

Image Preprocessing for Face Recognition

Enoch Ansumanah Stephen

Chongqing University of Post and Telecommunications, Chongqing 400065, China

Abstract: Face recognition is a key area in biometrics with broad use in access control, e-commerce, justice, and surveillance. A standard pipeline includes preprocessing, feature extraction, and classification; this paper concentrates on preprocessing. Using MATLAB, I implement histogram equalization, mean/median filtering, smoothing, sharpening, thresholding, edge detection, and homomorphic filtering for illumination compensation. I perform face alignment (rotation, cropping, scaling) and apply geometric and gray-level normalization to produce standardized inputs. Face regions are localized via anatomical landmarks and mathematical morphology. Simulation experiments compare configurations and show that high-quality preprocessing especially illumination normalization, denoising, and alignment improves visual clarity, stabilizes input variation, and boosts downstream localization and recognition accuracy. Overall, a well-designed preprocessing module is essential for robust face recognition in diverse conditions.

Keywords: Preprocessing; Median filtering; Difference of Gaussian Filtering (DOG); Image Processing; Gamma Correction; Contrast equalization; Illumination; Grayscale.

1. Introduction

1.1. Background

Image preprocessing refers to the set of operations applied to raw images before feature extraction and model training. Common operations include resizing, geometric alignment, color and grayscale conversion, and contrast enhancement. Proper preprocessing can shorten training time, improve inference speed, and increase recognition accuracy.

Face recognition has become increasingly important due to applications in mobile devices, intelligent access control, surveillance, and e-commerce. However, face images are influenced by illumination, pose, expression, age, occlusion, and sensor noise. These factors degrade the recognition rate if faces are used directly without preprocessing.

A typical face recognition system first performs image preprocessing to enhance useful facial structure while suppressing noise and irrelevant background. Then face detection and localization are performed to find the face region, followed by feature extraction and classification.

1.2. Basic Preprocessing Operations

Typical preprocessing methods used in this work include:

1. Histogram equalization to adjust global contrast.
2. Linear and non-linear grayscale transformation to stretch or compress intensity ranges.
3. Mean, median, and Wiener filtering to reduce noise.
4. Geometric operations such as cropping, rotation, and resizing to normalize the face area.
5. Illumination compensation through homomorphic and high-pass filtering.
6. Edge detection and morphological operations to highlight important boundaries.

These operations are implemented and tested in MATLAB to form a flexible preprocessing toolbox for face images.

1.3. Research Motivation and Objectives

Modern applications such as mobile authentication and intelligent video surveillance require accurate and fast face recognition over a wide range of environments. Changes in

lighting, pose, background, and camera quality can severely distort the appearance of the same person. The main goal of this thesis is to study how image preprocessing can normalize these variations and provide stable inputs for recognition algorithms.

The specific objectives are:

1. To review key preprocessing techniques used for face recognition.
2. To implement representative methods in MATLAB and analyze their effects.
3. To compare several spatial and frequency-domain filters for denoising and contrast enhancement.
4. To explore normalization and edge detection methods that preserve discriminative facial features.
5. To summarize practical guidelines for constructing a robust preprocessing pipeline.

2. Chapter 2 Literature Review

Face recognition is a dynamic area of computer vision and artificial intelligence that focuses on identifying individuals by analyzing distinct facial characteristics in digital images or videos (Phillips et al., 2000). The effectiveness of recognition systems heavily depends on image preprocessing — a step that enhances quality, removes noise, and standardizes illumination and geometry before feature extraction (Shan et al., 2003). This chapter outlines the evolution of face recognition research and highlights key preprocessing techniques that improve accuracy under diverse conditions.

2.1. Evolution of Face Recognition

The roots of face recognition trace back to the 1960s with Bledsoe and Chan's semi-automatic systems, which required manual marking of facial points. The introduction of Principal Component Analysis (PCA) by Pearson (1901) and its adaptation into the Eigenfaces model by Turk and Pentland (1991) revolutionized automatic recognition. Later, the emergence of Support Vector Machines (SVMs) (Osuna et al., 1997) and benchmark datasets like FERET (Phillips et al., 2000) advanced machine learning applications in this field.

2.2. Role of Image Preprocessing

Preprocessing mitigates the adverse effects of lighting, pose variation, and noise (Adini et al., 1997). Common techniques include histogram equalization, grayscale normalization, and filtering, which enhance feature extraction and classification performance (Jain, 2006). Illumination normalization, such as the Self-Quotient Image (SQI) method (Wang et al., 2004) and local contrast normalization (Shan et al., 2003), significantly boosts robustness in uncontrolled environments.

2.3. Filtering and Feature Enhancement

Filtering reduces noise while retaining key details. Median and Wiener filters are popular for balancing clarity and smoothness (Zhu, 2005; Liu, 2007). Frequency-based filters, including Butterworth and high-pass filters, refine edges and textures. Studies indicate that combining spatial and frequency-domain filtering improves recognition accuracy (Wang et al., 2004; Zhao et al., 2016).

2.4. Edge Detection and Feature Extraction

Edge detection algorithms such as Sobel, Canny, and Log outline facial structures crucial for recognition (Jain, 2006). The Canny operator remains favored for its precision and noise resistance. Advanced techniques like Local Binary Patterns (LBP), Gabor filters, and Kernel ICA (Bach & Jordan, 2002) extract unique and stable facial features for classification.

3. Methodology

3.1. Face recognition technology

In order to better perform face recognition, it is necessary to reduce the size, angle, expression, and illumination of the face. For the factors that affect the recognition rate, first to proceed Preprocessing, such as face detection, face positioning, etc., then get in on Comparison of facial features. Face detection and positioning need to describe and extract the characteristics of the specified image, store it in the database, and carry out the corresponding work. In order to better perform face recognition, a large amount of face information needs to be stored. This process is called repeated recognition of facial features, and it is also a method to improve the quality of the subsequent step of recognition.

Face recognition technology has been continuously developed in recent years, and considerable recognition results have been achieved. The preprocessing process is to segment the input face image during the entire face image recognition and process, such as segmentation, feature extraction, matching, face recognition, etc., The preprocessing process studied in this paper includes: grayscale transformation, various filtering methods in space and frequency domain, edge detection, wavelet transformation, etc. Preprocessing is the goal of removing irrelevant information from an image, accurately locate the facial features, making the details of the image more obvious. For example, the continuous development of face recognition technology in China can recognize faces under natural light conditions and non-natural light conditions. Because of the above advantages, face recognition is of great significance to the study of face recognition.

3.2. Methods and research design

1)Face straightening (rotate, cut and zoom): Correct face

map like.

2)Face image enhancement: Improve image the quality can be recognized with better visual effects, and it is also required to remove the influence of light and perform light compensation.

There are three main ways:

1. Linear gray scale transformation
2. Non-linear gray scale transformation
3. Histogram equalization

3)Normalization of face image: Makes the image Consistent size, gray value roughly the same. The general way is:

1. Size normalization
2. Gray normalization

4)Filtering and de-noising: Due to the influence of external factors, the image becomes a noisy image. The general method is:

1. Adaptive Wiener filter
2. Wavelet de-noising, etc.

5)Edge detection: In the process of image acquisition, the face image needs to be edge detected due to the complex background.

6)Image segmentation: Separate the main object and background of the face image.

7)Face detection: Detect whether there is a face in the image. There are mainly the following ways:

1. Method based on knowledge rules
2. Method based on visual features
3. Based on the last shift matching method

3.3. Image preprocessing

Pretreatment Part of the process is not available in face recognition. Due to the different collection environment, such as: light intensity, equipment performance is different. In addition, the distance and focal length make the size and position of the face uncertain. Therefore, in order to obtain the consistency of face size, position and image quality, to proceed Pretreatment. The so-called preprocessing is to eliminate the noise in the picture, enhance the detectability of the image, and simplify the data for feature extraction, segmentation, and matching recognition. With the continuous improvement of electronic technology and computer technology, image pre-processing technology has achieved tremendous development in video surveillance systems. The video surveillance system is used to monitor the noise in the environment and the noise generated during the transmission and reception of the image, which reduces the image quality and blurs the image. We can improve the quality of images through image enhancement technology. In an image, people are only interested in certain targets in the image. We use image segmentation technology to divide the image into different regions to separate each object in the image, and then obtain the features of the object from these regions to extract our feelings. The goal of interest. With the continuous development of image preprocessing technology, the ability of computers to process images has also improved, and preprocessing technology has become more and more important. There are also related processing technologies in daily life, such as industry, medical, aviation, military and other fields that play an important role. Image preprocessing technology plays an important role and has a huge impact on human life.

The preprocessing flow chart is shown in Figure.1. The preprocessing stage should reduce the influence of factors

such as illumination, imaging system, and external environment on the input image. In the subsequent process, the quality of the image should be improved. Different face images should be extracted under the same conditions. The process of face image preprocessing mainly includes basic tasks such as image enhancement and image edge detection. Image enhancement is to eliminate the noise generated by

light and external factors on the image, and use various filtering methods to perform image segmentation technology to improve the quality of the image. The edge of the image is to allow the input image that has been subjected to external factors to have a good visual effect. Let the image be displayed more clearly.

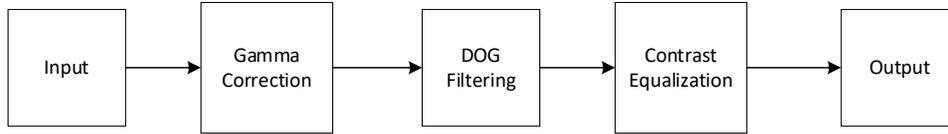


Figure 1. Pretreatment process

4. Face image grayscale transformation

4.1. Overview of image enhancement

After inputting the image, the image will produce various noises and changes due to various external factors, which will greatly reduce the quality of the image. Need to perform various transformations of the image to change the sharpness of the image, the important features in the image are effectively selected, and unnecessary features are weakened, so as to facilitate the analysis and processing of the image. The image restoration technology is to understand the problem of image quality degradation. It is necessary to establish a 'quality reduction model, and then use this model to restore the original image. In comparison, the two types of methods have overlapping parts, although the enhancement has recovery characteristics. But it can highlight useful information more, and can make the image can be partially restored. Different problems use different methods.

4.2. Grayscale transformation

Grayscale transformation can change and enhance the dynamic range of different images, and expand the contrast of images, the characteristics of the image will become more obvious. Grayscale The most direct way to process images. It is also one of the most important components of digital software. The purpose of gray scale transformation is to transform a specific selected target according to a specific transformation The gray value transformation can also improve the quality, allowing the visual effect is better. Grayscale transformation is also called image stretching and contrast enhancement. For example, to improve the clarity of a specific detail, you need to change the gray value or expand (compress) a section of it.

The ways of image contrast enhancement are generally divided into two categories:

(1) Direct contrast enhancement

(2) Indirect contrast enhancement. And Grayscale transformation direct enhancement. Direct contrast uses the gray value function between input and output $T(D)$ Relationship, assuming that the gray value of the pixel is $D = f(x, y)$. After processing, the gray value is transformed into $D' = g(x, y)$, Gray-scale enhancement expression formulas shown in 1.

$$g(x, y) = T[f(x, y)] \text{ Or } D' = T(D) \quad (1)$$

From the above gray scale transformation formula, it can be known that when the function is determined, each point in the image will be determined accordingly, and the

enhancement method will also be determined. In the same way, the gray scale transformation function is different from other transformation functions. Even if it is obtained from the same image, the result is still different. When using the grayscale transformation, the processing purpose must be clarified. The purpose of the grayscale transformation is to increase the pixel range after the image transformation and expand the contrast. The result is to make the image clearer, more detailed and easier to be recognized.

There is and only after the grayscale difference value of two adjacent pixels reaches a certain degree, can the vision be clearly distinguished. If the gray value D changes in a small range, it is difficult to distinguish clearly. But proceed after the transformation, D the scope of application will be expanded, the visual effect will increase, and the image will be easier to distinguish. As a result, the difference in brightness between the original image and the background becomes larger, so that the image that cannot be distinguished by the human eye at the beginning can be detected more easily, and the definition becomes more improved.

Therefore, due to the gray value D is in the interval k Inside The value of is different, transform Of After that, the gray value D is in Within the interval and also meeting with k Change and change, the result of the change is increased contrast. If Continuous image the difference in the gray value detected in is very small, but the range of quantization value is at the same level, then in Middle image meeting disappear, the result is that grayscale enhancement cannot achieve the effect. It is difficult to highlight the target image.

4.2.1. Linear gray scale transformation

The purpose of linear grayscale transformation is to change the pixels according to the change of linear grayscale function. When the light is insufficient or the light intensity is too high, the gray value will be limited to a small range, there will be blurry, no gray level images. In this way, a linear function can be selected to improve and expand the pixels of the image to make the sharpness of the image more obvious.

If the gray levels of most pixels in the image are distributed in the $[a, b]$ interval, and a small part exceeds this interval, in order to change the enhancement effect of the image, use 2 The transformation relationship.

$$g(x, y) = \begin{cases} d, & f(x, y) > b \\ \frac{d-c}{b-a} [f(x, y) - a] + c, & a \leq f(x, y) \leq b \\ c, & f(x, y) < a \end{cases} \quad (2)$$

The gray value is completely transformed according to the linear transformation function, $f(r)$ is a one-dimensional linear function, and its general formula is:

$$f(r) = a \times r + b \quad (3)$$

In the above general formula, “a” is the slope and b is the intercept on the y-axis.

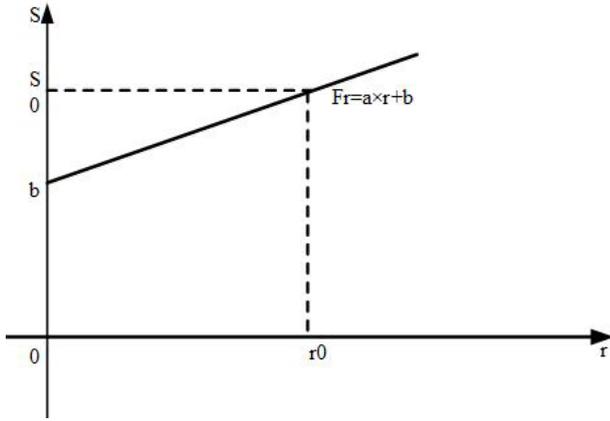


Figure 2. Linear gray scale transformation diagram

According to the image, the following conclusions can be drawn:

- 1) When $a > 1$, the contrast increases.
- 2) When $a < 1$, the contrast Less small.
- 3) When $a = 1$ and $b \neq 0$, the gray value moves up, the sharpness of the image changes Bright; the image gray value shifts down, the image The clarity of darken.
- 4) When $a = 1$ and $b = 0$, Original image and transformed the image is the same.
- 5) When $a = -1$, $b = 255$, Transformed the image grayscale is reversed.
- 6) When $a < 0$, $b > 0$, the brightness of the image is reversed.

The reverse transformation expression is:

$$g(x, y) = 255 - f(x, y) \quad (5)$$

If you enter image $f(x, y)$ the gray scale range is $[z_1, z_2]$, the image after linear transformation $g(x, y)$ the gray value range is $[z_1', z_2']$, it is possible to use the proportional relationship of triangles to find the linear transformation formula. Therefore, $f(x, y)$ versus $g(x, y)$ There is the following relationship:

$$g(x, y) = z_1' + \frac{z_2' - z_1'}{z_2 - z_1} (f(x, y) - z_1) \quad (6)$$

Due to $[z_1', z_2']$ much larger than $[z_1, z_2]$, even if the number of pixels before and after transformation is the same, but the gray scale difference and contrast between different pixels will be enlarged. If the gray level difference between the background and the target image is small, it cannot be clearly distinguished in the interval $[a, b]$.

4.2.2. Non-linear gray-scale transformation

Use some non-linear functions, if the logarithmic function, exponential function, etc. are supposed to be mapping functions, the corresponding non-linear transformation of the image can be performed. The following is the general formula of logarithmic transformation.

$$g(x, y) = a + \frac{\ln[f(x, y) + 1]}{b \ln c} \quad (7)$$

The graph of logarithmic transformation is shown in Figure

3.

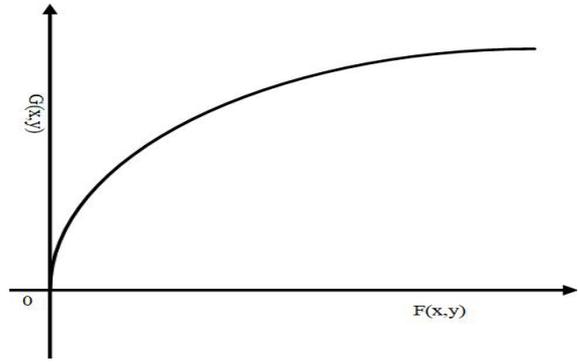


Figure 3. Logarithmic transformation curve

The characteristic of the logarithmic function is to realize the gray scale expansion (compression), the function is to compress the low expansion degree image to the high gray value image, and present a good visual effect. For this feature, logarithmic transformation is applied to enhance the dark details in the image. In the above formula, a, b, and c are adjustable parameters, which are used to adjust the shape and position of the curve. If the dynamic range of the original input image is too large, beyond the allowable dynamic range, at this time, logarithmic transformation can be used to reduce the dynamic range. The following code demonstrates the basic method of image gray scale logarithmic transformation. The function $\log()$ manipulates the elements in the input image matrix, but it can only process double-type matrix data. In addition, deal with as much as possible the values that may be out of range.

Run the above program code, the high-brightness part of the display image is shrunk in the narrow space of the histogram, while the low-gray value is spread out. As shown in Figure 4.



a).Original Image



b).Log Transformations v=10



Log Transformations v=100



d).Log Transformations v=200

Figure 4. Image after logarithmic transformation

The general formula for index change is:

$$y = b^{c(x-a)} - 1 \quad (8)$$

In the above general formula, the values of b and c are the shape of the control function, and the value of “a” is the position of the control function. The exponential transformation function is shown in Figure 5.

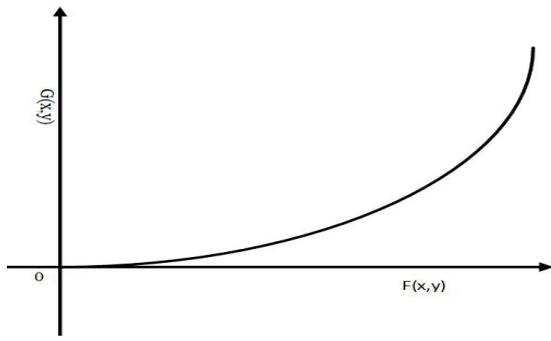


Figure 5. Exponential transformation curve

The purpose of exponential transformation is to expand the gray scale of the high range value and compress the gray scale of the low range value. It can be seen from the above image that the slope of the function image is relatively stable, indicating that the gray-value pixels distributed in the low range in the input image are mapped to the output image and compressed in a small interval; on the contrary, in the input image the gray-scale pixels distributed in the high range are mapped on the output image and spread in a large interval. But compared with the power function, the effect of the exponential transformation function is more obvious. By selecting appropriate parameters, the image can be obtained with a more obvious contrast, and the range can be expanded. As shown in Figure 5, the above face image is a reality with unreasonable parameters. The following code is to implement exponential transformation.

Run Face image after as shown in Figure 6.

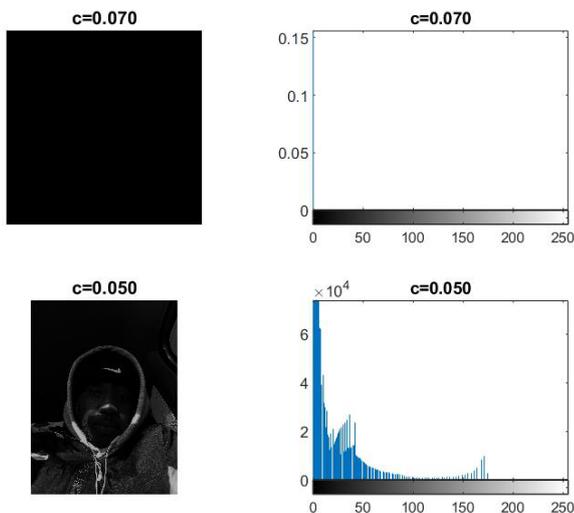


Figure 6. Exponential transformation curve

According to the basic principle of gray linear transformation, combining linear transformation and non-linear transformation, the face image can be compared as follows.

The face image under linear gray scale transformation is shown in Figure 7.

It can be seen from the linear gray scale transformation performed in Figure that the comparison of face images in different gray scale ranges shows that the sharpness of the face images obtained in different states is different.

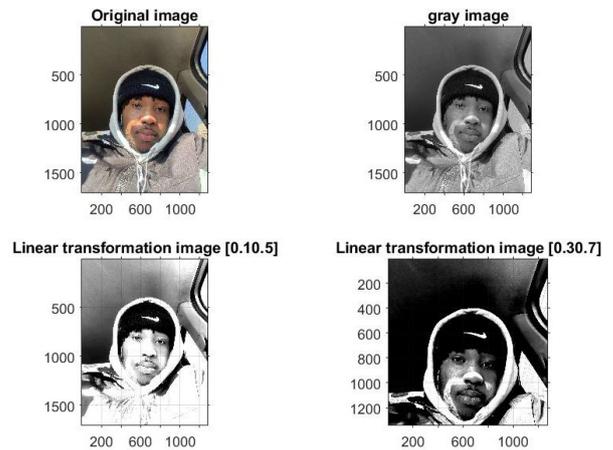


Figure 7. Linear gray scale transformation

4.3. Histogram equalization

Histogram Equalization is a computer image processing technique used to improve contrast in images. It accomplishes this by stretching out the intensity range of the image. This method usually increases the global contrast of images when its usable data is represented by close contrast values. The purpose of histogram equalization is to correct the input image into a uniform and smooth histogram. If the image is processed by histogram equalization, the result will be the histogram of is flat (balanced distribution probability), and

the image looks clearer than the original image. $s = T(r)$ The purpose of the function is to make the histogram change flat, so that the transformed image maintains the monotony nature of black and white. And the range of transformation is similar to the original range to make this variable darker or brighter, so the following two points must be specified.

- 1) $n \ 0 \leq r \leq 1$ in, $T(r)$ Is monotonically increasing, and $0 \leq T(r) \leq 1$.
- 2) Inverse transformation $r = T^{-1}(s)$ Is also monotonically increasing, $0 \leq s \leq 1$.

This allows for areas of lower local contrast to gain a higher contrast.



Figure 8. Input Image

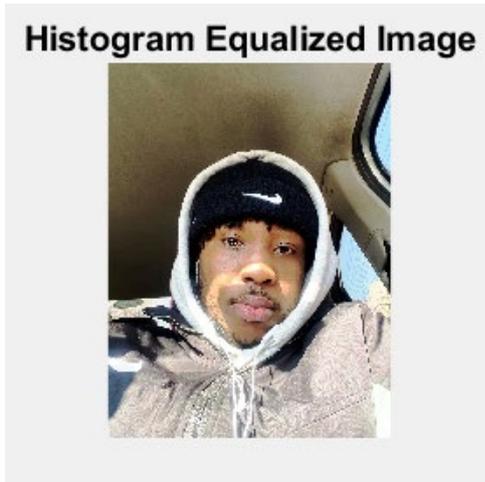


Figure 9. Output image

After comparing Figure 8 and Figure 9 As can be seen, Histogram equalization makes dark images brighter and details clearer.

4.3.1. The advantages of histogram specification

An image histogram is a type of histogram that acts as a graphical representation of the tonal distribution in a digital image. It plots the number of pixels for each tonal value. By looking at the histogram for a specific image a viewer will be able to judge the entire tonal distribution at a glance. Image histogram is present on many modern services. Photographer can use them as an aid to show the distribution of tones captured, and whether image detail has been lost to blown out highlights or blacked-out shadows. This is less useful when using a raw image format, as the dynamic range of the displayed image may only approximate that in the raw file. The horizontal axis of the graph represents the tonal variations, while the vertical axis represents the total number of pixels in that particular tone. The left side of the horizontal axis represents the dark areas, the middle represents mid-tone values and the right-hand side represents light areas. The vertical axis represents size of the area (total number of pixels) that is captured in each one of these zones. Thus, the histogram for a very dark image will have most of its data points on the left side and center of the graph.

Conversely, the histogram for a very bright image with few dark areas and /or shadows will have most of its data points on the right side and center of the graph. The histogram plots the number of pixels in the image (vertical axis) with a particular brightness or tonal value (horizontal axis) Figure 10 shows an image and the histogram of it.

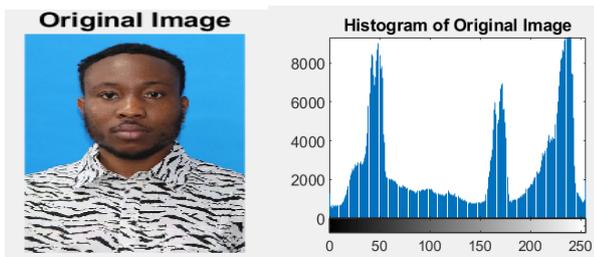


Figure 10. Original Image and Histogram

5. Image edge detection

5.1. Image normalization

Due to the influence of external factors, the brightness and size of the image will change differently during the process of

collecting the face image. Therefore, the normalization method is an important process that can effectively improve these factors. Normalization is also for feature extraction and better processing of face recognition. Generally speaking, there are two types of image normalization: size normalization and gray-scale normalization.

5.1.1. Size normalization

The main purpose of size normalization is to transform facial expression images into uniform sizes. The specific operation methods are as follows:

(1) Calibration feature points, calibration two eyes and the nose three special Feature points, mainly with a mouse to manually mark to obtain the coordinate values of the three characteristic points.

(2) Rotate the image according to the coordinate values of the left and right eyes to ensure the face the direction of consistency. Set up the distance between the eyes is d , where the point is.

(3) According to facial features point Too for center, about Each, Vertical direction Take $0.5d$ and $1.5d$ rectangular areas for Crop.

(4) Yes, the expression is processed for size normalization, the purpose is to extract features.

The face model diagram is shown in figure 11.

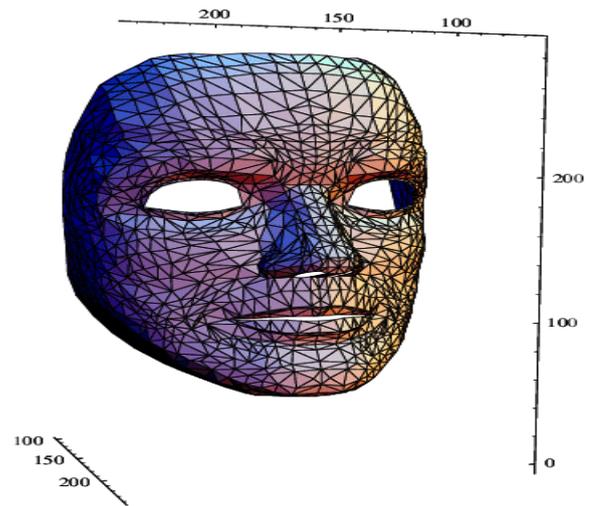


Figure 11. Facial geometry model

The purpose of size normalization is to unify the format of all images, and obtain a normalized image after processing and improving the image. Therefore, some basic processing methods such as cropping and adjustment of the input image are required. The following is a simulation program for size normalization.

The simulation diagram of the normalized face image size is shown in Figure 12.



Figure 12. Normalization of face image size

5.1.2. Gray normalization

Gray normalization it can also be said to be histogram equalization, the purpose is to count the number of pixels in each gray level in the histogram, and to make these pixels evenly distributed with each gray level.

Gray normalization is to extend the gray distribution in the original image to an image with the entire gray level by using a gray-scale stretching method. As will be used here, the face image is an 8-bit grayscale image, and there should be 256 grayscale levels. However, due to the influence of the illumination and location during acquisition, the grayscale levels of the image are concentrated in one or several gray levels, at this time, the image is expanded to 256 gray levels by gray-scale stretching. The normalization formula of gray scale transformation is shown in 9.

$$N(a,b) = \frac{I(a,b) - \min}{\max - \min} \cdot 255 \quad (9)$$

In the above general formula Medium, I and N represent the gray value of the transformed image, and min Represents the minimum gray value of the input image, max is the maximum gray value of the input image. The simulation procedure of image gray normalization is as follows.

The gray-scale normalized simulation diagram of the face image is shown in Figure 13.



Figure 13. Gray normalization of face image

5.2. Overview based on edge detection

In general image processing, edges become contours. It represents the boundary line of the object, and also represents the characteristic elements that can vividly depict the line. These elements are just like the lines in the sketch. In addition, color, brightness, etc. are all important characteristic elements. For example, the charts, illustrations, etc. used in daily life are all basic ways to describe the edge line. Although this will seem monotonous, we can still clearly see the specific picture.

The performance of the edge on the image is local and discontinuous, but it is one of the most important segmentation methods in image preprocessing, which can effectively obtain edge points and segment specific objects.

There is also a common method based on edge segmentation: edge tracking, which uses serial technology to achieve tracking. But this method is different from edge connection. Common ones include Hough and Sobel transforms. The purpose of these two transformations is to take the generated unordered collection as the input object, and then output this object as an ordered collection. In short, Edge detection uses local information to judge and determine the edge of an image, while edge tracking is related to processing based on pixels.

5.2.1. Threshold segmentation

Threshold segmentation Again for Segmentation technology, Correct Object and background play an important role in segmentation. Because the visual effects are intuitive, clear and easy to achieve, they occupy an important position in image segmentation technology. Among them, the gray threshold method is a commonly used method.

The basic purpose of the gray threshold method is to divide the gray level of the face image according to different levels, and then set the gray threshold effectively to segment the image boundary. There are generally two methods: Banalization and threshold segmentation. The image contains a lot of noise, set a certain threshold to t, and divide the image data into two parts: pixel groups larger than t and smaller than t. The input image is $f(x, y)$, the output image is $f'(x, y)$. General functional as shown in 10.

$$f'(x, y) = \begin{cases} 1, f(x, y) \geq t \\ 0, f(x, y) < t \end{cases} \quad (10)$$

Or

$$f'(x, y) = \begin{cases} 1, f(x, y) \leq t \\ 0, f(x, y) > t \end{cases} \quad (11)$$

The above general formula is threshold segmentation processing, purpose Yes Find the threshold t, and use t to divide the image $f(x, y)$ Divided into Two areas.

Since the desired image and the actual image cannot necessarily be distributed between the two grayscale ranges, it is necessary to use more than two (including two) thresholds for extraction. Hypothesis, use (t_1, t_2) the range of to take the threshold, the following is the general function of the banalization process.

$$f(x, y) = \begin{cases} 1, t_1 \leq f(x, y) \leq t_2 \\ 0, \end{cases} \quad (12)$$

$$f(x, y) = \begin{cases} 1, \\ 0, t_1 \leq f(x, y) \leq t_2 \end{cases} \quad (13)$$

Generally, it must be assumed that when threshold segmentation is used to make grayscale images, that is to say, there must be a single peak grayscale composition in the image. The gray value between the elements in the image is related to the height, but there is a big difference with the gray value on both sides of the junction. If the images meet the above conditions, the resulting image is composed of single-peak histograms. If the distribution size and number are the same, but the means are far apart and the variance is small, the result is a bimodal histogram. Generally, this kind of image will use the threshold segmentation method.

Here select the different changes of the input face image when the threshold is 90, 150, the following is the program code

The resulting face threshold image is shown in Figure 14.



Figure 14. Threshold segmentation image of human face

5.2.2. Laplacian of Gaussian (Log)

Laplace the purpose of the processing method is to suppress noise and prevent it from affecting edge detection. For example, the first derivative is used in the Canny operator, and the Log operator to use the second derivative, because the Laplacian operator is more sensitive to noise when detecting edges, so the Gaussian smoothing template must be used for smoothing before edge detection. Suppose first try to get the Laplacian of Gaussian $\Delta G_\sigma(x, y)$, and then convolve it with the input image to complete the desired operation, then an operator of Log is obtained, or its convolution kernel is defined as:

$$LoG = \Delta G_\sigma(x, y) = \frac{\partial^2}{\partial^2 y} G_\sigma(x, y) + \frac{\partial^2}{\partial^2 x} G_\sigma(x, y) \quad (14)$$

$$G_\sigma(x, y) = \frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} e^{-\frac{x^2 + y^2}{2\sigma^2}}$$

From the above convolution formula, the first-order and second-order function graphs as shown in Figure 15 can be obtained.

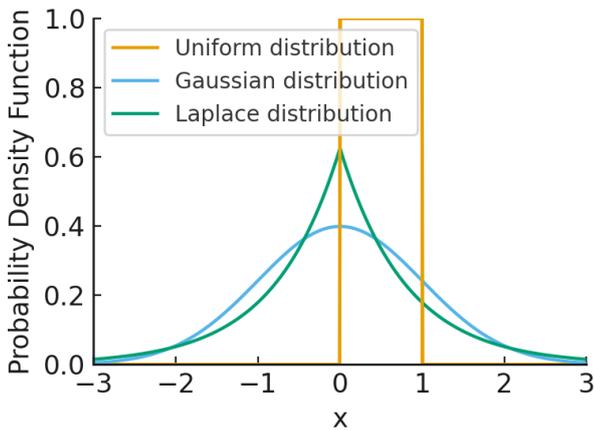


Figure 15. Gaussian Laplacian function graph

The general template of the common two-dimensional Log operator is expressed as follows.

$$\begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix} \quad (15)$$

Generally, any convolution kernel template can be created by Log operator get. But in the calculation process, it is

necessary to ensure that the sum of all elements in the general formula is zero.

5.2.3. Gauss difference operator (DoG)

The Gaussian difference operator is similar to Laplacian in processing, and Gaussian convolution template is still used in the processing to smooth the image and reduce noise. Taking DoG as an operator or a convolution kernel, its general definition is:

$$DoG = G_{\sigma_1} - G_{\sigma_2} = \frac{1}{\sqrt{2\pi}} \left[\frac{1}{\sigma_1} e^{-\frac{x^2+y^2}{2\sigma_1^2}} - \frac{1}{\sigma_2} e^{-\frac{x^2+y^2}{2\sigma_2^2}} \right] \quad (16)$$

The difference of Gaussian function is shown in Figure 16 below.

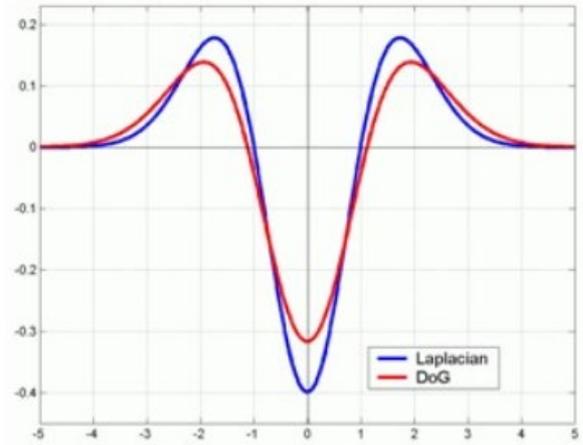


Figure 16. Gaussian difference function graph

DoG the operator is actually a band-pass filter, its role is to remove the higher and lower frequency parts of the signal. The so-called high-frequency part is: noise; the low-frequency part is: non-edge information area (homogenization). When using the Dog operator again, it is necessary to calculate to ensure that the sum of the elements is zero.

Comparing the curve of the DoG function with the curve of the Log function, the DoG efficiency is high. Generally, the DoG function is used instead of the Log function, because the effect of DoG in reducing noise interference is slightly better.

5.2.4. Canny edge detection

In the late 1960s and 1980s, it was the golden age of the development of edge detection theory. The Log operator and DoG operator given above were also born during this period. The effect of edge detection using the gradient method is particularly obvious, but it is not obvious for smooth edges. Considering the strong influence of noise in the Laplacian operator, the edge detection method has been continuously and systematically After verification and demonstration, the Canny edge detection algorithm, which is now widely used, and the three edge detection criteria of the Canny criterion, which is popular in image processing, are finally proposed. Its purpose is to use numerical calculation methods to find the most suitable filter under the assumption of the signal and filter, and to compare the performance of the filter. The Canny guidelines specifically include the following three contents:

- 1) Good detection effect
- 2) Accurate positioning of edge detection
- 3) Low response times to the same edge

There are five steps in the implementation of Canny's algorithm:

- 1) Use a two-dimensional Gaussian template for convolution to eliminate noise.
- 2) Use the finite difference calculation of the first-order partial derivative.
- 3) Suppress the gradient of non-maximum value.
- 4) The dual-threshold algorithm detects and connects edges.
- 5) Use multi-scale technology to optimize the processing results.

From the above five steps may feel a bit cumbersome and complicated, but it is still very convenient in the MATLAB algorithm, and the commonly used edge () function is needed. The following is a comparison of several algorithms with program code

Run the above code, the face images displayed by different algorithms are shown in Figure 17.

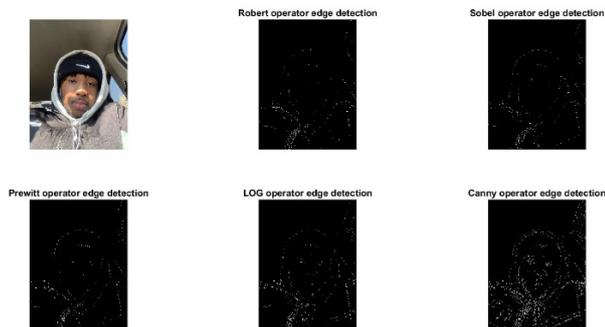


Figure 17. Comparison of different operators

It can be seen from Figure 17 that the effect of the face image obtained by the edge detection of the canny operator is the best. It can clearly outline the characteristics of each point in the face recognition process.

6. Chapter 6 Results and Discussion

6.1. MATLAB Tools for Face Processing

MATLAB provides convenient functions for image processing and computer vision. The Computer Vision Toolbox includes the vision.Cascade Object Detector system object, which implements the Viola-Jones algorithm for detecting faces, eyes, nose, mouth, or upper body.

The typical workflow is:

1. Create a vision.Cascade Object Detector object and choose the desired model (e.g., Frontal Face CART).
2. Pass an input image to the detector to obtain bounding boxes.
3. Crop and preprocess the detected face region using the techniques described in previous chapters.

The detector has several important properties such as classification model, minimum and maximum object size, scale factor, merge threshold, and optional region of interest. These parameters control detection accuracy and speed.

6.2. Example MATLAB Code

Simple MATLAB code for face detection and cropping is as follows (conceptually):

- Create the detector: detector = vision.Cascade Object Detector;
- Read the input image: I = imread('face.jpg');
- Run detection: bbox = step(detector, I);
- Draw rectangles and crop the first detected face.

- Convert the cropped face to grayscale, apply histogram equalization, and filter with a median or Wiener filter.
- Optionally perform edge detection using edge (Igray, canny).

6.3. MATLAB result for Face Recognition

Face recognition technology is widely used in biometric user authentication processes. This authentication process can be done via human faces features extraction and the classification processes. Here, we will see an example of face recognition using MATLAB code. The code is shown below: The Result is shown in FIG.18



Figure 18. Face Detection.

7. Conclusion

This research mainly introduces the prospects and prospects of face recognition, and the method of image preprocessing in face recognition. This research mainly focuses on related research on image preprocessing, such as grayscale transformation, histogram equalization, high-pass, low-pass, smoothing, sharpening, banalization, wavelet transform, edge detection, etc. It has nothing to do with the face image in the input. The process of pre-processing the information, so that the image is enhanced, the details of the image are more prominent, and even the facial amount is positioned by the projection method.

In the preprocessing process, through the comparison of several methods, it is known that not every processing method is suitable and can highlight the face image in a complete image. There are external factors and technical deficiencies. For example, there are many difficulties in preprocessing in face recognition, which will be highlighted in future research and the next step of work and exploration will be carried out:

1. Establish face recognition database samples to realize face recognition completely.
2. A lot of learning sample knowledge, get a higher recognition rate.
3. Cancel the external factors (restrictions of the background), and can be more adapted to complex scenes.

This work has presented a survey about Image preprocessing in face recognition; the expected goal can be achieved initially. However, there will be some problems in the preprocessing process. The image quality will be higher after the laxative transformation in the frequency domain and spatial domain enhancement, but corresponding information will be lost. After the histogram is equalized, the abrupt

unevenness and the gray level will be changed. Cut back. Although wavelet transform can be effective when the known noise frequency and frequency band are separated from each other, it is not widely used. In the follow-up research, it is hoped that these shortcomings can be improved to make the image preprocessing more prominent. Face recognition is also resulting in other dares, like expression recognition or body motion recognition. Overall, face recognition techniques and the emerging methods can see use in other areas. Therefore, it isn't just a unresolved problem but also the source of new applications and challenges.

Acknowledgements

I am grateful to God, my parents, and my friends for their constant support and encouragement. This paper on face recognition within image preprocessing was completed through sustained research, clear communication, hard work, and dedication. I offer special thanks to my mentor, whose guidance, feedback, and learning materials greatly deepened my understanding of Telecommunication Engineering and strengthened this manuscript. Their support was instrumental in bringing this work to completion.

References

- [1] Adini, Y., Moses, Y., & Ullman, S. (1997). Face recognition: The problem of compensating for changes in illumination direction. *IEEE TPAMI*, 19(7), 721–732.
- [2] Bach, F., & Jordan, M. (2002). Kernel independent component analysis. *Journal of Machine Learning Research*, 3, 1–48.
- [3] Bledsoe, W. W., & Chan, H. (1965). A man-machine facial recognition system – Some preliminary results. Panoramic Research, Inc.
- [4] Brunelli, R., & Poggio, T. (1993). Face recognition: Features versus templates. *IEEE TPAMI*, 15(10), 1042–1052.
- [5] Crunchant, A. S., Egerer, M., & Loos, A. (2020). Automated face detection for occurrence and occupancy estimation in chimpanzees. *American Journal of Primatology*, 79(3), 1–12.
- [6] Jain, A. K. (2006). *Digital image processing foundations*. Beijing: Tsinghua University Press.
- [7] Kraus, O. Z., & Frey, B. J. (2016). Computer vision for high content screening. *Critical Reviews in Biochemistry and Molecular Biology*, 51(2), 102–109.
- [8] Phillips, P., Moon, H., Rizvi, S., & Rauss, P. (2000). The FERET evaluation methodology for face-recognition algorithms. *IEEE TPAMI*, 22(10), 1090–1104.
- [9] Shan, S., Gao, W., Cao, B., & Zhao, D. (2003). Illumination normalization for robust face recognition. *Proceedings of AMFG*, Washington, DC.
- [10] Wang, H., Li, S., & Wang, Y. (2004). Face recognition under varying lighting conditions using self-quotient image. *IEEE AFGR*, 819–824.
- [11] H. Wang, S. Li, and Y. Wang, “Face recognition under varying lighting conditions using self-quotient image,” in *Proc. IEEE Int. Conf. Autom. Face Gesture Recognition*, 2004, pp. 819–824.
- [12] A. S. Crunchant, M. Egerer, A. Loos et al., “Automated face detection for occurrence and occupancy estimation in chimpanzees,” *American Journal of Primatology*, vol. 79, no. 3, pp. 1–12, 2020.
- [13] Q. Zhao, S. S. Ge, M. Ye, S. Liu, and W. He, “Learning saliency features for face detection and recognition using multi-task network,” *International Journal of Social Robotics*, vol. 8, no. 5, pp. 709–720, 2016.
- [14] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71-86, 1991.
- [15] H. Wang, S.Z. Li and Y. Wang. Face recognition under varying lighting conditions using self-quotient image. In *IEEE International Conference on Automatic Face and Gesture Recognition (AFGR)*, pages 819-824. 2004.
- [16] Y. Adini, Y. Moses, and S. Ullma. Face recognition: the problem of compensating for changes in illumination direction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:721–732, 1997.
- [17] N. Ahmed, T. Natarajan, and K. R. Rao. Discrete cosine transforms. *IEEE Transactions on Computers*, 23:90–93, 1974.
- [18] F. Bach and M. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, 2002.
- [19] M. Ballantyne, R. S. Boyer, and L. Hines. Woody bledsoe: His life and legacy. *AI Magazine*, 17(1):7–20, 1996.
- [20] W. W. Bledsoe. Some results on multicategory patten recognition. *Journal of the Association for Computing Machinery*, 13(2):304–316, 1966.
- [21] W. W. Bledsoe. Semiautomatic facial recognition. Technical report sri project 6693, Stanford Research Institute, Menlo Park, California, 1968.
- [22] W. W. Bledsoe and H. Chan. A man-machine facial recognition system- some preliminary results. Technical report pri 19a, Panoramic Re- search, Inc., Palo Alto, California, 1965.
- [23] R. Brunelli and T. Poggio. Face recognition: Features versus tem- plates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(10):1042–1052,
- [24] J. S. Bruner and R. Tagiuri. The percepton of people. *Handbook of Social Psychology*, 2(17), 1954.
- [25] Wang, S.Z. Li and Y. Wang. Face recognition under varying lighting conditions using self-quotient image. In *IEEE International Conference on Automatic Face and Gesture Recognition (AFGR)*, pages 819-824. 200
- [26] F. Raphael, B. Olivier, and C. Daniel. A constrained generative model applied to face detection. *Neural Processing Letters*, 5(2):11–19, April 1990.
- [27] P. Phillips, H. Moon, S. Rizvi, and P. Rauss. The ferret evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1090–1104, October 2000.
- [28] urk, M., & Pentland, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1), 71–86
- [29] Liu, W. (2007). *Digital image acquisition and processing*. Beijing: Electronic Industry Press.