Project: Literature Review

Matthew Shane Kingon

April 27, 2015 – May 27, 2015

Supervisor:
Prof. Peter Wentworth

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 an attempt is made to make use of additional information available.

Such additional information include 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 this project tries to take advantage of is the knowledge that prior to lectures it is already known whom 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 Specific Fields

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 The Big Picture

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.)

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 you wish to perform more complex arithmetic procedures you need to take into account and obey their 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.

4 Eigenface Algorithm

4.1 Intuitive description

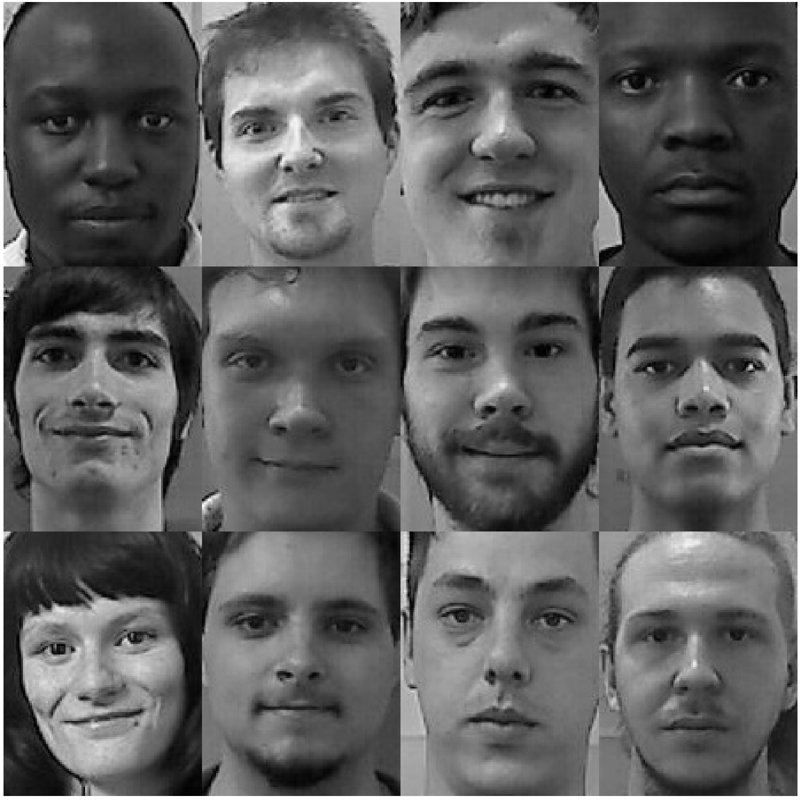
The Eigenface method of facial recognition works by taking the high dimensional face images represented mathematically as an $m \times n$ matrix, Providing it with N such images it takes them and finds the average of the matrices(images) i.e. sum them together pixel by pixel and divide by N. With this "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 to best represent these "difference faces".

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. Indeed, given the training image you should be able to get coefficients that reconstruct the original face exactly.

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 1:

Figure 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:

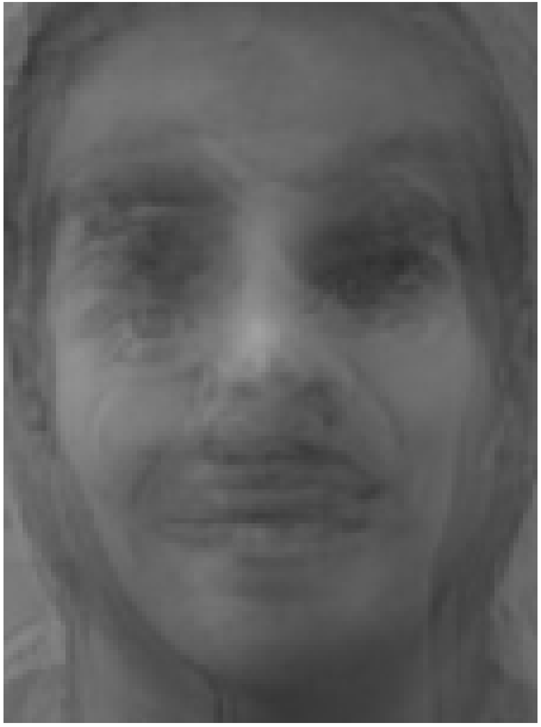
$$S = \Gamma_1, \Gamma_2, \Gamma_3, ..., \Gamma_N$$

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:

Figure 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 (3) above shows this, each facial image below maps to the corresponding input face as was shown in Figure (1) minus the average face which was displayed in Figure (2).
We next obtain the covariance matrix C which we need for its Eigenvectors/values (μ, λ) respectfully. We obtain C via:

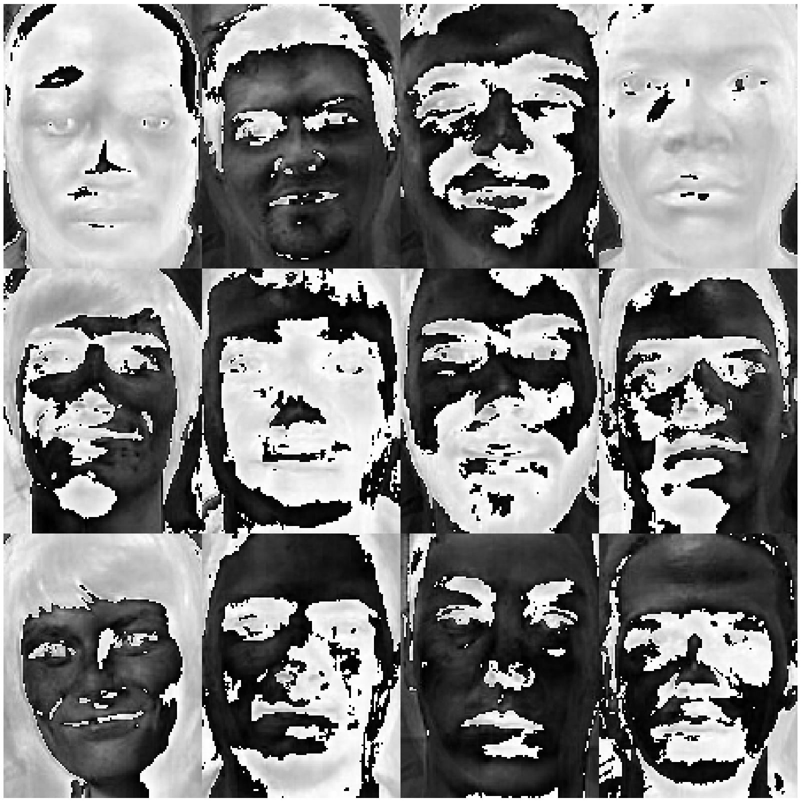
$$C = \frac{1}{N} \sum_{n=1}^{N} \phi_n \phi_n^T$$

$$= AA^T$$

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

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

Figure 3: The "Ghost" set created via subtraction of each face from the mean.



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.

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 to large the algorithm will generate many false positives. If the image is a face but unknown, then you could choose to add it into the set of known faces and repeat the training steps.

4.4 Summary of EigenFaces

So in summary, you provide the Eigenface Algorithm with a set of face images to train on, then once it is trained you give it 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 images 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. 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 [5].

5 Normalization

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 [6]. 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.

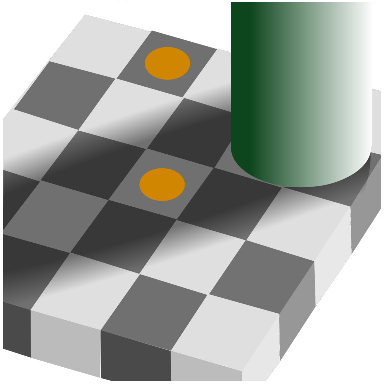
Preliminary testing does indeed show that cropping an image has an effect on the recognition rate. However, these have been manual crops that do not resize the image nicely. Also, some faces are simply not well cropped with a lot of background left in the image. Another improvement that can be attempted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

5.2 Lighting

Lighting plays a very big role in all facial recognition, identification or just about any image processing problem. As even though, to the human eye it is almost undetectable, to a computer that looks intently at each and every pixel, even the slightest change in lighting makes each pixel value change dramatically.

To better appreciate the problem it should be noted that there are an abundance of illusionary images that fool the human brain. The example shown in 4 may be an old one, but it does illustrate the point, the two blocks with the orange dot are in fact, the same shade of grey. 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 darker than the above as it has a shadow cast upon it.

Figure 4: Optical Illusion example



One approach to overcome such lighting issues is to use the Mean Illumination Estimation of an image and then train/recognize with this new, normalized, image.

5.3 Mean Illumination Estimation

This method takes the smoothing approach to lighting normalization. This is done so as to remove the layer of the image responsible for illumination changes. Firstly it notes that according to the Illumination-Reflection model described in detail in [7], a pixel in a facial image f_{xy} gets its value from two parts. 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:

$$f_{xy} = r_{xy} \times i_{xy} \tag{1}$$

Now, as r(x,y) 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(x,y) changes little in value within a small area while in the presence of a weak light source. Thus we wish to find an estimate of our image f(x,y) that will allow us to do this separation. To attain such an estimate we apply a logarithmic transformation to our image f(x,y) we call this new function g(x,y)

$$\begin{aligned} g_{xy} &= ln(f_{xy}) \\ &= ln(r_{xy}) \times ln(i_{xy}) \end{aligned} \tag{2}$$

Now the mean estimate for g_{xy} is obtained as:

$$\begin{aligned} \hat{g}_{xy} &= \frac{1}{n^2} \sum_{(s,t)\in\omega_{nn}} g_{st} \\ &= \frac{1}{n^2} \sum_{(s,t)\in\omega_{nn}} ln(r_{st}) + \frac{1}{n^2} \sum_{(s,t)\in\omega_{nn}} ln(i_{st}) \end{aligned} \tag{3}$$

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). The Quotient image is constructed from equations, (2),(3) we do so to eliminate i_{xy} (or make its contribution to the image negligible):

$$\begin{aligned} d_{xy} &= g_{xy} - \hat{g}_{xy} \\ &= ln(r_{xy}) - \frac{1}{n^2} \sum_{(s,t)\in\omega_{nn}} ln(r_{xy}) + \sigma \end{aligned} \tag{4}$$

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:

$$\begin{aligned} d_{xy} &= g_{xy} - \hat{g}_{xy} \\ &\approx ln(\frac{r_{xy}}{(\prod_{(s,t)\in\omega_{nn}} r_{st})^{\frac{1}{\pi^2}}}) \end{aligned} \tag{5}$$

now, d_{xy} represents the ratio between the current points reflectance and the average reflectance around it. when the materials 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:

$$\alpha = \frac{1}{ab} \sum_{(x,y)\in f_{a\times b}} |d_{xy}| \tag{6}$$

with $a =$ No. of rows and $b =$ No. of columns in an image $f_{a\times b}$. α will represent the average grey value ratio of the facial skin and features the global difference is hence reduced as:

$$h_{xy} = \exp \frac{d_{xy}}{\alpha\dot{\beta}} \tag{7}$$

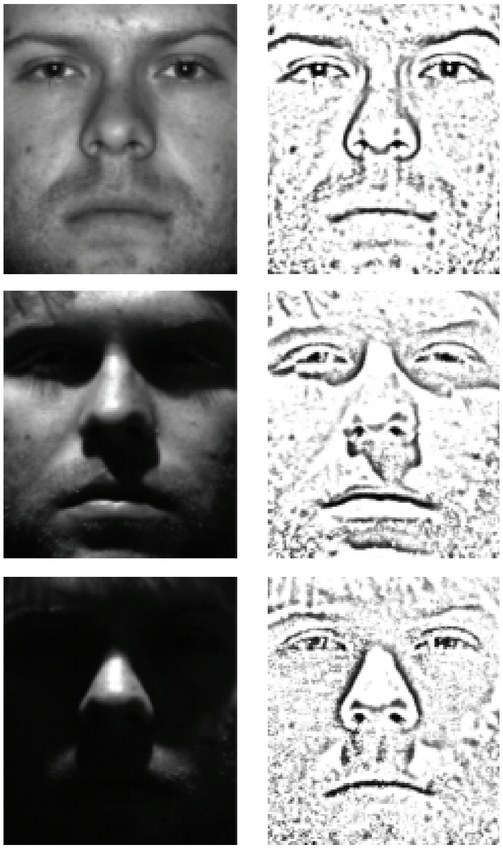
with β being a controllable scaling factor. However, it is usually set in the range of 2-3 thus, finally to highlight facial features and reduce impact of background noise, post processing is done as:

$$\hat{o}_{xy} = \begin{cases} h_{xy} & h_{xy} < 1 \\ 1 & h_{xy} \geq 1 \end{cases} \tag{8}$$

$$o_{xy} = [\frac{255 \times (\hat{o}_{xy} - c)}{1 - c}] \tag{9}$$

With o_{xy} being the final result image to be used in training or recognition. and 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. 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, despite the drastic differences in illumination on the left side, the right side varies little from face to face.):

Figure 5: Mean Illumination Estimation



5.4 Summary of Mean Illumination Estimation

The complications arise from the converting from a simple image f_{xy} into two separate images r_{xy} namely the facial structure/texture map and i_{xy} the intensity with the goal of removing i_{xy} from the equation. We are forced to achieve this by a logarithmic transform which only slightly 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. The formula and reasoning were put forward by and learned from [8] an article that attempts to find a better way of solving the lighting issue with positive results for their effort.

5.5 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 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 is 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. Sadly unless you intend to do 3d modelling of a head a full frontal facial image will be required. This has proven to be a major problem to the field.

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.

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.

This 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 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.

References

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

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

[7] 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.

[8] 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.