise the most important features of an image [4].

It is noted that OpenCV makes use of Numpy, a mathematical matrix library, for many of its built in procedures. This is possible as OpenCV interprets the way Numpy represents matrices as images. Which is useful to client programs as Numpy can thus be used to take care of the heavy lifting with regards to maintaining an image's meta data. Thus providing the client with a simply view of an image as a 2d array that can be manipulated as such.

2.4 Eigenface Algorithm

2.4.1 Intuitive description

The Eigenface method of facial recognition works by taking the high dimensional face images represented mathematically as an $m \times n$ matrix, Providing it with N such images it takes them and finds the average of the matrices(images) i.e. sum them together pixel by pixel and divide by N. With this "Average face" new images are created by subtracting the training images from the average image. This represents each face as a difference from the average. Once this has been done a set of orthonormal basis matrices are calculated. Each difference face can then be projected onto the bases, resulting in a feature vector of co-efficients. To best represent these "difference faces". Indeed, once a system is trained with a set of images, given any of these training images you should be able to get coefficients that reconstruct the original face exactly. Represented by an Average face plus a linear combination of the basis matrices. The feature vector obtained determines the weighting of each basis matrix to enable complete reconstruction of the training face.

With these we can construct a face that somewhat represents one of the individuals we used in our training set by taking the average face and adding varying components, determined by a set of coefficients, of our basis images. This set of coefficients is called the feature vector of the difference face providing us the means of recognition, as for similar faces (presumably of the same person) the feature vectors will be very close.

2.4.2 Training

The Eigenface method requires training, this means that it needs to be given images of the faces it should recognize. For example the set of faces shown in figure 2.1:

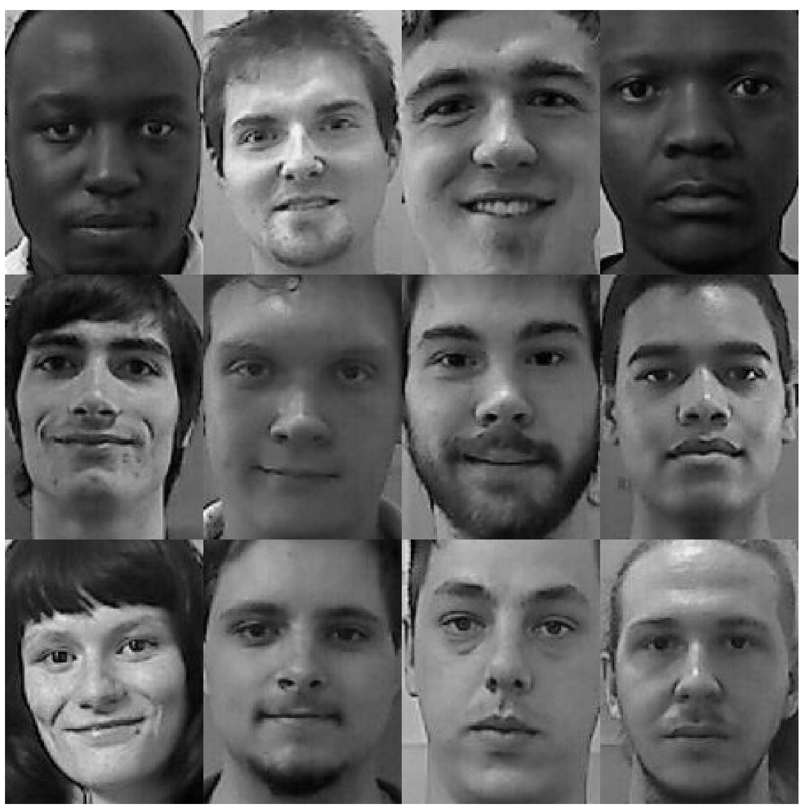

Figure 2.1: example of a training set

It then takes each image and converts it into a high dimensional vector created as:

$$\Gamma_n = (Width) \times (Height) \quad | \quad n = 1, .., N$$

Where N is the Number of training images you have. You then get a set S of N such face image matrices:

$$S = \Gamma_1, \Gamma_2, \Gamma_3, ..., \Gamma_N$$

After this is done the method finds the average face given as:

$$\varphi = \frac{1}{N} \sum_{n=1}^{N} \Gamma_n$$

The average face constructed from this training set can be seen shown below in Figure 2.2:

Figure 2.2: The Average Face

Once the "average face" is determined the method calculates the difference ϕ between it and each image in the training set.

$$\phi_n = \Gamma_n - \varphi$$

Figure 2.3: The "Ghost" set created via subtraction of each face from the mean.

Figure (2.3) above shows this, each facial image below maps to the corresponding input face as was shown in Figure (2.1) minus the average face which was displayed in Figure (2.2). We next obtain the covariance matrix C which we need for its Eigenvectors/values (μ, λ) respectfully. We obtain C via:

$$C = \frac{1}{N} \sum_{n=1}^{N} \phi_n \phi_n^T$$

$$= AA^T$$

$$A = \phi_1, \phi_2, \phi_3, ..., \phi_n$$

$$L_{mn} = \phi_m^T \phi_n$$

Allowing us to find the eigenvector/values by:

$$\mu_i = \sum_{n=1}^{N} \nu_{ik} \phi_k \quad i = 1, ..., N$$

Once done, we find a set M of orthonormal vectors μ_n that best describe the distribution of the difference faces. We choose vector k, μ_k such that:

$$\lambda_k = \frac{1}{N}\sum_{n=1}^{N}(\mu_k^T \phi_n)^2$$

is maximized, subject to the constraint:

$$\mu_n^T \mu_k = \delta_{nk} = \begin{cases} 1 & \text{if n=k} \\ 0 & \text{if } n \neq k \end{cases}$$

Note, the superscript T implies the corresponding matrix is transposed.

2.4.3 Prediction

Once the recognizer has been trained with the input training data, what it stores are the average face, orthonormal basis matrices and the feature vectors for each individual in the training data. Then it can be fed some unseen images of the people it has trained on and see how it fares. Herein we describe the procedure of taking a new image and testing it against our trained recognizer.

First we subtract the average face, Then we produce its feature vector by:

$$\omega_n = \mu_n^T(\Gamma - \Phi) \quad \Omega^T = [\omega_1, \omega_2, ..., \omega_n]$$

We now determine which of the training faces is the best fit by finding the feature vector of the training face with a minimum euclidean distance to the feature vector of the probe face:

$$\varepsilon_n = \|\Omega - \Omega_n\|$$

It should be added that Euclidean distance is not the only means of determining how different vectors are. Indeed, for our purposes it is possible that it could even be detrimental to the recognition rates. Another solution would be to use the absolute distance. This is still not 100 percent ideal but it may not have as much of an effect on our accuracy scores.

If ε_n is below a certain threshold defined within the algorithm the face is considered to be known and represented by Ω_n. If instead the ε_n is above the threshold the image is determined not to be face from the training data or indeed a face at all. If the threshold value is chosen too small only very close approximations to our training set will be accepted by the algorithm leading to a higher accuracy, at the other end, if the threshold is too large the algorithm will generate many false positives. If the image is a face you know belongs to one of your subjects but the system determines it is an unknown, then you could choose to add it into the set of known faces and repeat the training steps i.e. incorporate the latest image of the individual into the system.

2.4.4 Summary of EigenFaces

So in summary, the eigenface algorithm is provided with a set of face images to train on, then once it is trained it is given an image, presumably a face of one of the people from the training set, it will then ideally match it to the correct person and report how close the two feature vectors are to each other are in the space provided.

It is important to note even though this method is called the Eigenface method, nothing about it forces the use of facial images, indeed it is simply a image recognizer that has been shown to work well on faces. An example of PCA systems being used on non-face objects can be seen in [5] where they use the idea to identify different plant types based on individual leaves from each plant. Also as it is an image recognizer and not a face recognizer, one known weakness is that lighting will have a very large impact on its performance, as opposed to other methods. Though in other methods lighting does play a part and is something you wish to remove, it is highly detrimental to the Eigenface method. Thus to build a robust system lighting will need to be normalized and compensated for.

The above Mathematical proof and understanding of the implementation of the eigenface algorithm was achieved via the tutorial found at [6]. With Recognition given too Matthew Turk and Alex Pentland for developing the original algorithm [7] .

2.5 Normalization

2.5.1 Cropping

As has been stated the Eigenface algorithm is an image recognizer, thus background image data will have a drastic impact on its performance and recognition rates. For this reason it is important to get rid of as much of an images background as possible. Cropping an image is easy by hand, but the point of this whole exercise is to automate the process of facial recognition as much as possible. Hence, to crop a face out of an image the face would first need to be found. To this end, we would need a face identifier.

This work already implements such a feature in a separate component of the project [8]. As this work attempts to create a facial recognition tool-kit, the client can specify the bounds of the cropping procedure once the face is found giving said user the ability to crop it aggressively or not at all.

Preliminary testing does indeed show that cropping an image has an effect on the recognition rate. However, these have been manual crops that do not resize the image nicely. Also, some faces are simply not well cropped with a lot of background left in the image. Another improvement that can be attempted is to white/black out the remaining background so that for all images the background makes the same contribution to the algorithms scoring system.

2.5.2 Lighting

Lighting plays a very integral role for the eigenface algorithm. As such, we expand upon this concept more fully in chapter 3 by providing a strong background as to why lighting is an issue and experimenting with various ways to overcome said issues posed by illumination.

2.5.3 Alignment and scaling

The final big normalization issue would be face alignment and scaling, when a photo is taken/camera is run, the faces wont be all similarly scaled or aligned. This is something system needs to take this into account.

Hence this becomes a software problem. Most alignment algorithms find suitable features in a face with which to re-align the face correctly [9]. Such features can include the location of the eyes in a face and use these to re-align the head so that it is as straight on as possible. This could be achieved by taking the eyes, drawing a line between them and levelling this line out so that it is straight. It should be noted that the nose can also be used as an alignment feature but the eyes are favoured as they provide a longer axis to align with.

Scaling of a face can also be achieved through the distance between the eyes. The image as a whole can be scaled bigger or smaller so that this distance conforms to a fixed value.

2.6 Conclusion

This work aims to create a facial detection and recognition tool-kit. However, the true extent of this goal is beyond the scope of this paper. The main goal of this work, at least initially, is to get an end-to-end system up and running even if many features require manual input. For example; cropping and alignment can be done by hand by prompting the client to locate the centre of the eyes in an image. Taking these locations, it can realign the head and set the eyes to fixed locations.

The bulk of the current work focusses on the eigenface algorithm for facial recognition. More importantly, it attempts to fully understand and implement the normalization technique called the Mean Illumination Estimation [10] and ascertain the benefit on egienfaces accuracy rating by using this pre-processing technique.

As was explained earlier a key aspect of the system is that its framework and structure will be easy to extend and utilise. Allowing future researchers to add the functionality they desire whether they wish to add extra pre-processing methodologies or a more robust recognition algorithm.

Chapter 3

Illumination

3.1 The Issue of Illumination

Lighting plays a very big role in all facial recognition, identification or just about any image processing problem. To the human eye, it can be an almost undetectable irregularity in the world we observe but to a computer that must inspect each and every pixel within an image to determine what it is observing, even the slightest change in lighting make each pixel value change dramatically. This problem can complicate any image processing techniques.

To better appreciate the problem it should be noted that there are an abundance of illusionary images that manage to fool the human brain. These examples are often very specifically designed for this very purpose. However, it does illustrate the point. The example shown in 3.1 may be an old one, but will still often fool most people you show it to; the two blocks with the orange dot are in fact, the same shade of grey, the two orange dots are also the same colour. Some may see it right away but even so it takes effort to see it. This is due to the fact that your brain logically assumes that the lower dot is on top of the brighter block set and hence should be brighter than the darker block set around it which it is. However, it is not brighter than all dark blocks in the image, this occurs because your brain doesn't take into account the shadow cast upon the blocks.

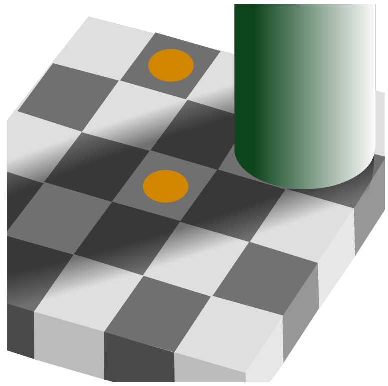

Figure 3.1: Optical Illusion example

When it comes to faces there are two main ways illumination could degrade our images. One, a uniform change in ambient light. This occurs when the light source is straight on, visually the whole image gets darker or brighter. Two, localised lighting changes, e.g. the light source is to the right of the face, so the left side is darker than the right. This is a distinction we will come back to later. Illumination is clearly something any face recognition service will have to take into account if it wishes to accurately recognize a set of faces. Thus, illumination is a logical and readily available normalization target to test the ease of use of my system.

3.2 Comparing Two Methods

The plan will be to test a newly proposed method of illumination correction, The Mean Illumination Estimation (MIE) [10], against something with proven results, namely, OpenCV's built-in histogram equalization. We have the proposition put forward by [10] that their algorithm bests the current standard for illumination correction. This implies that it should fare better than the simple quick and easy OpenCV solution.

3.2.1 Histogram Equalization

As stated we will be employing OpenCV's implementation of this algorithm. However, presented below is an overview on how the algorithm works and why it is theoretically useful to correct for different illumination conditions.

Before we get into the math of the method an intuitive description would be useful. Ideally what would happen using histogram equalization is given an image with 256 pixels and applying histogram equalization, one pixel will have a value of 1, another would have value 2, another would have value 255 etc. Or if we had an image of 512 pixels, two would get a value of 0, another two would get 1 etc. However, doing so we would lose structure to our image as pixel values originally of the same intensity get mapped to different values [11]. So, at a minimum we need to make sure that pixels of the same intensity value in the base image get transformed to the same "normalized" intensity. We also need to be careful that we don't affect the base structure of the image in another way i.e. pixel (x_0, y_0) which is brighter than pixel (x_1, y_1) put through this algorithm to get (x_0', y_0') will still be brighter than (x_1', y_1') though both values will likely have changed.

Mathematically we are mapping a clustered distribution to a wider (0-255) more uniform spread of intensity values. Let f be the given image represented by a $n \times m$ matrix of pixel intensities ranging from (0 to L-1) where L is the number of possible values (usually 256) Now p will denote the normalized histogram of our image f as a list of bins per intensity value:

$$p_n = \frac{x_n}{Total} \quad n = 0, 1, 2, ..., L-2, L-1 \tag{3.1}$$

where x_n is the No. of pixels with intensity 'n' and 'Total' is the total No. of pixels in the image. We will define the Histogram equalized image as 'g';

$$g_{ij} = floor((L-1) \sum_{n=0} f_{ij} p_n) \tag{3.2}$$

Note floor rounds down the value. The explanation used here can be found from the University of California [12]

3.2.2 Summary of Histogram Equalization

Now we have a normalised image with which we can hopefully perform recognition on with greater success. We note that technically it doesn't really care that there is illumination issues present. Histogram equalization is a global image operation that will effect the entire image in the same way, regardless of the actual lighting conditions present in portions of the image. I.e. if we took a face image with one half bright and one half dark and put it through the algorithm, as already stated above, the brighter pixels on the one half will still be brighter after the equalization is performed which still equates to an illumination problem for the eigenface algorithm. However, these are very much edge cases. For the most part illumination issues will be of the form of global lighting differences.

3.2.3 Mean Illumination Estimation

This method takes a localised smoothing approach to lighting normalization. This is done so as to remove the components of the image responsible for illumination changes. Firstly it notes that according to the Illumination-Reflection model described in detail in [13], a pixel f_{xy} in a facial image gets its value from two components. r_{xy} represents the reflection component of an image at the point (x, y) and i_{xy} represents the illumination component. Thus we get the equation:

$$f_{xy} = r_{xy} \times i_{xy} \tag{3.3}$$

Now, as r_{xy} is dependant purely on the surface material in question and not affected by illumination it would be an intrinsic representation of the facial image. Suppose i_{xy} changes little in value within a small area while in the presence of a weak light source. The key idea now is to estimate the regional value of the illumination and use this to cancel out the illumination. Thus we wish to find an estimate of our image f_{xy} that will allow us to do this separation. To attain such an estimate we apply a logarithmic transformation to each pixel in our image f_{xy} we call this new function g_{xy}

$$\begin{aligned} g_{xy} &= ln(f_{xy}) \\ &= ln(r_{xy}) + ln(i_{xy}) \end{aligned} \tag{3.4}$$

Now $\hat{g}_{xy}$ becomes a mean estimate for g_{xy} and is obtained via:

$$\begin{aligned} \hat{g}_{xy} &= \frac{1}{n^2} \sum_{(s,t) \in \omega_{nn}} g_{st} \\ &= \frac{1}{n^2} \sum_{(s,t) \in \omega_{nn}} ln(r_{st}) + \frac{1}{n^2} \sum_{(s,t) \in \omega_{nn}} ln(i_{st}) \end{aligned} \tag{3.5}$$

Note ω_{nn} is the area around a given pixel (x, y) with (s, t) being the enumeration of these pixels and n is the width/height of said kernel around (x, y). Next, the quotient image d_{xy} is constructed from equations, (2.4),(2.5) we do so to eliminate i_{xy} (or make its contribution to the image negligible):

$$\begin{aligned} d_{xy} &= g_{xy} - \hat{g}_{xy} \\ &= ln(r_{xy}) - \frac{1}{n^2} \sum_{(s,t)\in\omega_{nn}} ln(r_{xy}) + \sigma \end{aligned} \tag{3.6}$$

Where: $\sigma = ln(i_{xy}) - (\frac{1}{n^2}) \sum_{(s,t)\in\omega_{nn}} ln(i_{st})$ we note that σ will be a very small value and can hence be omitted from (4) leaving us with the relation:

$$\begin{aligned} d_{xy} &= g_{xy} - \hat{g}_{xy} \\ &\approx ln(\frac{r_{xy}}{(\prod_{(s,t)\in\omega_{nn}} r_{st})^{\frac{1}{\pi^2}}}) \end{aligned} \tag{3.7}$$

Now, d_{xy} represents the ratio between the current point's reflectance and the average reflectance around it. When the materials in the localised kernel around (x,y) are the same d_{xy} tends towards zero, but when they are different e.g. facial skin and facial features d_{xy} becomes notably none-zero. Let us consider:

$$\alpha = \frac{1}{ab} \sum_{(x,y)\in f_{a\times b}} |d_{xy}| \tag{3.8}$$

With a = number of rows and b = number of columns in an image $f_{a\times b}$. α will represent the average grey value ratio of the whole image and features. Each pixel is hence expressed as the ratio exponent relative to the average grey value:

$$h_{xy} = \exp \frac{d_{xy}}{\alpha\dot{\beta}} \tag{3.9}$$

With β being a controllable scaling factor. However, it is usually set in the range of two to three. Finally to improve contrast and correct the resulting brightness level, highlight facial features and reduce impact of background noise, post processing is done as:

$$\hat{o}_{xy} = \begin{cases} h_{xy} & h_{xy} < 1 \\ 1 & h_{xy} \geq 1 \end{cases} \tag{3.10}$$

$$o_{xy} = [\frac{255 \times (\hat{o}_{xy} - c)}{1 - c}] \tag{3.11}$$

With o_{xy} being the final result image to be used in training or recognition. c is the minimum value of $\hat{o}_{xy}$. The math may be complex but the idea is rather simple, given an image, we separate out the intensity factor from the structure of the face, setting it to zero and rebuilding the face. An example is provided below.

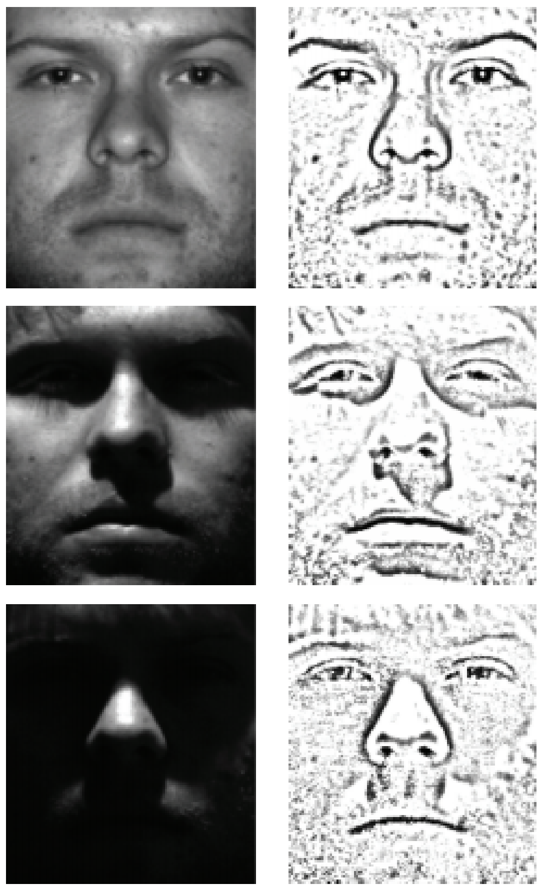

Figure 3.2: Mean Illumination Estimation

This leaves us with an image that for the most part is devoid of all extra light. Ideally, multiple images of the same object under different lighting conditions that are put through this algorithm will end up looking the same. An example of it in use can be seen below. Note how, in the above fig:[3.2], despite the drastic differences in illumination on the left side, the right side varies little from face to face.

The above images were 168x192 in size and it was discovered that for this size an n of 11 was optimal along with $\beta = 2.2$ where β is affected by facial reflectance of the subjects tested on. However, it stays at a value of 2.2 as the facial reflectance of different people varies little.

3.2.4 Summary of Mean Illumination Estimation

The complications arise from estimating the illumination component so as to subtract it from the image, highlighting the remaining reflectance of the image. We are forced to achieve this by a logarithmic transform which only slightly (computationally) affects the image, the fact remains that it still does change the image.

In conclusion, any attempt to normalize an image's illumination will undoubtedly degrade some aspects of the image we would rather retain. However, one would hope that the gained standardization of facial images out way this degradation and yield more accurate recognition rates. We also note that this method is a localised correctional algorithm. That is to say the afore mentioned issue with histogram equalization not being very effective for images with half bright, half dark faces doesn't apply here. The formula and reasoning were put forward by and learned from [10] an article that attempts to find a better way of solving the lighting issue with positive results for their effort.

Using the workstation, the task now is to gain more insight into the algorithm by attempting to duplicate the literature results and to further validate the strengths and weaknesses of the algorithm.

3.2.5 Validation of MIE implementation

Before the experiment can be run we first attempt to validate that aspects of the system work as intended. This is done to provide validation to the results obtained from the experimental procedure. Though we predominantly wish to do so for our implementation of the Mean Illumination Estimation, for completion, we also confirm OpenCV's built in Histogram Equalization method works as intended.

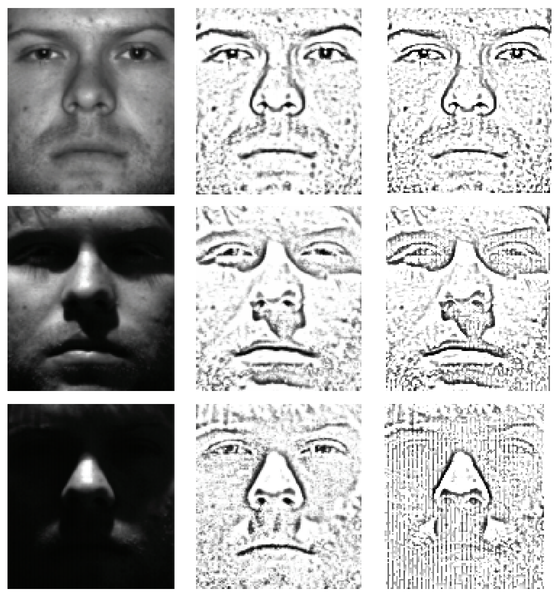

Figure 3.3: My MIE implementation [col 3] tested against provided example from [10] [col 2]

To start, we compare the output my MIE implementation provides against that of the output attained according to [10]. We note above in fig:[3.3];

We observe that my implementation provides near identical results to that of the article for the base face. However, as the images become darker our results seem to deviate more from those in the published paper. We postulate that these artefacts are being generated from the conversion between image to PDF, and back to image again i.e. slight smudging is introduced in the dark regions from the conversion algorithm, possibly from compression algorithms.

To confirm our hypothesis, here are several more images put through my algorithm that are not extracted from the paper of [10] PDF. We see none of the below show similar artefacts to those found in fig:[3.3].

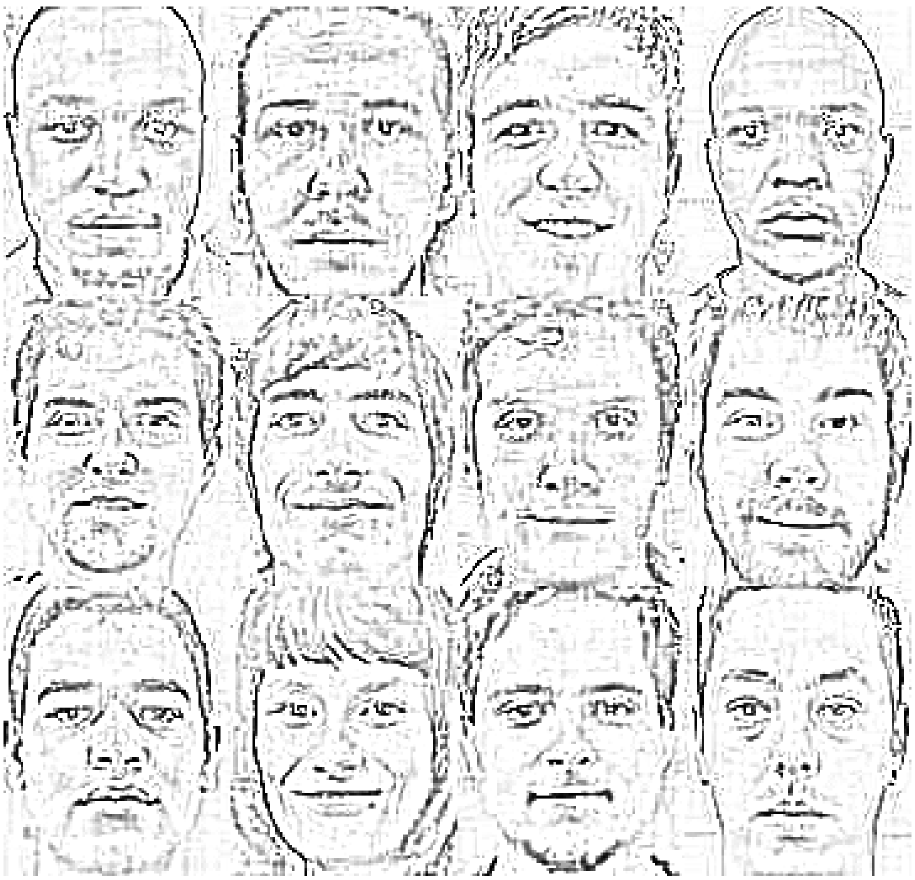

Figure 3.4: MIE implementation applied to CSC Honours 2015 Class list, Rhodes University

To further investigate, we took the original PDF images and ran them through the OpenCV implementation of histogram equalization Below fig:[3.5], we see the same faces as The MIE comparison test fig:[3.3] above. However, instead of the MIE applied to them they have undergone histogram equalization.

Figure 3.5: Histogram Equalization applied to article example faces

We see that the intensities have been stretched for all images, improving visibility, We also note the far right image, once again the same artefact pattern emerges. This points again to an issue with the image itself, not the algorithms performed upon them. Now that we have confidence in our implemented code we can construct an experiment to test the effects on accuracy they have.

3.3 Reading The Graphs

The algorithm "classifies" a face provided the matcher gets a "deviation" score less than the provided threshold. Therefore at each threshold point we get 3 regions; going up; the first region defines the number of successfully classified images, the second region shows the number of unsuccessful classifications and the remaining portion of the graph are those images that were not yet classified under the current threshold but may still be under the next threshold level.

3.4 Experiment 01: Setup

First up we will test our system against the AT&T database, this database contains 40 subjects with 10 images per subject totalling 400 images. We have evenly split this data into training and test data i.e. 5 images per subject goes to training and the other five to test data. The results we will be comparing are the total number of correct connections made by the eigenface algorithm. This database primarily focuses on orientation of the subjects faces and has limited illumination variance, as such, we expect the eigenface algorithm to perform with some degree of accuracy and the application of our two normalisation techniques to have minimal if any effect on the accuracies obtained.

3.4.1 Experiment 01: Results

We start with a base line, no normalization techniques used, just the plain dataset run through the eigenface algorithm. Doing so we get the graph;

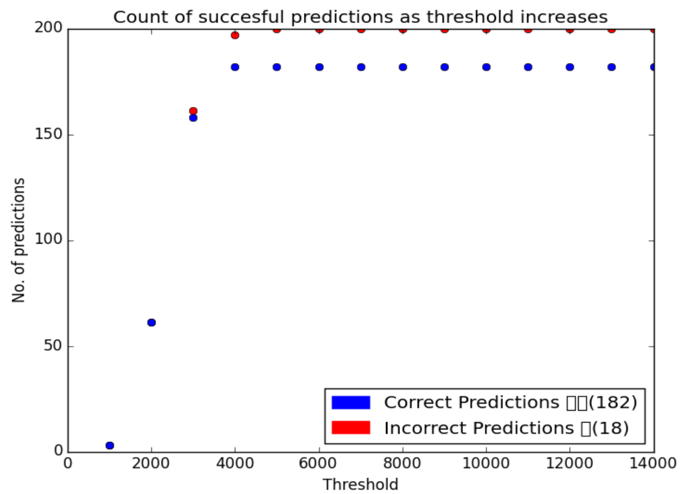

Figure 3.6: Base line, no illumination normalization used; AT&T

From Figure: 3.6 we see that we correctly recognize $\frac{182}{200}$ subject images i.e. equating to a 91% accuracy. We run the experiment again but this time perform histogram equalization upon both test and training data. This provides us with the graph;

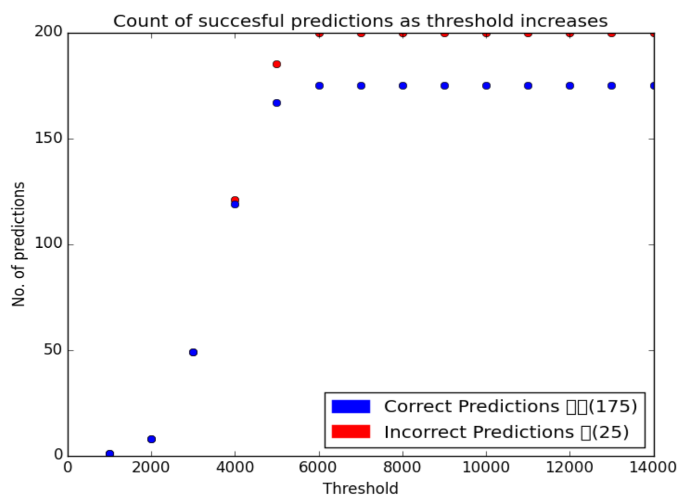

Figure 3.7: Histogram Equalization experiment; AT&T

We now see from Figure: 3.7 that the system can correctly recognize $\frac{175}{200}$ of the subject images, roughly 87% accuracy rating, a slight drop from the base line but still a viable accuracy.

We now run the experiment again. However, now we test our MIE algorithm. Doing so, we obtain the results;

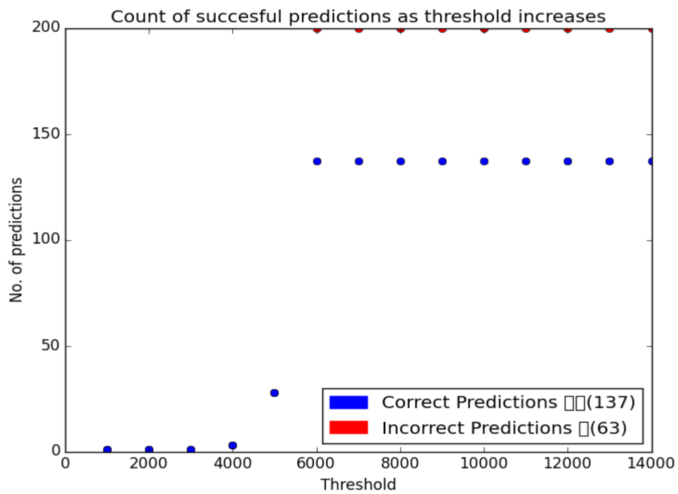

Figure 3.8: MIE experiment; AT&T

Here we see greatly diminished results with, $\frac{137}{200}$ recognized faces, equating to roughly 68% accuracy. Once again, we try two more experiments, 1.) we run histogram equalization first, then MIE upon this normalised data. 2.) the other way round, test MIE first, then histogram equalization;

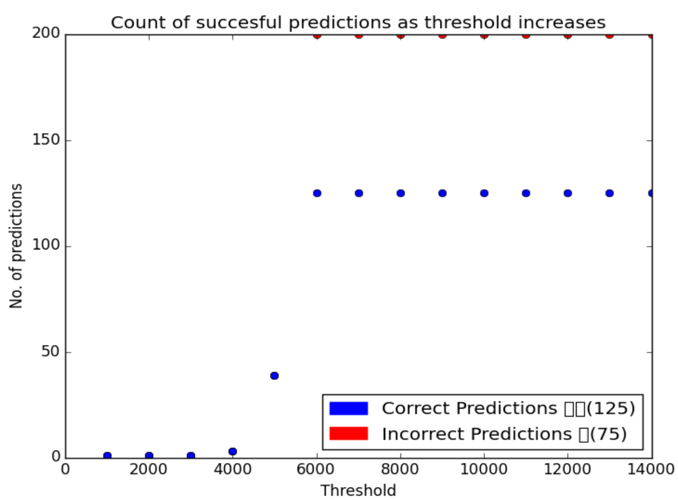

Figure 3.9: Histogram then MIE; AT&T

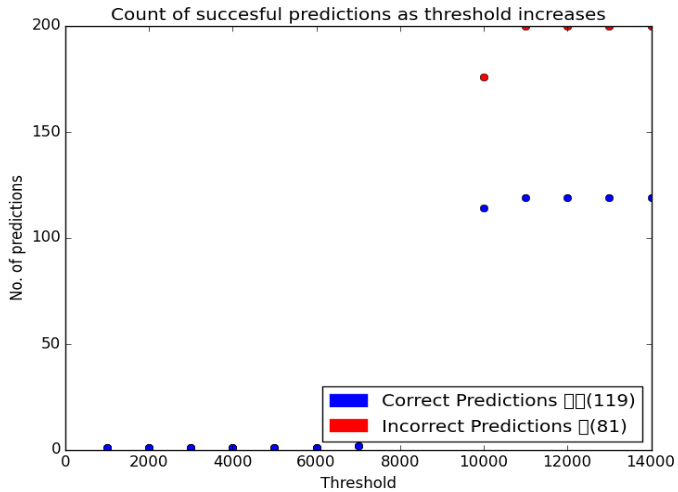

Figure 3.10: MIE then Histogram; AT&T

Aside from the over all inaccuracy of the combination we see a connection emerge; histogram equalization then MIE performs in accordance with MIE on its own if slightly less accurate but the other way round provides far worse results.

3.4.2 Experiment 01: Discussion

Initially we observed what we expected, the baseline performed with a high degree of accuracy due to the lack of illumination change in the images. However, to our disappointment when we run our correction algorithms we see diminished results with MIE being completely impractical for normal use.

3.5 Experiment 02: Setup

For this experiment the conditions were; Yale B face database [14] which contains 38 individuals with 65 images per subject (2470 total). However, to reduce computation time I will only used a subset of the data, the first 20 subjects. Note, the eigenface algorithm cannot handle images of varying size, so many of the subjects have images that need to be discarded as they are larger than the others. Also, some of the images were provided as corrupt, presumably as further testing data, they too have been removed. Furthermore we also take 10 images per subject to use as the training set. What remains is 20 subjects and 1072 images as testing data. The results we will be comparing are the total number of correct connections made by the eigenface algorithm. based on the literature and as noted above, we expect MIE to surpass OpenCV's Histogram equalizations recognition accuracy.

3.5.1 Experiment 02: Results

As before we start with a base line, no normalization techniques used, just the plain dataset run through the eigenface algorithm. Doing so we get the graph:

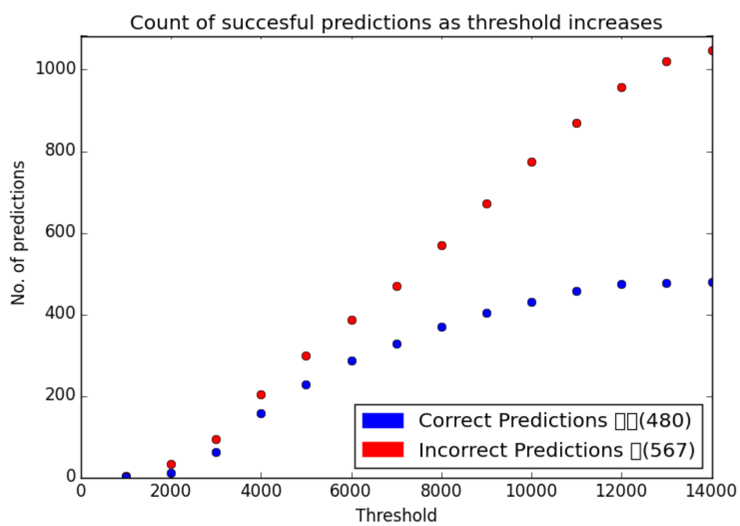

Figure 3.11: Base line, no illumination normalization used; Yale B

From Figure: 3.11 we see that we correctly recognize $\frac{480}{1072}$ subject images i.e. roughly 45% accuracy. We run the experiment again but this time perform histogram equalization upon both test and training data. This provides us with the graph;

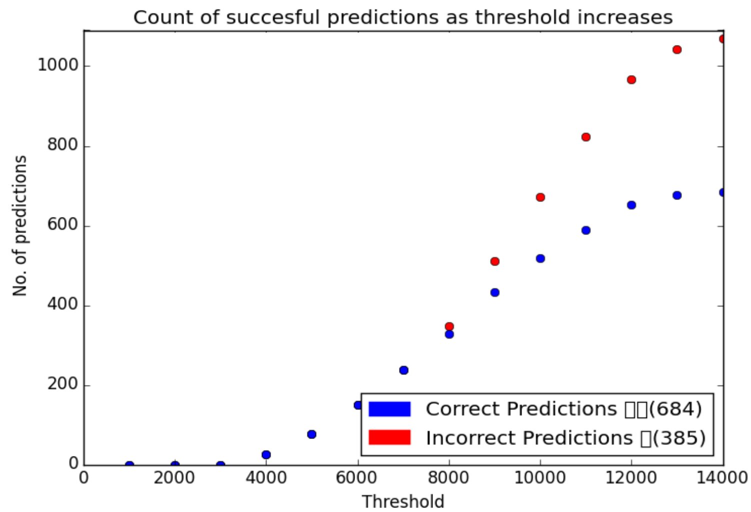

Figure 3.12: Histogram equalization experiment; Yale B

We now see from Figure: 3.12 that the system can correctly recognize $\frac{684}{1072}$ of the subject images, roughly 64% accuracy rating, a meaningful improvement on the base line.

We now run the experiment again. However, instead of Histogram Equalization, we test our MIE algorithm. Doing so, we obtain the results;

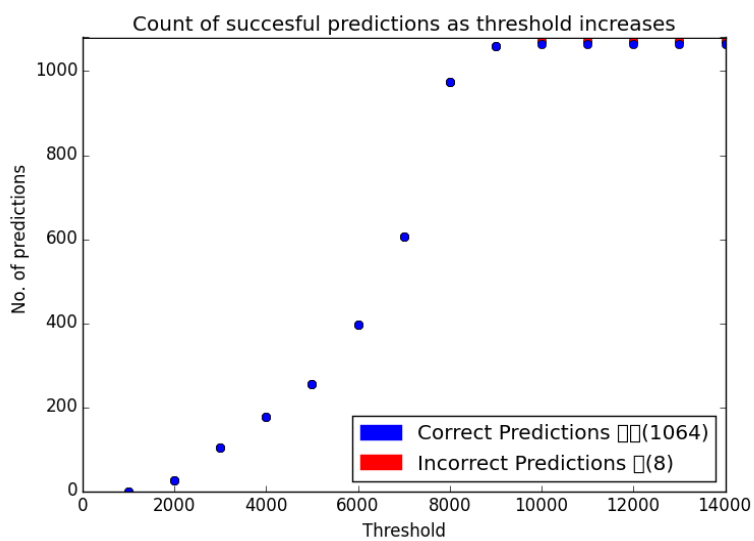

Figure 3.13: MIE experiment; Yale B

Here we see greatly improved results with, $\frac{1064}{1072}$ recognized faces, equating to roughly 99% accuracy.

In a final attempt to yield even better results, we performed two more experiments, 1.) we run histogram equalization first, then MIE upon this normalised data. 2.) the other way round, apply MIE first, then apply histogram equalization:

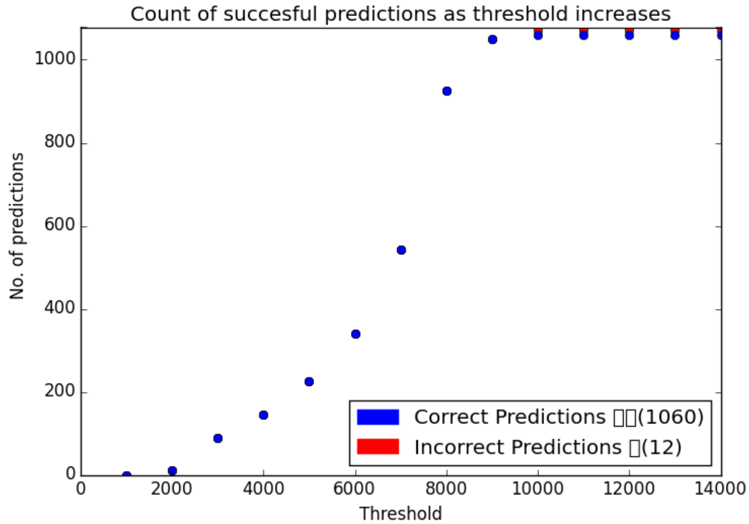

Figure 3.14: Histogram then MIE; Yale B

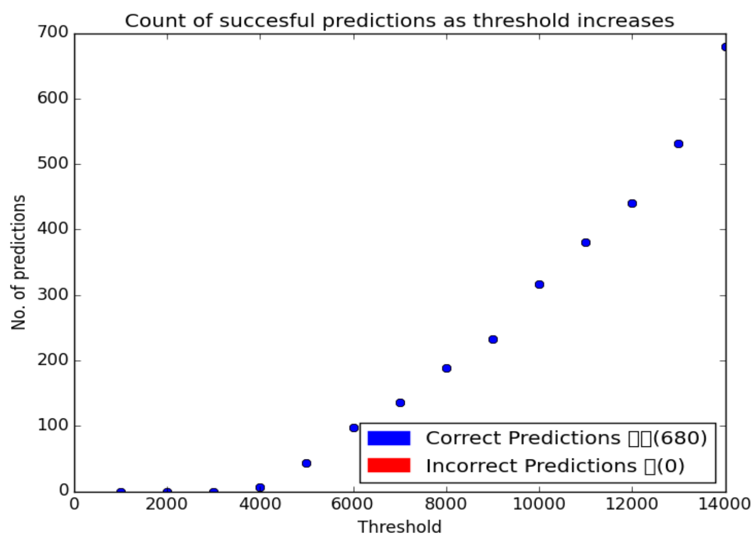

Figure 3.15: MIE then Histogram; Yale B

We see from these experiments that Performing Histogram Equalization first and then MIE on top of that performs nearly as well as MIE on its own, but the other way round yields shocking results, failing to even score all images under the maximum 14000 threshold.

3.5.2 Experiment 02: Discussion

The Baseline experiment, fig:[3.11] provides a very poor score with this facial database, this was to be expected due to the high impact of lighting conditions on the eigenface algorithm, yielding a score of $\frac{480}{1072}$, roughly 45% accuracy. On top of that, it technically doesn't even score all the images, 3 where missed i.e. their threshold values were higher than the upper limit of our graph.

The first experiment we try is histogram equalization. Here we see an improvement in recognition at roughly 64%. We already know from the explanation of the workings of histogram equalization that it is not ideal for the eigenface method when used on images with localised lighting conditions, something that the Yale B database has in abundance. This aside, all in all, it is a notable improvement on the none-normalised score but still far from the ideal.

Next we hope to achieve better results with the MIE version of the data. We get results which far surpass those provided by histogram equalization, at roughly 99%. This is because the MIE is a very localised oriented algorithm and as the Yale B database has nearly insignificant change in orientation and angle, purely lighting changes, the MIE performs well. As such, these results most certainly agree with the claim published in [10].

With regards to the compound tests I deduce the reason for such discrepancy between the order is that performing histogram equalization first outputs a very lightly affected image with just a simple contrast change that MIE can operate on in much the same way as before, while trying to do it the other way, MIE outputs a vastly different image from the original, more resembling a sketch than a photo, with large variation in colour intensities, as such the histogram equalization algorithm falters as we have reduced the colour space to an even more compact region of the spectrum. Thus when we pass these new images through the histogram equalization method it just does more damage by stretching it further.

What we take away from this is that although MIE performs phenomenally against strong illumination conditions it tends to degrade the image it is working on to an extent that other normalization techniques are ineffective at best and further degrading at worst.

In reality we probably wouldn't want to perform multiple normalization techniques that target the same issue upon our dataset as we would expect such conflicts to arise. On the other hand we would hope that combining these techniques with others that target different aspects of normalization would not conflict with each other. Indeed, we would hope that doing so would provide a higher degree of accuracy in recognition.

3.6 Summary

To conclude, yes, MIE looks good, but only for a narrow category of normalization problems namely, those with localized regions of lighting variation on that note, I find it hard to believe any algorithm could improve upon those accuracies. Sadly that's where the downsides must be mentioned. The algorithm doesn't play well with other techniques and even less so when there are no local illumination issues present, tending to be more of a hindrance than a help. Taking note of this, for our implementation in classroom attendance tracking, we have a semi-controlled environment where lighting conditions will be more along the lines of uniform differences across our subject faces. As such OpenCV's histogram equalization would be the algorithm of choice for our needs.

The Python implementation of MIE was horribly time inefficient. As a benchmark, OpenCV's histogram equalisation was performed upon the 1272 images and did so in under a minute while my Python implementation of MIE took roughly 10hrs to do the same number of images. Again, this speed difference comes with the benefit of being a very localised method. Each pixel is compared to a 11x11 kernel around itself as well as the average intensity for the whole image. This means it can handle images which are dark on one side but light on the other without problems.

It was beyond the scope of this work to attempt to optimise this method. However, we do note the following possible optimisations; 1.) Converting the MIE code logic into c++ or c that is then wrapped into Python, doing so could offer substantial speed-ups 2.) The current implementation is fairly naive: the 11x11 region is recomputed for each pixel in the image. Many efficient image processing algorithms operate by sliding a difference window over the image and computing a running estimate of local intensity in our case we would reduce the 121 computations to 22 per pixel. This would most certainly improve the performance of the method. Finally, this per pixel kernel methodology indicates that this method could see a large speed-up if performed through a GPU. However, these are ideas for future work to look into.

Chapter 4

Assignment Problem

4.1 The Problem

Now that we have determined a sensible illumination normalization technique to use, we turn our attention to another means of improving recognition rates. When we consider the problem we are trying to solve, namely facial recognition for classroom attendance tracking, we realise that the current method of classification used by the eigenface method doesn't make any use of the intrinsic properties of our problem.

We know exactly who and how many subjects should present in each lecture from the get go. Knowing this, we can in theory improve our recognition rates by doing away with eigenfaces current greedy, first match algorithm and in its place use a global assignment algorithm like the Hungarian Algorithm which we shall look into in the following sections.

In its current state the provided eigenface algorithm makes use of a greedy first match algorithm, ~~that has the benefit of speed~~. When you don't know the sets you are dealing with and weather or not the face you are testing is even in your data set, it is a ~~perfectly~~ adequate solution, It will take a test face and try find the best match it can in its training data and ~~using the confidence value outputted you decide if you should believe what it's telling you.~~

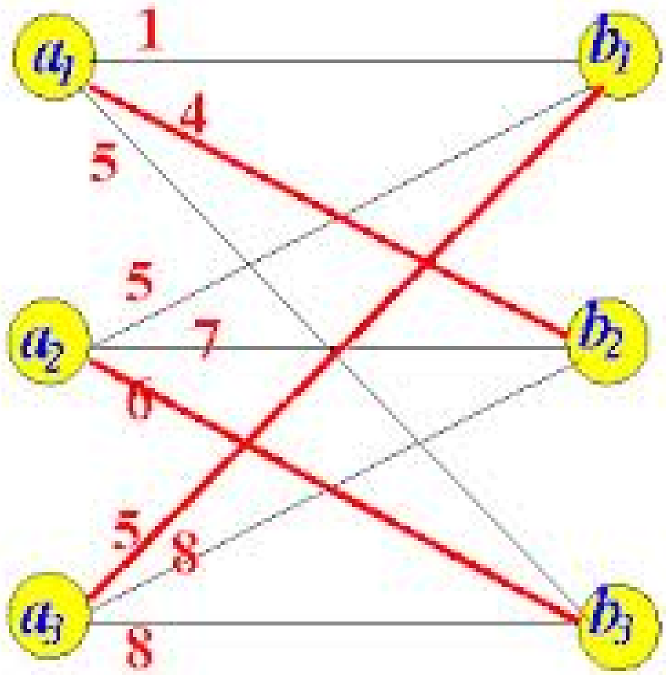

Figure 4.1: The Assignment problem

However, we do know the sets and are sufficiently confident that every face we test is in our training data. Thus we can use this knowledge by making use of a multiple assignment optimisation algorithm as seen in fig:[4.1]. Such an algorithm scores every test face against all training faces, thereafter, optimise the classification of who's who to minimize the total score of the entire classification.

4.2 Hungarian Algorithm

The Hungarian Algorithm is a proposed solution to the assignment problem that is solved by minimizing ~~some element,~~ as we work with a quantity that is better the smaller it is the solution fits rather well.

In order for the algorithm to work it needs to be able to build up a square matrix ie the two sets you wish to match must be of the same length. With this in place we would score the data, putting the confidences attained in an output matrix $m_{n \times n}$ ow the problem involves choosing a combination of scores such that the total combination of these scores is as small as possible.

We could do this by hand for small sets, but the problem quickly escalates as the sets become bigger in fact the problem done this way is of O(n!) not ideal. However at ~~our solutions~~ core is the theorem: If a number is added to or subtracted from all of the entries of any one row or column of a cost matrix, then an optimal assignment for the resulting cost matrix is also an optimal assignment for the original cost matrix [15].

The algorithm from [15];

1. firstly, subtract the smallest entry in each row from all the other entries in that row.

2. do the same for each ~~row~~

3. select rows and columns so that all zero elements are selected with the minimum number of selections

4. (I) if the number of selections is n, then we are done, we have found the optimal assignment (II) if we have less than n selections an optimal selection is not yet possible, so we proceed

5. determine smallest entry not yet covered by any line, subtract this value from each unselected row and add it to each selected row return to step 3

So, mathematically, we can manipulate along rows or down columns by the same operation without it affecting the integrity of the matrix. We know this thanks to the afore mentioned ~~T~~heorem the proof of which is beyond the scope of this work. However, we have another aspect to look at, how exactly do we find the minimum selection of rows/columns that contain all our zeroed elements? One way this can be done is as follows;

~~given:~~

0	a2'	a3'	a4'
b1'	b2'	b3'	0
0	c2'	c3'	c4'
d1'	0	d3'	d4'

Figure 4.2: selection of rows & columns step 1

- First, select as many rows/columns as possible.

0'	a2'	a3'	a4'
b1'	b2'	b3'	0'
0	c2'	c3'	c4'
d1'	0'	0	d4'

Figure 4.3: selection of rows & columns step 2

- Now, mark all rows having no assignments
- Next, mark all unmarked columns having zeros in newly marked rows
- mark all rows having assignments in newly marked columns
- Repeat for all non-assigned rows

Doing the above we end up with:

x				
0'	a2'	a3'	a4'	x
b1'	b2'	b3'	0'	
0	c2'	c3'	c4'	x
d1'	0'	0	d4'	

Figure 4.4: selection of rows & columns step 3

- Finally make your selections on all marked columns and UNMARKED rows.

x				
0'	a2'	a3'	a4'	x
b1'	b2'	b3'	0'	
0	c2'	c3'	c4'	x
d1'	0'	0	d4'	

Figure 4.5: selection of rows & columns step 4

This is just one way of making the minimum number of selections. The above method was learned from wikepedia.

4.3 Summary

This chapter looked into the possibility of using a global optimisation technique for assignment of two sets. We predict that doing so will improve accuracy of the recognition system used by the system. We focused on the Hungarian algorithm as the implementation option detailing the steps needed to make it work. also noting that eigenfaces method of scoring lends itself nicely to such a minimising solution.

Sadly we cant actually implement such a system as we ~~cant~~ force the eigenface method to score each test data against all training vectors. Hence this is proposed as future work for when such manipulation is provided by the OpenCV library.

Chapter 5

Conclusion

5.1 Summing up

In this work we looked into the viability of creating a facial recognition system with the purpose of tracking classroom attendance. Doing so, we discovered that there are many ways to improve recognition rates of such a system both from general normalization methods, as well as ways to use the nature of the problem at hand to help in accuracy.

The second aspect we noticed was that there was no readily available system that one could test new ideas against existing solutions. At this point, the purpose of this work changed to creating a workstation from which a user could add a new normalization technique (or theorised improvement of an already existing method) and compare results to past solutions to the targeted problem.

5.1.1 Working Example

As an illustration of the proposed usage of the system this work looked into the viability of the Mean Illumination Estimation in conjunction with the eigenface method of facial recognition both in general and specifically for our needs in classroom tracking. The experiment probed from two directions, one; we compared it against OpenCV's inbuilt histogram equalization solution to illumination issues. Two; we ran the experiment twice, one against the Yale B database and again against the AT&T database.

The results of this experiment have already been discussed as can be seen in chapter 3. However, a short run-down of the highlights will follow. MIE works really well when up against strong illumination conditions; directional lighting, strong global change in ambient light etc. However, it greatly degrades the quality of the image worked on meaning it wouldn't work well with a multiple normalisation technique solution.

It was noted that OpenCV's histogram equalization would suit our purposes more adequately than MIE as we deal more with global ambient change as opposed to local lighting changes and histogram equalization does not have as much of an effect as MIE on the images processed.

5.1.2 Theory of future work

Another aspect to this work was to look into the benefits of using a global optimisation technique that would replace eigenfaces greedy first match classification system. This was looked into in Chapter 4; Due to limitations with the C++ implementation of OpenCV's Eigenface algorithm, such manipulation of its inner workings is currently not available to us. However, it has been stated in an answer by the creators to a similar request [16] that such a capability could be implemented at a later stage.

Going with this, this work explores how such data could be exploited to yield better accuracies from systems where both the training set and test set are known (or should be known). Having looked specifically at the Hungarian algorithm as it offers a solution in polynomial time [**?**] as opposed to factorial based time that the problem is inclined towards.

5.2 Looking ahead

This work is such that it will never truly be finished, one can always add more normalization techniques for comparison, or add different recognition algorithms to classify test faces.

Strong candidates for future work are solutions to the assignment problem brought up previously as well as alignment normalization techniques. Both of these are pressing issues that will need to be taken into account if one wishes to create an extensive recognition system.

Appendix A

Appendages

Bibliography

[1] U. A. Patel and D. S. P. R, “Development of a student attendance management system using rfid and face recognition: A review,” *International Journal of Advance Research in Computer Science and Management Studies*, vol. 2, no. 8, pp. 109–119, 2014.

[2] W. A. Naveed Khan Balcoh, M. Haroon Yousaf and M. I. Baig, “Algorithm for efficient attendance management: Face recognition based approach,” *IJCSI International Journal of Computer Science Issues*, vol. 9, no. 4, pp. 146–150, 2012.

[3] M. M. D. J. Mr. C. S. Patil, Mr. R. R. Karhe, “Student attendance system and authentication using face recognition,” *International Journal of Engineering Research and Technology (IJERT)*, vol. 3, no. 7, pp. 373–375, 2014.

[4] A. K. Gary Bradski, *Learning OpenCV Computer Vision with the OpenCV Library.* 1005 Gravenstein Highway North, Sebastopol, CA 95472: O’Reily Media, Inc., 2008.

[5] S. V. ”R. Lagerwall, “”plant classification using leaf recognition”,” *”University of KwaZulu-Natal”*, pp. ”91–95”.

[6] D. University, “http://www.pages.drexel.edu/ sis26/eigenface03 May 2015.

[7] M. Turk and A. Pentland, “Eigenfaces for recognition,” *Journal of cognitive neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[8] M. Lorenco., “Part 01 for project: Facial recognition for classroom attendance tracking,” 2015.

[9] M. K. Hasan and C. J. Pal, “Improving alignment of faces for recognition,” in *Robotic and Sensors Environments (ROSE), 2011 IEEE International Symposium on*, pp. 249–254, IEEE, 2011.

[10] Y.-P. G. Yong Luo and C.-Q. Zhang, “A robust illumination normalization method based on mean estimation for face recognition,” *ISRN Machine Vision*, vol. 2013, no. 516052, pp. 0–10, 2013.

[11] “Digital multi-media design.” `http://dmmd.net/main_wp/intuitive-mathematics/histogram-equalization/`, 15 Oct 2015.

[12] “University of california.” `http://www.math.uci.edu/icamp/courses/math77c/demos/hist_eq.pdf`, 15 Oct 2015.

[13] J. Ho, B. V. Funt, and M. S. Drew, “Separating a color signal into illumination and surface reflectance components: Theory and applications,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 12, no. 10, pp. 966–977, 1990.

[14] K. Lee, J. Ho, and D. Kriegman”, “”acquiring linear subspaces for face recognition under variable lighting ”,” *”IEEE Trans. Pattern Anal. Mach. Intelligence”*, vol. 27, no. 5, pp. ”684–698”, 2005.

[15] “Harvard university.” `http://www.math.harvard.edu/archive/20_spring_05/handouts/assignment_overheads.pdf`, 16 Oct 2015.

[16] “Opencv.” `http://answers.opencv.org/question/3494/adding-new-method-in-facerecognizer/`, 19 Oct 2015.

[17] J. Munkres, “Algorithms for the assignment and transportation problems,” *Journal of the Society for Industrial and Applied Mathematics*, vol. 5, no. 1, pp. 32–38, 1957.