COMPUTER VISION

SPRING 2011

BINH LE
BENJAMIN SOIBAM
CHARU HANS
FATEMA DALAL
JOSEPH MATHEW
SON HOANG
VARUN MAHESHWARI

PROJECT REPORT

Automated assessment of VISA Photo requirements

Contents

OBJECTIVE 3
INTRODUCTION 3
DATASET 5
EXPOSURE ANALYSIS 6
INTRODUCTION: 6
PROPOSED METHOD: 6
DISCUSSION: 10
BLUR ANALYSIS 11
INTRODUCTION: 11
PROPOSED METHOD: 12
DISCUSSION: 14
EYE PUPIL DETECTION AND ANALYSIS 15
INTRODUCTION: 15
PROPOSED METHOD: 16
DISCUSSION: 20
FACE ORIENTATION AND HEAD COVERING 22
INTRODUCTION: 22
PROPOSED METHOD: 23
DISCUSSION: 24
OPEN/CLOSED MOUTH DETECTION 25
INTRODUCTION: 25
PROPOSED METHOD: 26
DISCUSSION: 29
RESULTS 30
CONCLUSION 32
REFERENCES 33

OBJECTIVE

The objective of this project is to identify acceptable photographs that are used for passport and visa applications.

The acceptable photographs for a passport photo have strict specifications which have to be adhered. This is achieved by processing the images through various analyses testing software created.

The analyses testing software that have been incorporated within this project are

1. Under and Over Exposure
2. Blur
3. Open / Closed Eyes
4. Facial Orientation (Tilting of the head)
5. Head dress / Scarfs Detection
6. Open / Closed Mouth

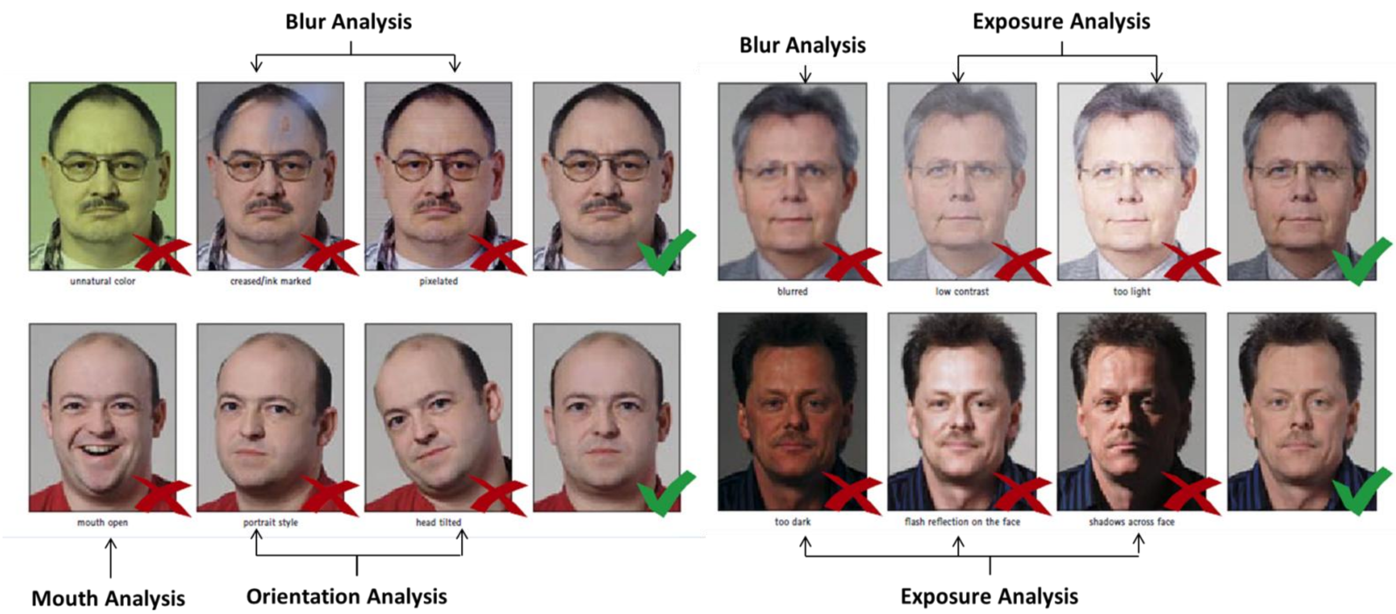
INTRODUCTION

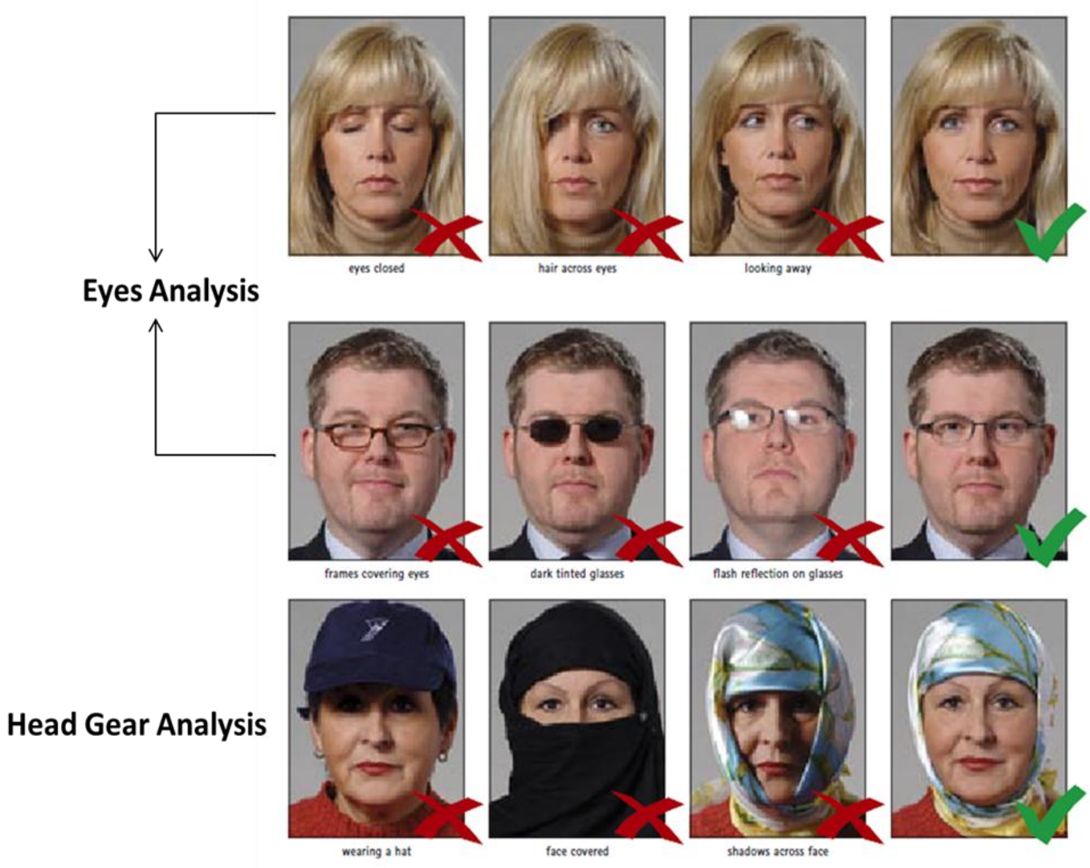
Passport photographs are required by every individual for getting their countries passport. This is the highest official document held by any individual which is also a valid document accepted by all the other countries.

Each individual country has its own specifications for the passport photographs. For example in the United States the specification for the passport image is as follows. Its size must be 2 X 2 inches, taken in color with a white or a light colored background and have no head coverings unless for religious purposes. In Europe the size specifications vary but otherwise the general criterion remains the same.

Thus, in this project the general criterion for a passport is considered for automatically rejecting or accepting an image based on its computed performance on the software created in this project.

Shown below are the passport photograph criteria from the German Embassy. Also shown is the type of analysis that is performed by the project to reject the photographs.





DATASET

We performed analysis on 2 data sets

The first data-set had 683 face images. These images had different exposure, lighting condition, tilted faces and closed eyes and open mouth.

The next of data-set we took was real-time. We took 10 images in this dataset under different lightening conditions, orientations and blur effects.

EXPOSURE ANALYSIS

INTRODUCTION:

In visa photographs we predominantly concentrate on the orientation, appearance and exposure of the face. For visa photographs the exposure of the face should be correct and uniform. Similarly for the background, the exposure of light should be uniform and there should not be any shadows. On the other hand if the photograph (face) is overexposed or underexposed or even the background is not correct then the photograph is rejected. For example in the photographs given below:

Fig 1.1 Image 1

Fig 1.2 Image 2

Fig 1.3 Image 3

Image 1 is not correct because the face is overexposed; in Image 2 the background has lots of shadows so it is wrong. In Image 3 the exposure level of the image and background is correct so this is considered right in terms of exposure.

PROPOSED METHOD:

In this project to determine the exposure of the face in the image we have used the image histograms. We first use the viola-jones face detector to find the coordinates of the facial region as shown in Fig 1.4.

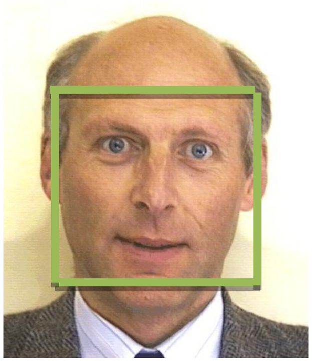


Fig 1.4

After that we build the histogram of the region within this bounding box. Then the image histogram is divided into five regions. The regions on the left correspond to dark regions and on the right corresponds to brighter regions. Therefore a well exposed image will have histograms in which majority of the intensity values will be in the middle.

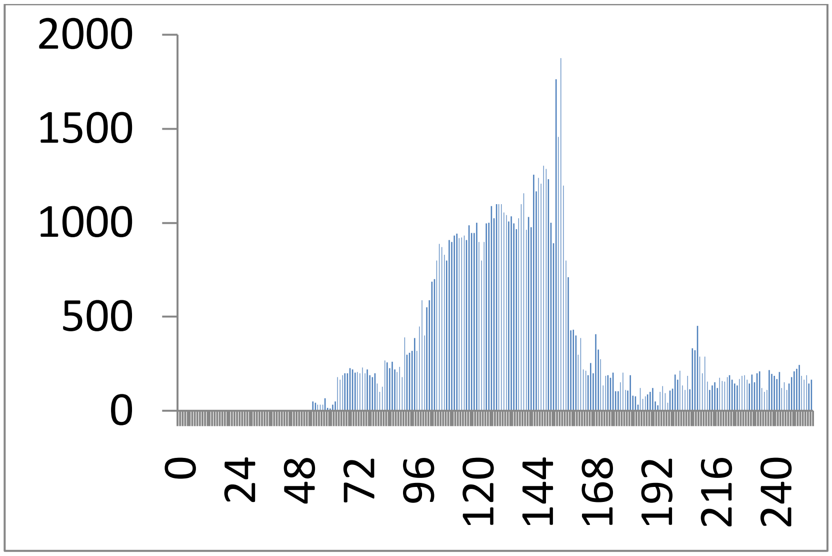


Fig 1.5 Histogram of facial region

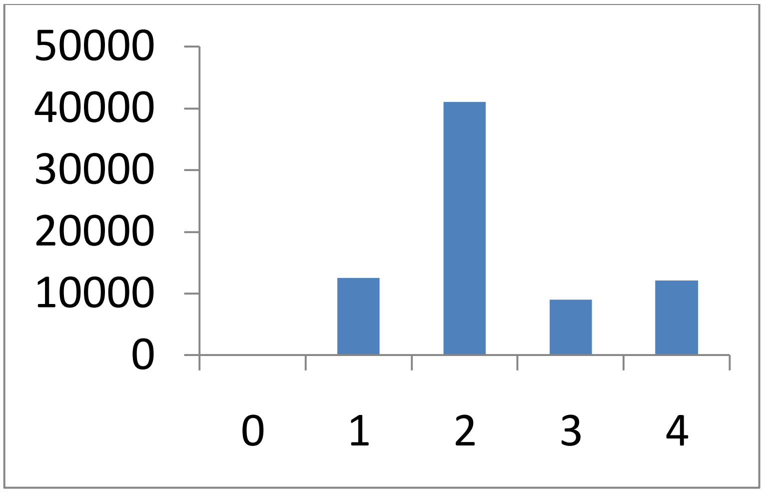


Fig 1.6 Histogram after combining values into 5 regions.

In order to determine the exposure level we build normalized histogram of the image. Then we divide it into five regions and calculate the Mean Sample Value (MSV) to determine the exposure.

The MSV is calculated using the formula:

$$\mu = (\sum (i + 1) \cdot xi) / \sum xi, \text{ where } xi \text{ is the sum of the values in region } i.$$

Using MSV, the image is correctly exposed when μ is approximately equal to 2.5. For detecting the correctness of background we first do the face detection. Then using a suitable mask we cover the face and concentrate on the remaining area which is predominantly the background.

For background to be smooth and free from shadows we first construct its histogram and then find the maxima and if the majority of the pixels occurrence is within this region of the maxima then we can conclude that the background is correct.

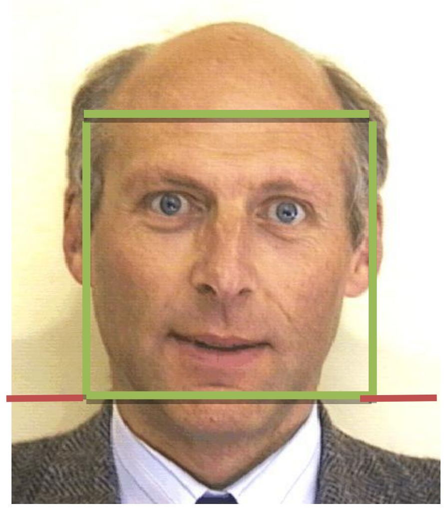


Fig 1.7

For this, we first detect the face and then mask out that area and then use the pixels above the red line and outside the bounding box for the histogram as shown above in Fig 1.7. Then we construct the histogram of the background, and if the background is smooth then most of the intensity values would be in a particular region of the histogram. Since the database we are working on has predominantly white background so if the background was correct most of the values would be in the righter region of the image histogram.

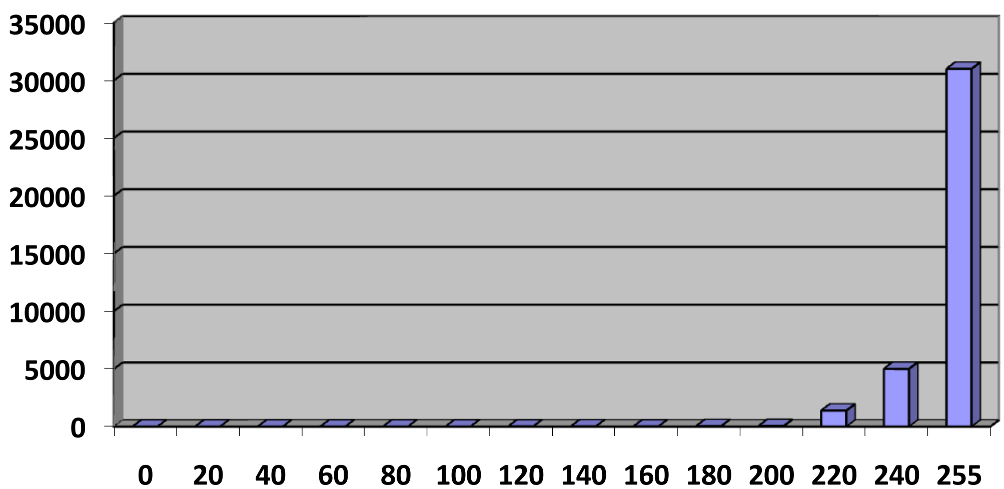


Fig 1.8

DISCUSSION:

Since the determination of background correctness depends on the face detection so if the face detection is not done correctly then the construction of the mask is inaccurate and thus we get false results. Also, since we need to separate the images using a certain threshold value so it many times it is difficult to set this value as it may work for some images but could be wrong for others. Another difficulty is construction of accurate masks so that we can have only the background but if the mask is inaccurate then we might have certain body parts like ear or hair as part of the background and thus would do incorrect classification.

BLUR ANALYSIS

INTRODUCTION:

Blur of the picture is the second analysis that is performed on the image in this project.

When taking digital photographs for passport the blurredness of the picture could be a result of either one or a combination of the four factors listed below:

1. Image is out of focus
2. The subject moves while the shutter of the camera is open
3. The camera moves while the shutter is open
4. Depth of the field is too shallow

All these factors results in the image being blurred.

Fig 2.1

Exposure analysis removes the images having out of depth field due to the non-uniformity of the background.

With a uniform background the rest of the factors causing blurriness of the image are taken care of in this analysis.

PROPOSED METHOD:

The proposed method performs Haar Wavelet Transform. This technique judges whether or not an image is blurred based on the edge analysis and also determines the extent of edge sharpness in the image.

There are basic 3 types of edges Dirac, Step and Roof edge structures. The step structure is further differentiated into A-Step and G-Step depending on whether the change of intensity is gradual or not.

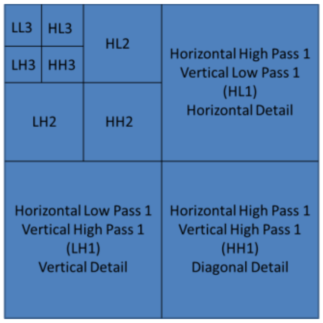
Edges	After Blur Effect
Dirac Structure	Roof Structure
A – Step Structure	G – Step Structure
G – Step Structure	G – Step Structure
Roof Structure	Roof Structure

When an image is blurred the transition from one edge to another happens in the format shown in the table above.

The algorithm is such that when an image gets blurred all the Dirac and A-Step structures in an image disappear. Thus, this algorithm determines if an image is blurred depending on the presence of the Dirac and A-Step structure. The blur extent in an image is calculated on the percentage of the G-Step and Roof structure present.

Algorithm for HWT for Edge Detection:

1. Perform HWT and decompose it to the third level.



2. Construct an edge map.

   The formula to the edge map is: $EMap_i = sqrt\{(LH_i)^2 + (HL_i)^2 + (HH_i)^2\}$

   where i = 1, 2, 3

3. Partition the image and find local maxima in each window with the window size being 2X2, 4X4 and 8X8.

Algorithm for Blur Detection:

1. Perform the algorithm 1 on the given image
2. Calculate the number of edges N_{edges} in the image using condition given below:

   $EMax_i >$ threshold where i = 1, 2, 3

3. Calculate the number of Dirac and A-Step structures (N_{da}) in an image using the condition given below:

   $EMax_1 > EMax_2 > EMax_3$

4. Calculate the number of roof and G-step structures (N_{rg}) in an image using the two conditions given below: For all edge points

   $EMax_1 < EMax_2 < EMax_3$ indicates a Roof or G-step structure and

   $EMax_2 > EMax_1$ && $EMax_2 > EMax_3$ indicates a Roof structure

5. For any roof structure of G-step structure calculate the number of edges which satisfies the condition (N_{brg}) if $EMax_1 < threshold$
6. Calculate if the image is blurred using the following condition

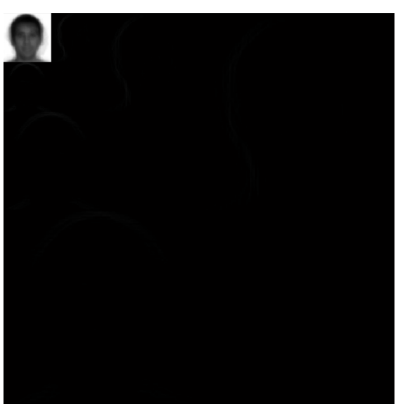
   $per = \frac{N_{da}}{N_{edges}} > threshold$

   If per is greater than threshold, the image is not blurred else it is blurred.
7. Calculate Blur extent using the condition

$$Blur\ Extent = \frac{N_{brg}}{N_{rg}}$$

DISCUSSION:

1. The proposed method is efficient and effective in uniform lightening. However, if the lightening is non-uniform the image is not properly detected.
2. The proposed method depends on the threshold of data. Thus, at times this threshold will vary for different data types with different lighting conditions.



Haar Wavlet Transform

Blurred Image

Fig 2.2

Haar Wavlet Transform

Acceptable Image

Fig 2.3

EYE PUPIL DETECTION AND ANALYSIS

INTRODUCTION:

In this section, we would be focusing on pupil detection. Pupil detection is important for finding the open eyes which are clearly visible, and are not covered by hair. Figure3.1 shows both valid and invalid images.

There are various challenges associated with pupil detection:

1. Bad lighting conditions
2. Low resolution images
3. Detection of eyes with spectacles on
4. Presence of dark/thick eyelashes and eyebrows

Fig 3.1. Left image shows eyes covered by hair, center image has close eyes and right image is valid

PROPOSED METHOD:

1. Pre Processing

For pre processing of eye region, we subjected eye region to contrast stretch and Gaussian smoothing. Contrast stretching is an image enhancement technique, used to improve the contrast in an image. It works by stretching the range of intensity values it contains to span a desired range of values. The Gaussian smoothing operator is a 2-D convolution operator that is used to `blur' images and remove detail and noise. A 2-D Gaussian kernel is given by:

$$G(x,y) = \frac{e^{\frac{-(x^2+y^2)}{2\sigma^2}}}{2\pi\sigma^2}$$

Gaussian smoothing helps us to remove noise and keep edges sharp.

2. Thresholding

Otsu thresholding is a brute force method to search for the best threshold which minimizes the between-class variance. Otsu thresholding minimizes the following term:

$$Variance = w_1^2(t) * sigma_1^2(t) + w_2^2(t) * sigma_2^2(t)$$

Where weights are the probabilities of the two classes separated by the threshold t and sigma^2 are the variance of the two classes. Otsu thresholding removes the requirement of picking the right parameter for our binary thresholding step. However, it doesn't work well with skin under arbitrary lighting. Thresholding results are shown in figure 3.2.

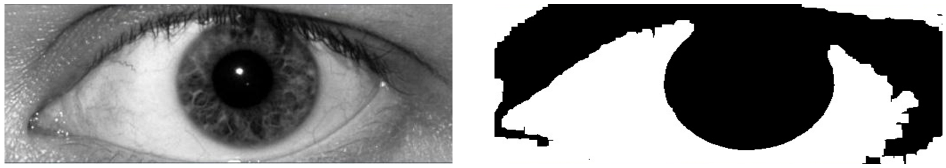


Fig 3.2. Result of using adaptive

3. Edge Detection

Edges characterize boundaries and are therefore very useful for the purpose of eye detection. Edges in images are areas with large change in gradient. By doing edge detection we can filter out the structures of uttermost importance to us. We used Canny edge detector for the extracting edges from the image.

Canny edge detector following steps:

- **Smooth the image with Gaussian filter** : It can be written as

$$S[i,j] = G[i,j,\sigma] * I[i,j]$$

Where,

$I[i,j]$ = Original image

$G[i,j,\sigma]$ = Gaussian smoothing filter

$S[i,j]$ = Smoothed image

- **Compute gradient** : Gradient magnitude and orientation is calculated using finite-difference approximations for the partial derivatives. Using the smoothed image,S, compute the x $(P[i,j\,])$ and y $(Q[i,j\,])$ partial derivatives. P, Q are computed with averaging the finite differences over the 2x2 square. The magnitude and orientation of the gradient can be computed as:

$$M[i,j] = \sqrt{P[i,j]^2 + Q[i,j]^2}$$

$$\theta[i,j] = \arctan(Q[i,j], P[i,j])$$

Where,

M = magnitude of the gradient

θ = orientation of the gradient

- **Nonmaxima Suppression :** an edge point is a point whose strength is maximum in the direction of the gradient. This process is useful in thinning the edges/ridges found by thresholding.
- **Thresholding:** Even after nonmaxima suppression image will have many false edges due to the presence of noise and fine texture. These false edges cam be removed by the process of thresholding, but the process of obtaining threshold value is difficult. Lower threshold may not remove false edges while higher threshold value might result in the removal of relevant edges. To overcome the problem, two threshold values, $r_1$and r_2 are applied to the image result of the previous step. Here $r_2 \approx 2r_1$. With these threshold values, two thresholded edge images $T_1[i, j]$ and $T_2[i, j]$ are produced. The image T_2 has gaps in the contours but contains fewer false edges. With the double thresholding algorithm the edges in T_2 are linked into contours. When it reaches the end of a contour, algorithm looks in T_1 at the locations of the 8-neighbours for edges that can be linked to the contour. This algorithm continues until the gap has been bridged to an edge in T_2. The algorithm performs edge linking as a by-product of thresholding and resolves some of the problems with choosing a threshold.

Figure 3.3 shows the result of applying edge detection:

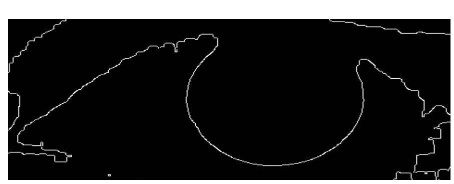


Fig 3.3. Result of edge

4. Hough Transformation

After estimation of edge pixel, next task is to find pupil of an eye. Hough transformation can be used to find pupil in an eye.

Assume we have some data points in an image which are perhaps the result of an edge detection process, or boundary points of a binary blob. We wish to recognise the points that form a straight line.

In theory any kind of curve can be detected if you can express it as a function of the form

$$f(a_1, a_2, \dots.., a_3, x, y) = 0$$

For example a circle can be represented as

$$(x - a)^2 + (y - b)^2 - r^2 = 0.$$

One then has to work in n dimensional parameter space (three dimensional space for a circle).

This model has three parameters: two parameters for the center of the circle and one parameter for the radius of the circle. If the gradient angle for the edges is available, then this provides a constraint that reduces the number of degrees of freedom and hence the required size of the parameter space. The direction of the vector from the center of the circle to each edge point is determined by the gradient angle, leaving the value of the radius as the only unknown parameter. Thus, the parametric equations for a circle in polar coordinates are

$$y = a + r\sin\theta$$

$$x = a + r\cos\theta$$

Now given the gradient angle θ at an edge point (x,y), we can compute $\cos\theta$ and $\sin\theta$. We can eliminate the radius from the pair of equations above to yield

$$b = a \tan\theta - x \tan\theta + y$$

DISCUSSION:

Figure 3.4 shows the result of pupil detection. In the first two images, we could detect pupil. In the third set of image, we were able to do pupil detection even with the glasses on. As long as glasses are not dark, our algorithm works perfectly fine. In the last set of image person's eyes are closed, so there was no pupil found.

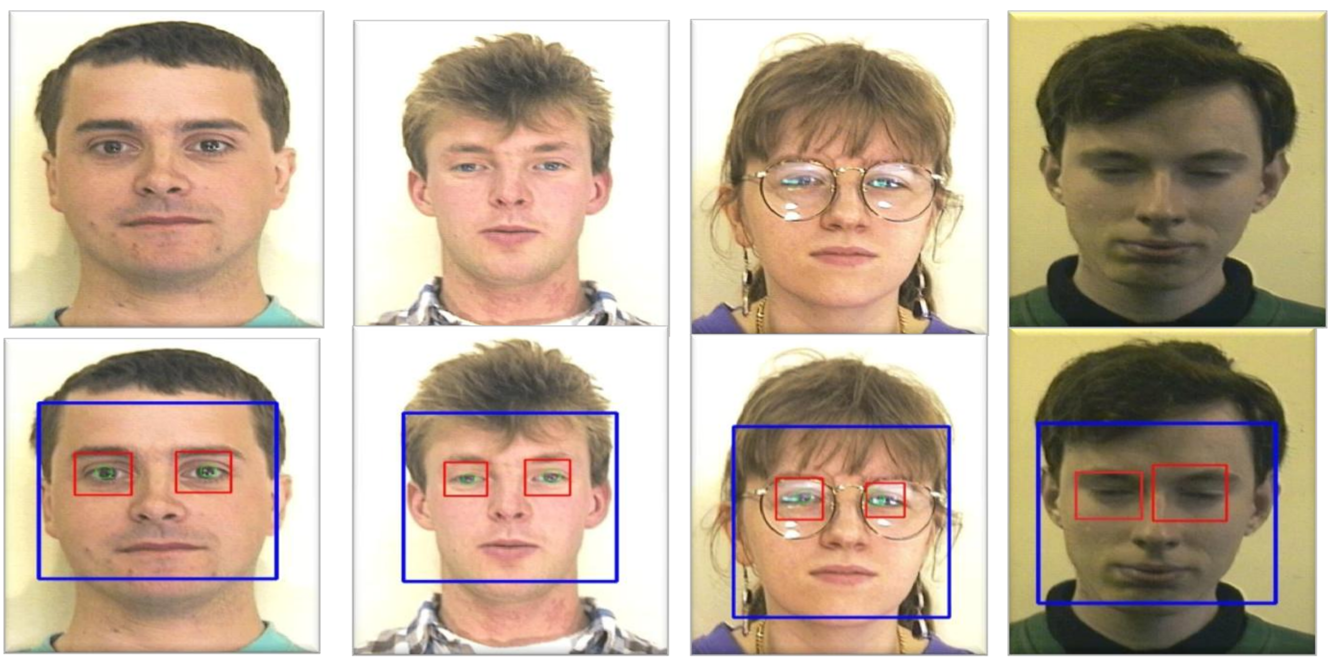


Fig 3.4: Results of pupil

Figure 3.5 shows the images where our algorithm is failing. First image set has under exposed image, second set has irregular lighting condition. In the final pair our algorithm fail to detect pupil even though it was present.

Fig. 3.5: Results showing failed pupil detection, because of uneven lighting

Our algorithm works well when the images are well exposed, face is in the frontal position and there are no occlusions near the eye region. The algorithm is too sensitive to lighting condition. Algorithm works even for faces with spectacles in, only when the lens of the spectacle is clear. We worked on the dataset that has low resolution images, algorithm works fairly well. This algorithm works much better on images with higher resolution.

FACE ORIENTATION AND HEAD COVERING

INTRODUCTION:

The face orientation and head covering are addressed in this section. In the one hand, visa photo requires that the face should be looking straight into the camera, not leaning or looking over one shoulder as normally occurred in portrait style. In the other hand, head coverings are not permitted except for the religious reason. However, the facial features from the chin to the forehead and two edges of the face must be clear. Figure 4.1 and 4.2 describe these issues.

Figure 4.1: Face orientation requirement

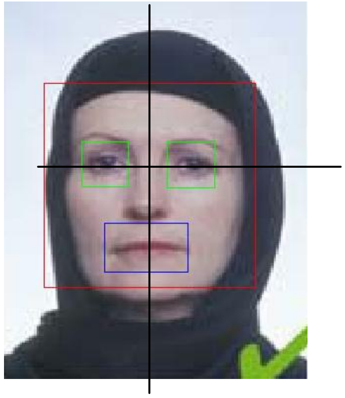
Figure 4.2: Head covering requirement

PROPOSED METHOD:

An efficient face detection algorithm is needed in order to check these requirements. Moreover, a fully automated facial symmetry axis should be determined to specify the orientation of the face. The facial features including eyes, mouth, and nose should also be detected clearly. Therefore, we employ face detector technique in opencv, an object detection based on the clever of boosting, to recognize front face, eyes, mouth of the person appearing in the photograph. It is good enough to detect any normal frontal face in which open eyes, mouth, nose should be clear. For some reason this technique can not detect any face in the visa photo, it means that the photo is not acceptable.

Having detected components of the face, we try to estimate the angle formed by a line crossing eyes and the vertical axis (Figure 4.3). If this angle is less than a certain threshold, let say 8 degree, then the face is in the right orientation (not leaning).

To check the face not to be looking over one shoulder, we check the symmetry of the face captured in the photo. Although we can take advantage of the positions of eyes, mouth which are having detected to get the facial symmetry axis (Figure 1.3), it's still hard to know whether the face is symmetric or not. The reason is that the hair and background and the head covering might be appear right beside the face and these things are not fix. However, the front face itself including skin segment only is always symmetric. As a result, we try to segment the image in order to get skin segment and none-skin segment. Watershed algorithm works well in this case when the some positions of the background, center of the face could be estimated. Having detected skin segment and facial symmetry axis (Figure 4.3), we can determine if the face is not symmetric when looking over one shoulder



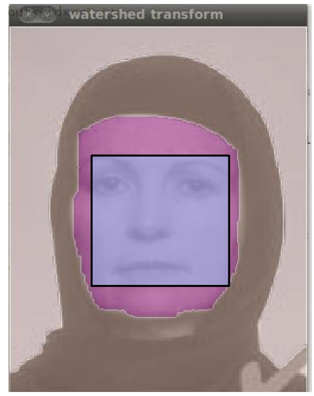


Figure 4.3: Facial symmetry axis and skin segmentation

We also use the skin segment to check whether or not all facial features including eyes, mouth, nose and two edges of the face should be seen. We calculate a rectangular containing eyes, mouth and broaden it a little bit. If this rectangular is completely inside the skin segment, then the frontal face should be seen clearly.

DISCUSSION:

The accuracy of the proposed method is depending on the ability of face detection in term of the position of eyes, mouth. Although opencv works pretty well with it but it is quite slow. The performance is also depending on the pre-processing steps of removing the bad quality of photo in term of blur, exposure, unneutral background. In order to solve these problems, a face detection and skin segmentation should be done based on skin color model, and a facial symmetry axis should be done independently [1][2].

OPEN/CLOSED MOUTH DETECTION

INTRODUCTION:

In this section we present an automatic algorithm to detect if the mouth is open or closed. As a requirement for a passport photograph, the subject should not be laughing or opening his or her mouth. A similar problem can also be researched for other purposes, such as building a smile detection function for digital cameras. However, those smile detection methods are quite different to this project, since they can take the temporal information while the subjects are tracked all the time. In this case, the only input we have is the single photo.

To tackle this problem, the base of the observation is the visibility of the teeth while the subject's mouth is opened. As the result, the mouth patch on the photo would show more details. For example, in figure 5.1, the patch of mouth open has more edges than the patch of mouth close. Using this observation, an edge detection algorithm is adopted in order to quantify the amount of detail in the mouth region.

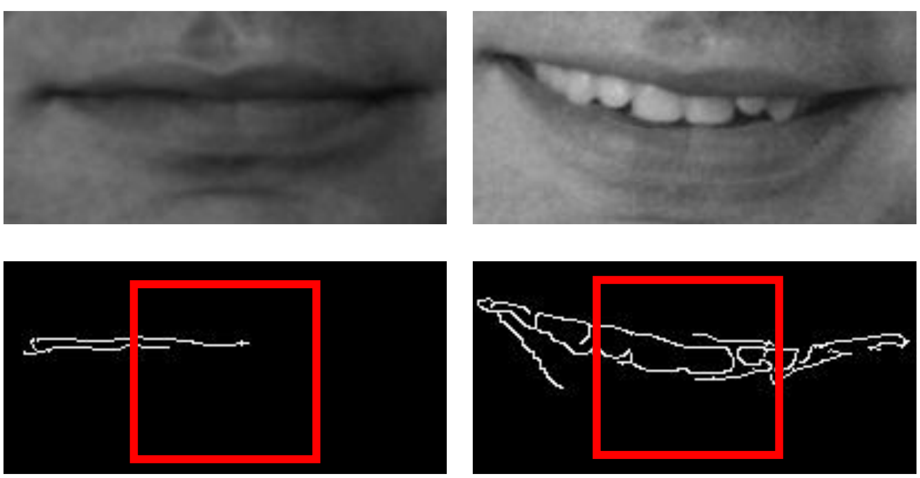


Fig 5.1. Canny edge detection results on mouth closing (left) and mouth opening patch (right).

PROPOSED METHOD:

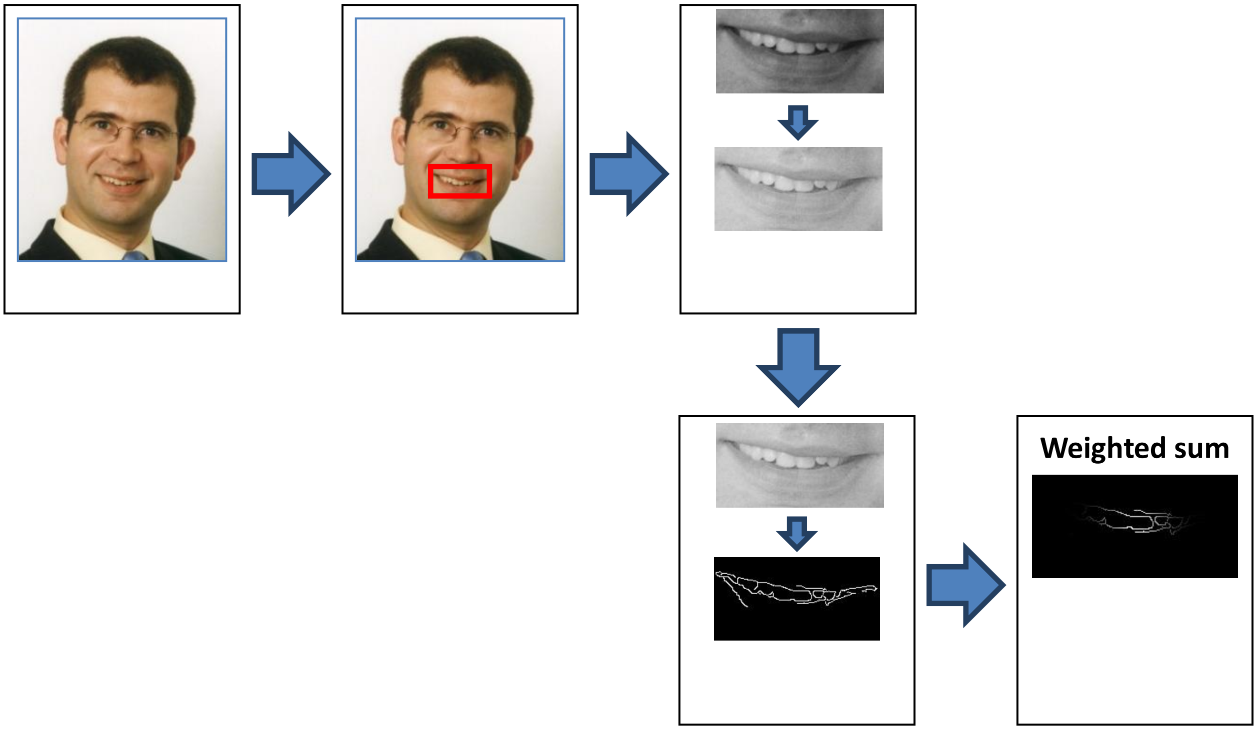


Fig 5.2. Overview of our approach.

The approach to mouth detection consists of four steps as shown in figure 5.2. On mouth detection step, the Haar feature-based cascade classifier is employed. And on the edges detection step, the Canny edge detector is used. In both these steps, the OpenCV implementation was used. The two other steps are as follows:

Luminance Correction

Since a perfect exposure and contrast cannot be expected in the input images a need to do correction to improve the performance for the later step of edge detection step has to be done. Here the adapt photoreceptors compression [9] to correct the luminance is employed. In contrast with a well-known histogram matching technique, which corrects the whole image by a global curving of histogram, the photoreceptors compression would dynamically adapt to the local luminance. As the result, the adapt photoreceptors compression can bring more details and give more consistent results than histogram matching (figure 5.3).

Original Histogram equalization Photoreceptors compression

Figure 5.3. Luminance correction comparison between two methods.

Here, the model of the photoreceptor compression by the Michaelis-Menten equation is shown below:

$$\begin{cases} r(p) = \dfrac{R(p)}{R(p) + R_0(p)} \cdot (255 + R_0(p)) \\ R_0(p) = \dfrac{V_0}{256} \cdot L(p) + (255 - V_0) \end{cases}$$

Where:

- p is an image pixel
- $r(p)$ is the corrected luminance
- $R(p)$ is the input luminance
- $R_0(p)$ is the photoreceptor gain
- $L(p)$ is the local luminance
- V_0 is a parameter

In this equation, the luminance is corrected by using the local luminance *L(p)*, which is calculated by blurring the input image and a parameter V_0. Here we set the V_0 = 230 as suggested by [10].

Thresholding

On this step, after getting the edges image from the Canny edge detector, a count of the sum of pixels on the edges is performed and threshoding with this sum is done to determine if the mouth is open or closed. Since the mouth patch is a rectangle while the mouth shape is generally an ellipse the edges of the image is multiplied with a Gaussian image before accumulating to the sum as illustrated in figure 5.4.

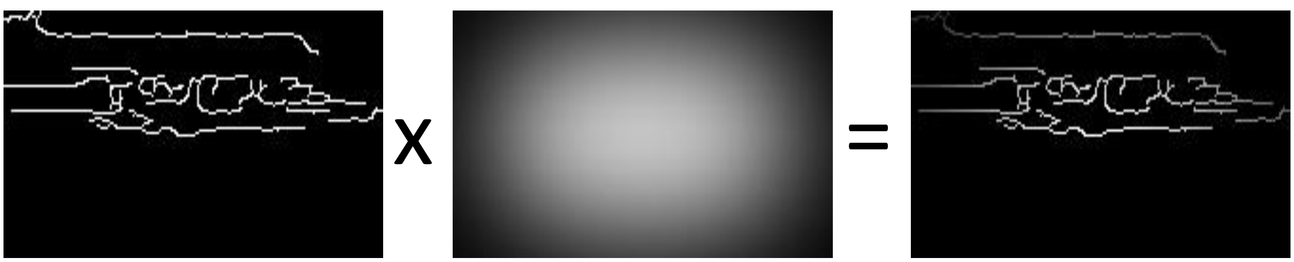


Figure 5.4. Multiplying the edges image with a Gaussian weighted image to extract the correct mouth region.

To validate the ability of discrimination by the proposed sum, manual selection of 40 mouth-opening images and 40 mouth-closing images is done and a plot of the distribution of the sum of the edges pixels is illustrated. The result is shown in figure 5.5, in which the mean of the sums in two groups are 151,130 and 78,615 respectively.

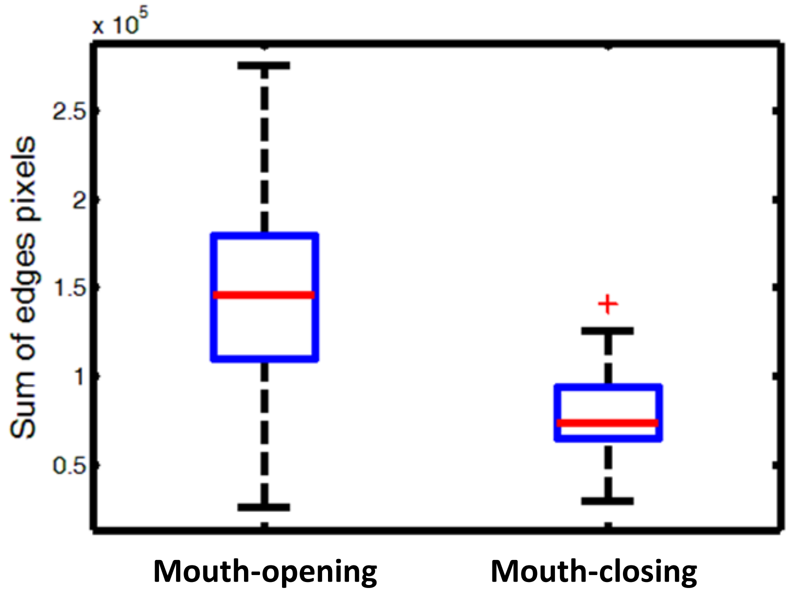


Figure 5.5. The distributions of the sum of the edges pixels for mouth-opening images set and mouth-closing images set.

DISCUSSION:

Below are some of the results shown:

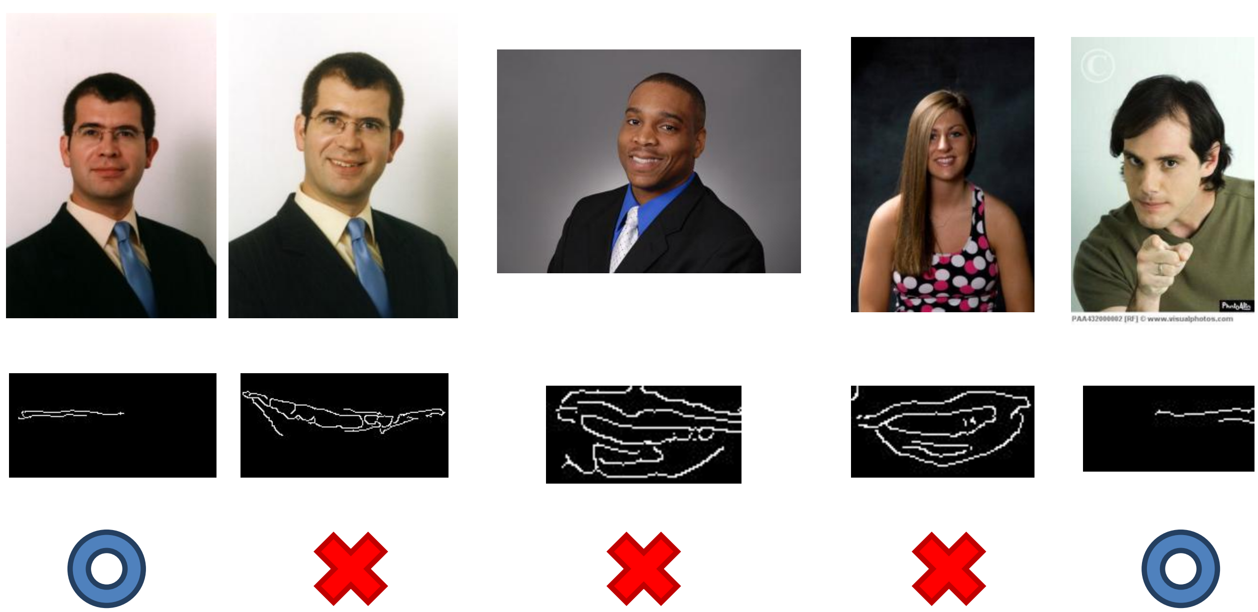
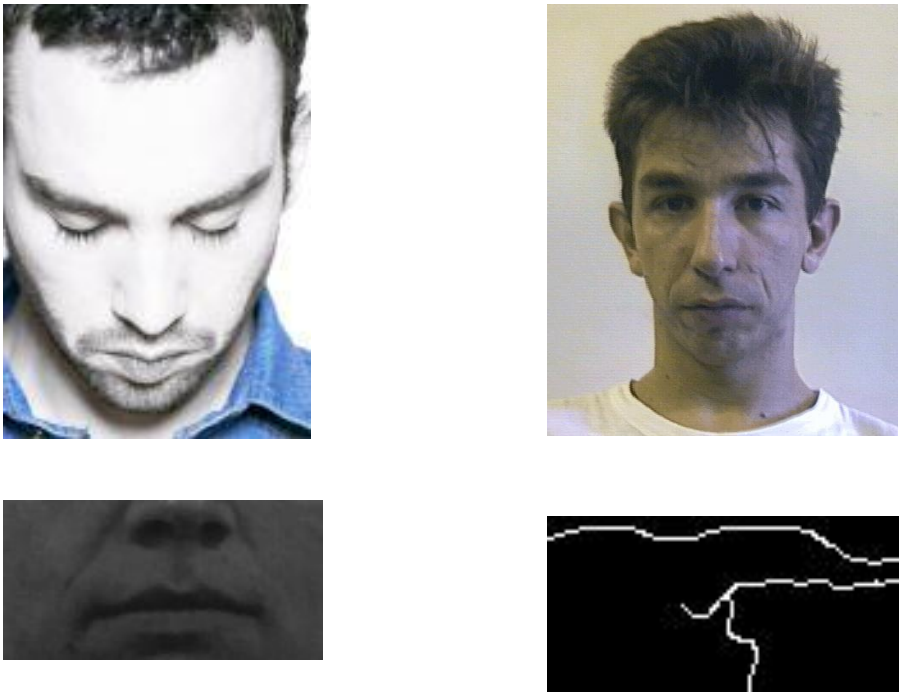


Figure 5.6. Some detection results.

However, this method still has some limitations which are listed below:

- The algorithm depends on the mouth detection.
- The algorithm will fail in irregular lighting cases.



Irregular lighting

Failed on mouth detection

Figure 5.7. Some failed cases

RESULTS

We performed analysis on 2 data sets.

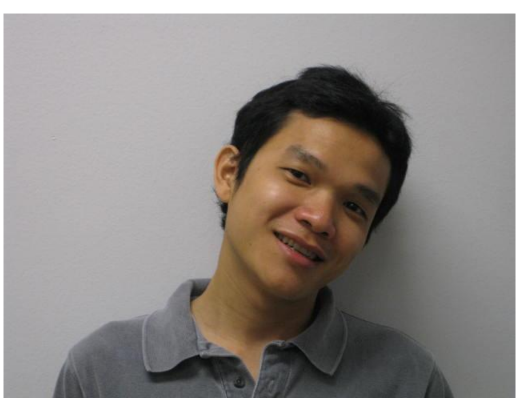
The first data-set had 683 images of which 128 images were considered a perfect image. Below are few sample images

However, those images could not be validated due to time constraints. We manually validated the result for four classes. Below is the table showing results.

	True Positive	True Negative	False Positive	False Negative
Open Eyes	523	29	131	0
Closed Mouth	606	18	27	11
Blur	558	0	0	135
Exposure	276	365	40	2

The next of data-set we took was real-time. We took 10 images in this dataset under different lightening conditions, orientations and blur effects.

We manually validated this dataset for all classes. Below is the table showing results

	True Positive	True Negative	False Positive	False Negative
Open Eyes	2	2	6	0
Closed Mouth	6	2	1	0
Blur	1	7	2	0
Exposure	2	3	5	0
Orientation	1	0	7	2

CONCLUSION

In this project we developed a system that can automatically qualify or disqualify a picture for VISA application based on well-defined criteria. Our criteria for a good picture were no blur, good exposure with uniform background lighting, open eyes, closed mouth, right face orientation and entire face visibility. We successfully implemented six independent different modules which were integrated later to achieve our goal. The system was tested on a set of about 683 single facial images, out of which 128 images were qualified as eligible pictures by our system. However, these results need to be manually validated to test the accuracy of our system. We were able to achieve this validation only for three methods. In order to overcome that limitation, we validated our system on 10 images. Dataset 1 and Dataset 2, were completely different from each other. When we ran our code on dataset 2, with same parameter as in dataset 1, method failed miserably. This shows that our system is not robust enough. Nevertheless, the project was successful in initiating an automated system that can access pictures for VISA photo requirements.

REFERENCES

1. These photographs have been taken from this http://www.immihelp.com/visas/schengenvisa/photograph-requirements.html website.
2. Automatic Camera Exposure Control, Navid Nourani-Vatani and Jonathan Roberts.
3. Canon. Canon technologies. Internet: http://www.canon.com/technology/, 2007.
4. Studying Aesthetics in Photographic Images Using a Computational Approach, Ritendra Datta, Dhiraj Joshi, Jia Li James, Z. Wang.
5. Blur Detection for Digital Images Using Wavelet Transform – Hanghang Tong et al
6. http://www.onthegosoft.com/passport_photo_specifications.htm
7. X. Chen, P. J. Flynn, K. W. Bowyer, Fully Automated Facial Symmetry Axis Detection in Frontal Color Images.
8. H.Chang,U.Robles-Mellin, http://www-cs- students.stanford.edu/robles/ee368/skincolor.html

9. Hérault, J., Durette, B.: Modeling Visual Perception for Image Processing. In: F. Sandoval et al. (Eds.): IWANN 2007, LNCS 4507, 662–675, Springer-Verlag Berlin Heidelberg (2007)

10. Christian Bouvier, Alexandre Benoit, Alice Caplier, and Pierre-Yves Coulon. 2008. Open or Closed Mouth State Detection: Static Supervised Classification Based on Log-Polar Signature. In *Proceedings of the 10th International Conference on Advanced Concepts for Intelligent Vision Systems* (ACIVS '08). Springer-Verlag, Berlin, Heidelberg, 1093-1102.

CONTRIBUTION

Face Detection	Benjamin
Report, Ppt	Fatima, Benjamin, Charu
Blur Analysis	Fatima
Scarf/ orientation	Son
Eyes	Joseph, Charu
Mouth	Binh
Exposure/lighting	Varun