Chapter 4

Preprocessing on Human Face Images

4.1 Introduction

Computer vision experiments like face recognition, signature verification, forensic document examination, character recognition etc. require preprocessing (pixel level or low level processing) of acquired images from camera or scanner arrangement. Different preprocessing algorithms are applied on captured images to improve their qualities to make them fit for further analysis and recognition [Ha & Bunke 1997], [Chanda et al. 1983], [Chanda et al. 1984].

Since we have used facial images captured using mobile phone camera, in the acquired digital image there are issues like poor contrast, presence of some noise etc. caused by inadequate lighting, aperture size, shutter speed or nonlinear mapping of image intensity. Therefore we put all the images acquired to undergo preprocessing technique like contrast stretching, smoothing and size normalization.

The first preprocessing technique we applied is size normalization. Though it is not mandatory, we made all the samples collected to undergo a size normalization process. As a result we could ensure all the face image samples in the database to have uniform height and width. Many a time it has simplified
the process of writing and applying certain algorithms to perform eventual image processing tasks like representation of images and feature extraction.

The second preprocessing technique that has been employed here is smoothing. This preprocessing is essential because there are spurious noises and small gaps in curves or lines in the gathered face images. Smoothing enables us to bridge these gaps and diminish the effect of noise prior to recognition [Gonzales & Woods 2002], [Chanda & Majumder 2000].

The third preprocessing technique we applied is contrast stretching. In this case the contrast of the face images acquired are improved by scaling gray level of each pixel, so that the image gray level occupy the entire dynamic range available. We found histogram of the each face image. The defects were clearly reflected in the range and shape of the gray level histogram. The operation performed for contrast stretching based on image histogram is normally termed as histogram stretching.

This chapter is organized in three sections. First section deals with the size normalization process applied to achieve size uniformity to the face image samples. In the second section we describe image smoothing algorithms for noise reduction and quality enhancement of the face images. The third section presents the face image contrast enhancement technique applied on face image samples of both KNUFDB and AT & T face databases.

4.2 Size normalization of face image samples using affine transformation and bilinear interpolation

The high variability in the face image size and shape pose serious problems in designing face recognition algorithms. By observing the face image samples in the image database it is clear that there are significant size variations in
the collected face image patterns. So it is necessary to apply normalization for size invariance on face images prior to feature extraction. For this purpose, face images are cropped into a minimum rectangle, which has the same height and width of the original face image pattern. It is then mapped into a standard window of 112 x 92 pixel size. For this mapping the Affine Transformation technique is applied on the face images with the Bilinear Interpolation algorithm using four nearest neighbours for interpolation [Kreyszig 1993] [Hearn & Baker 1986]. The method for finding the transformation matrix $T$ is given in algorithm 1.

**Algorithm 1**: To find the transformation matrix $T$

1. Initialize final window size $m \times n$.
2. Read the face image to be size normalized from the database.
3. Find the height ($\text{ImH}$) and width ($\text{ImW}$) of face image
4. Compute the scaling factors $S_x$ and $S_y$ using the expressions
   \[
   S_x = 1 + \frac{m - \text{ImW}}{\text{ImW}}
   \]
   \[
   S_y = 1 + \frac{n - \text{ImH}}{\text{ImH}}
   \]
5. Form the transformation matrix $T$ and stop
   \[
   T = \begin{pmatrix}
   S_x & 0 & 0 \\
   0 & S_y & 0 \\
   0 & 0 & 1
   \end{pmatrix}
   \]

While implementing the procedure for size normalization of the face images, in some cases, due to the rounding off error caused by the computation of scaling factors $S_x$ and $S_y$, the final image size may vary in a very small factor from the standard size. This problem is resolved by applying zero padding and hence the final output image size is exactly 112 x 92 pixels.
4. Preprocessing on Human Face Images

Experimental results

The face image samples of varying sizes are normalized into the standard output size using Affine Transformation and Bilinear Interpolation algorithms. The Figure 4.1 show the input face images of size 150 X 120 pixels, and the size normalized images obtained after the above explained size normalization operations is shown in 4.2. The experimental results indicate that the normalized images of 112 x 92 pixel size show comparatively less geometrical distortions after the transformation operation. This most favorable output window size (112 x 92 pixels) is fixed by the trial and error experiments.

![Figure 4.1: Images before size normalization in KNUFDB Face database](image)

4.3 Face image quality enhancement using digital smoothing filters

Human face images are acquired by digital mobile camera which uses photo electronic method. The sensing devices tend degrade the quality of the image
by introducing the noise, geometric deformation due to motion or camera mis-focus  [Gonzales & Woods 2002]. One of our primary concerns is to increase image quality and to moderate the degradation introduced by the mobile camera sensing and acquisition device. We used image restoration techniques for reconstruction or recovery of the images that have been degraded. We also used digital image enhancement techniques to increase the subjective image quality by sharpening certain image features (e.g. contrast, edges and boundaries) and by reducing noise. Digital image enhancement and restoration operation can be thought of as two-dimensional digital filter function. We employed spatial operations for digital image enhancement.

4.3.1 Estimation of noise spectrum

A prior knowledge about the degradation phenomenon is required to design image restoration model for reducing the noises present in the face images acquired using digital mobile camera. For this we studied the characteristic of the system noise using a simple method. A set of images of flat environments.
4. Preprocessing on Human Face Images

(a) Flat gray board  
(b) Histogram of the gray board

Figure 4.3: Noise present due to camera - through histogram analysis

For this we used a solid gray board that illuminated uniformly as the *flat* environment. Resulting images as well as the shape of the histogram are good indicators for system noise. On analysis of the digital images, it is found that the noise is present in the high frequencies of the image spectrum as illustrated by figure 4.3. Therefore, a lowpass filter may be used for noise removal. The following section describes implementation of 2D digital filters for face image noise reduction.

4.3.2 Face image quality enhancement using two dimensional digital filters

2D linear finite impulse response (FIR) digital filters are linear shift-invariant two-dimensional systems, whose region of support is finite [Pitas 2000]. In many cases the FIR impulse response $h(n_1, n_2)$ is non-zero only in the rectangle

$$R_{m_1 m_2} = [0, M_1) \times [0, M_2) = (n_1, n_2) : 0 \leq n_1 < M_1, 0 \leq n_2 < M_2$$

The output $y(n_1, n_2)$ of FIR digital filter is given by linear convolution.

$$y(n_1, n_2) = \sum_{k_1=0}^{M_1-1} \sum_{k_2=0}^{M_2-1} h(k_1, k_2) x(n_1 - k_1, n_2 - k_2) \quad (4.3.1)$$
Frequently, the region of support of the FIR digital filter is \([-v_1, v_1] \times [-v_2, v_2]\], where \(M_i = 2v_i + 1, i = 1, 2, \ldots\) are odd-valued filter lengths. In this case, the output of FIR filter is given by

\[
y(n_1, n_2) = \sum_{k_1=-v_1}^{v_1} \sum_{k_2=-v_2}^{v_2} h(k_1, k_2) x(n_1 - k_1, n_2 - k_2)
\]  
(4.3.2)

We employed this definition for the filter designed. The moving average filter is the simplest two dimensional lowpass FIR digital filter. If its lengths are odd numbers \(M_i = 2v_i + 1, i = 1, 2, \ldots\) it can be defined as follows.

\[
y(n_1, n_2) = \frac{1}{M_1 M_2} \sum_{k_1=-v_1}^{v_1} \sum_{k_2=-v_2}^{v_2} x(n_1 - k_1, n_2 - k_2)
\]  
(4.3.3)

| \((-1,-1)\) | \((-1,0)\) | \((-1,1)\) |
| \((-2,-1)\) | \((-2,0)\) | \((-2,1)\) |
| \((-3,-1)\) | \((-3,0)\) | \((-3,1)\) |

Figure 4.4: A sample filter

Its impulse response is given by \(h(n_1, n_2) = \frac{1}{M_1 M_2}\) for \((n_1, n_2) \in R_{m1m2}\). The figure 4.4 shows the filter mask used. Figures 4.5 and 4.6 show a sample of how the face images will be affected by applying moving average filter on AT&T and KNUFDB face databases respectively. Here top left image of each subfigure shows face images in the database before applying filter and the bottom image shows face image after applying filter. In each case their histograms are displayed in the right side of the respective face images.

On visual inspection of filtered images in figures 4.5 and 4.6, it is found that the filter operates only in the interior \([v_1, N_1 - v_1] \times [v_2, N_2 - v_2]\) of
the image of size $N_1 \times N_2$. To obtain filtering of the entire image, if desired, the image must be extended to the size $(N_1 + M_1 - 1) \times (N_2 + M_2 - 1)$ by introducing suitable border pixel values. Another observation is it tends to blur edges and image details and degrade image quality.

![Figure 4.5](image1.png)  
![Figure 4.6](image2.png)

**Figure 4.5:** Result of moving average filtering on AT & T face database (top non-filtered image and bottom filtered image).

**Figure 4.6:** Result of moving average filtering on KNUFDB face database (top non-filtered image and bottom filtered image).

To remedy this we used nonlinear lowpass filters that remove noise effectively and preserve image edges and details [Pitas & Venetsanopoulos 1990]. Such a filter class is based on data ordering. Let $x_i, i = 1, 2, \ldots, n$ be $n$ observations, whose number $n = 2v + 1$ is odd. They can be ordered according to
their magnitude as follows:

\[ x(1) < x(2) < \ldots x(n) \ldots . \]

\( x(i) \) denotes the \( i^{th} \) order statistics, \( x(1) \) and \( x(n) \) are maximum and minimum observations respectively [Davies & Plummer 1981]. The observation \( x(v + 1) \) lies in the middle and it is called median of the observations [Pitas & Venetsanopoulos 1990, Turk & Pentland 1991]. A two dimensional median filter has the following definition

\[ y(i, j) = \text{med}\{x(i + r, j + s), (r, s) \in A, (i, j) \in Z^2\} \] (4.3.4)

where \( Z^2 = Z \times Z \) denotes the digital image plane. The set \( A \subset Z^2 \) defines the filter window.

Figures 4.7 and 4.8 show a sample of how the face images will be affected by applying median filter on AT&T and KNUFDB face databases respectively. Here top left image of each subfigure shows face images in the database before applying filter and the bottom image shows face image after applying filter. In each case their histograms are displayed in the right side of the respective face images.

(a) ![Image](image1.png)
(b) ![Image](image2.png)

**Figure 4.7:** Result of median filtering on AT & T face database(top non-filtered image and bottom filtered image).
4. Preprocessing on Human Face Images

![Figure 4.8](image)

**Figure 4.8:** Result of median filtering on KNUFDB face database (top non-filtered image and bottom filtered image).

On visual analysis of figures 4.7 and 4.8, it is evident that median filter behaves better than moving average filter. The quality of the images can further be enhanced by using adaptive median filtering. Based on median filter theory (Eq. 4.3.4) adaptive median filter is designed. In this filter behavior changes based on statistical characteristics of the image inside the filter region defined by $A \subset \mathbb{Z}^2$, the procedure used for this purpose is given in algorithm 2.

On application of adaptive median filter almost all impulse and additive noises were removed. Figures 4.9 and 4.10 show a sample of how the face images will be affected by applying adaptive median filter on AT&T and KNUFDB face databases respectively. Here top left image of each subfigure shows face images in the database before applying filter and the bottom image shows face image after applying filter. In each case their histograms are displayed in the right side of the respective face images. The results show that the resulting face images are suitable for further processing like feature extraction or recognition tasks.

61
Algorithm 2: Two level Adaptive median filter

\[ Z_{\text{min}} = \text{minimum gray level value in } S_{xy} \]
\[ Z_{\text{max}} = \text{maximum gray level value in } S_{xy} \]
\[ Z_{\text{med}} = \text{medium of gray levels in } S_{xy} \]
\[ Z_{xy} = \text{gray level at coordinates } (x, y) \]
\[ S_{\text{max}} = \text{maximum allowed size of } S_{xy} \]

- Level P
  1: Compute \( P_1 = Z_{\text{med}} - Z_{\text{min}} \)
  2: Compute \( P_2 = Z_{\text{med}} - Z_{\text{max}} \)
  3: if \( P_1 > 0 \) and \( P_2 < 0 \) then
  4: go to Level Q
  5: else
  6: increase windowsize
  7: end if
  8: if \( \text{windowsize} \leq S_{\text{max}} \) then
  9: repeat level P
  10: else
  11: output \( Z_{xy} \)
  12: end if

- Level Q
  1: Compute \( Q_1 = Z_{xy} - Z_{\text{min}} \)
  2: Compute \( Q_2 = Z_{xy} - Z_{\text{max}} \)
  3: if \( Q_1 > 0 \) and \( Q_2 < 0 \) then
  4: output \( Z_{xy} \)
  5: else
  6: output \( Z_{\text{med}} \)
  7: end if

Figure 4.9: Result of adaptive median filtering on AT & T face database (top non-filtered image and bottom filtered image).
4. Preprocessing on Human Face Images

Figure 4.10: Result of adaptive median filtering on KNUFDB face database (top non-filtered image and bottom filtered image).

Figures 4.11 - 4.14 show more experimental results of adaptive median filtering procedure performed on the face images of the typical subjects in the AT&T and KNUFDB face databases. Figure 4.11 shows before applying adaptive median filter and 4.12 after applying adaptive median filter using AT&T face database. Similarly figures 4.13 and 4.14 before and after the application of adaptive median filter using KNUFDB face database respectively. The results obtained by applying the above filtering algorithm on various face images show that the method is effective and can be used to enhance the image quality of the raw face images collected.

Figure 4.11: Before the application of adaptive median filtering on face images in AT&T face database
4. Preprocessing on Human Face Images

Figure 4.12: Results of application of adaptive median filtering on face images in AT&T face database

Figure 4.13: Before the application of adaptive median filtering on face images in KNUFDB face database

Figure 4.14: Results of application of adaptive median filtering on face images in KNUFDB face database
4.4 Histogram based quality enhancement of face images

Since images are collected under most natural environmental conditions using mobile phone camera, image contrast and quality has been affected badly. The images were poorly contrasted and contained spurious noises. The poor contrast of the face images were due to large number of pixels occupied only a small portion of available range of intensities. Through histogram modification each pixel is reassigned with a new intensity value so that dynamic range of gray level is increased. The underlying principle is to stretch the dynamic range of the pixel values.

Linear contrast stretching and histogram equalization are two widely utilized methods for global image enhancement. The linear contrast stretching linearly adjusts the image’s dynamic range and histogram equalization uses the input-output mapping relation obtained from the integral of the image histogram. Histogram equalization is a technique which consists of adjusting the gray scale of the image so that the gray level histogram of the input image is mapped on to a uniform histogram.

The basic assumption used here is that the information conveyed by an image is related to the probability of occurrence of gray levels in the form of histogram in the image. By uniformly distributing the probability of occurrence of gray levels in the image, it becomes easier to perceive the information content of the image. Thus through histogram modification each pixel is reassigned with a new intensity value according to its original intensity. It is quite obvious that by suitably stretching the pixel values the overall contrast of the image will increase [Gonzales & Woods 2002].
4.4.1 Face image histogram and their equalization

Histogram of an image represents the relative frequency of occurrence of the various gray levels in the image. For a digital image with gray levels in the range \([0, L - 1]\), the histogram is a discrete function \(p(r_k) = \frac{n_k}{N}\), where \(r_k\) is the \(k^{th}\) gray level, and \(n_k\) is the number of pixels in the image with \(r_k\) gray level. \(N\) is the total number of pixels in the image and \(k = 0, 1, \ldots, L - 1\).

Histogram equalization is a technique which consists of adjusting the gray scale of the image that the gray level histogram of the input image is mapped onto a uniform histogram. The histogram equalization technique is based on transformation using the histogram of a complete image. In histogram equalization, the goal is to obtain a uniform histogram for an output image.

Let \(r\) be a random variable which indicates the gray level of an image. Initially we can assume that \(r\) is continuous and lies within the closed interval \([0 : 1]\) with \(r = 0\) representing black and \(r = 1\) representing white. For any \(r\) in the specified interval let us consider a transformation of the form

\[
    s = T(r)
\]

The transformation produces a level \(s\) for every pixel value \(r\) in the original image. It is assumed that transformation \(T\) satisfies the following criteria:

(i) \(T(r)\) is a single valued function, monotonically increasing in the interval \([0 : 1]\)

(ii) \(T(r)\) lies between 0 and 1

The first condition preserves the order from black to white in the gray scale, and the second one guarantees that the function is consistent with the allowed range of pixel values. The inverse transform from \(s\) to \(r\) can be represented by

\[
    r = T^{-1}(s)
\]
Let the original and transformed gray levels can be characterized by their probability density functions \( p_r(r) \) and \( p_s(s) \) respectively. Then from elementary probability theory, if \( p_r(r) \) and \( p_s(s) \) are known and if \( T^{-1}(s) \) satisfies the first condition stated above in (i) then the probability density function of the transformed gray level is given by

\[
p_s(s) = \left[ p_r(r) \frac{dr}{ds} \right]_{r=T^{-1}(s)}
\]

If the transformation is given by:

\[
s = T(r) = \int_0^r p_r(w)dw
\]

Then substituting \( \frac{dr}{ds} = \frac{1}{p_r(r)} \) in equation 4.4.3 we obtain \( p_s(s) = 1 \). Thus it is possible to obtain a uniformly distributed histogram of an image by the transformation described by Eq. 4.4.4.

From the above discussion, it is clear that using a transformation function equal to the cumulative distribution of \( r \) (as given by 4.4.4) produces an image whose gray levels have a uniform density. This implies that such a transformation results in an increase in the dynamic range of the pixel gray values which produce a pronounced effect on the appearance of the image. Algorithm 3 gives the algorithm for histogram equalization [Acharya & Ray 2005] that has been used in this work.

It is shown in figures 4.15 and 4.16 that while histogram of the original image is very non-uniform due to lot of undulations, the histogram of the equalized image has more uniform density function. It also shows that the contrast of the face images have been greatly increased which in turn improved the image appearance.
Algorithm 3: Histogram equalization for human face images

- For every pixel in the image get gray value in variable $i$, $hist[i] = hist[i] + 1$ when $i = 0$ to $L - 1$ for a $L$ level image.
- From the histogram array, get cumulative frequency of histogram $hist_{cf}[i] = hist_{cf}[i - 1] + hist[i]$
- Generate the equalized histogram as $eqhist[i] = \left\lfloor \frac{L \cdot hist_{cf}[i] - N^2}{N^2} \right\rfloor$ where $L$ is the number of gray levels present in the image, $N^2$ is the number of pixel in $N \times N$ image, $[x]$ is the truncation of $x$ to the nearest integer.
- Replace the gray value $i$, by $eqhist[i]$ for each $i$. the $eqhist$ contains the new mapping of gray values.

![Figure 4.15: Results of histogram equalization of images on AT&T face databases](image)

![Figure 4.16: Results of histogram equalization of images on KNUDFB face databases](image)
4.5 Conclusion

Various preprocessing techniques that are capable of rendering the original gray-scale images in the database for further feature extraction studies are discussed in this chapter. Firstly, to bring about size uniformity, the face image samples are normalized into 112 x 92 pixel size using Affine Transformation and Bilinear Interpolation algorithms. Then, face image samples in the database are subjected to moving average, median and adaptive median filters so that the noise present in face images are considerably reduced. Finally histogram equalization is performed on the face image samples to bring about contrast enhancement.