Volume 3, Issue 4, April 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Appliance of Biometric and Median Filter Algorithm to Identify Lookalike Faces

Mrs. Vasudha Vashisht* *Assistant Professor from Lingaya's University(India)*
Mr. Tanupriya Choudhury *Assistant Professor from Lingaya's University(India)*
Prof.Vivek Kumar *Principal DCTM-Palwa, India.*
Mr. Himanshu Srivastava *Undergraduate degree CSE from G.B.T.U.(India)*

Summary -- The objective of this paper is to present an accurate technique to identify two lookalike faces based on the concepts of biometrics and the Median Filter Algorithm. This paper gives some useful insight on the application of median filter and biometrics, where biometrics refers to methods for uniquely recognizing humans based upon one or more intrinsic physical or behavioral traits. It is also used to identify individuals in groups that are under surveillance. Biometric recognition based on human characteristics for personal identification has attracted great attention. The performance of biometric systems highly depends on the distinctive information in the biometrics. However, lookalike faces having the closest genetics-based relationship are expected to have maximum similarity between their biometrics. Identifying such lookalike faces is a challenging problem for most automatic biometric systems. Physiological biometrics has been used to operate on still images of the face. The challenge is to get rid of high degree of noise in the images before any further processing is done. The median filter is used for this purpose in this project which is a non-linear digital filtering technique. The project has been successfully implemented with an efficiency of about 85%.

Key words--Lookalike Face Identification, Biometric, Median Filter

I. INTRODUCTION

Human beings have been using physical uniqueness to identify and recognize each other for eg facial appearance, voice, way of walking, etc. for years. With the technological advances, biometrics has become an up-and-coming technology for identifying individuals using their biological traits. This technology makes use of the verity that each person has some precise unique physical traits that are one's characteristics which can't be lost, borrowed or stolen. By using biometrics it is possible to corroborate or set up identity based on "who the individual is", rather than by "what the individual possesses" (e.g., an ID card) or "what the individual remembers".

The aim of the project is based at distinguishing two identical twins. To achieve this, we have used the concepts of biometrics and an algorithm known as Median Filter Algorithm and implemented this concept using Java Swings and SQL Server 2000.

1.1 Biometrics

Biometrics refers to methods for uniquely recognizing humans based upon one or more intrinsic physical or behavioral traits. In information technology, in particular, biometrics is used as a form of identity access management and access control. It is also used to identify individuals in groups that are under surveillance.

Biometric characteristics can be divided in two main classes:

(i) Physiological are related to the shape of the body. Examples include, but are not limited to fingerprint, face recognition, DNA, hand and palm geometry, IRIS recognition, which has largely restored retina, and odor/scent.

(ii) Behavioral are related to the behavior of a person. Examples include but are not limited to typing rhythm, way of walking and voice. Some researchers have coined the term behaviometrics for this class of biometrics.

The project uses the physiological class of the biometric as by using the images of the faces to be identified for their alikeness.

1.2 The Median Filter Algorithm

The median filter is a spatial filter, and it restores the center value in the window with the median of all the pixel values in the window. The kernel is usually square but can be any shape.

In image processing it is usually necessary to perform a high degree of noise reduction in an image before performing higher-level processing steps, such as edge detection. The median filter is a non-linear digital filtering technique, often used to take away noise from images or other signals. The idea is to scrutinize a sample of the input and decide if it is representative of the signal. This is performed using a window consisting of an odd number of samples. The values in the window are sorted into numerical order; the median value, the sample in the center of the window, is selected as the output. The oldest sample is discarded, a new sample acquired, and the calculation repeats [3, 4, 7].

Median filtering is a common step in image processing. It is particularly helpful to reduce speckle noise and salt and pepper noise. Its edge-preserving nature makes it useful in cases where edge blurring is disagreeable.

1.2.1 The optimization problem

The main problem of the median filter is its high computational cost (for sorting N pixels, the secular complexity is O(N•log N), even with the most efficient sorting algorithms). When the median filter must be carried out in real time, the software implementation in general-purpose processors does not usually give good results. ASICs have been obligatory traditionally. However, they imply a limited functionality due to their predefined architecture. Also, the price for application as well as the construction time is usually prohibitive. [1]

Reconfigurable computing architectures are sufficiently supple so that new operations can be executed in the existent hardware, and they are quite quick for real-time execution. Moreover, the price/performance ratio of these systems crafts them a broadly competitive alternative to ASICs. FPGAs have been identified as the natural platform for CCMs due to their reprogram capability.

For these reasons, it is believed that FPGAs are a good alternative to achieve the solution of this problem: the real-time implementation of median filter. This implementation allowed us to use this operation within automated visual inspection systems, maintaining the requirements of real-time operation and abridged cost so that these systems can be introduced into an industrial environment.

II. PROBLEM DESCRIPTION

The problem of identifying lookalike faces using biometrics and the median filter algorithm comprises of following modules:

(i) Search Image
(ii) Calculate Data
(iii) Accessing database
(iv) Construct grayscale Image
(v) Comparison of Images

These modules are small part of this project. This plays a very important role in the project and in coding concepts. In Software Engineering concept, we treat it has a small part of a system but whereas in our programming language it is a small part of the program, which we also called as function in, some cases which constitute the main program.

(i) Search Image: Search Image is a module that is considered with browsing the images stored somewhere on the computer. This is done for both the images and taken as input for further calculations.

(ii) Calculation of Data: Calculation of Data includes reading the image, converting it into bytes and then into bits. This module converts the image in binary form.

(iii) Accessing Database: Database here contains the information about image i.e. the path where the images are stored along with a name assigned to them. The names are stored in database.

When the instruction is given to find from the database, it search whether the name of that image is stored in database or not.

(iv) Constructing Grayscale Image: After reading the data, the algorithm known as median filter is applied. As a result of applying this algorithm, a grayscale image is obtained which is the black and white copy of the original image.

(v) Comparison of Images: This module includes comparing the grayscale image of both the inputted images obtained after applying the median filter technique.

III. EXPERIMENTAL CONSIDERATION

3.1 Design Principles

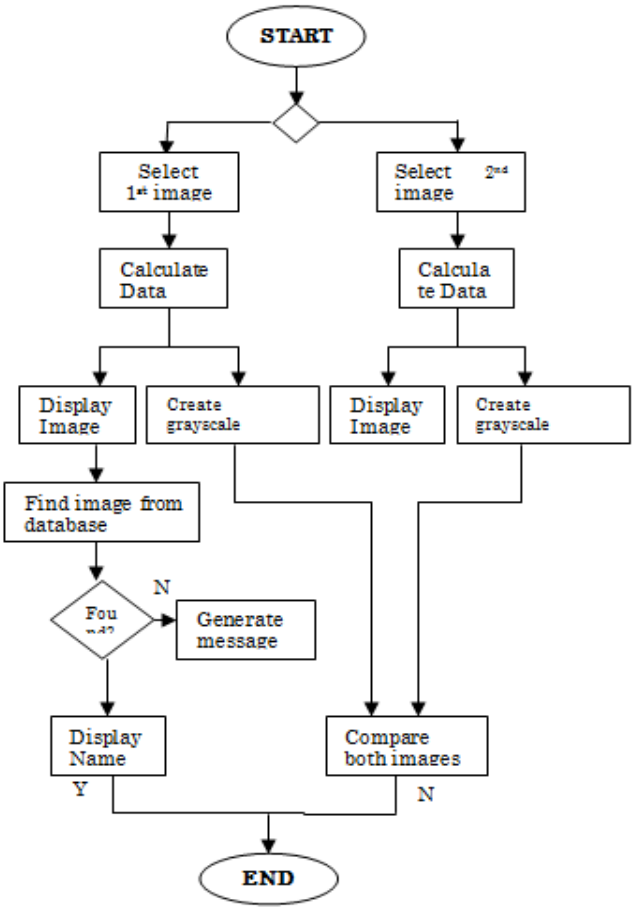

Fig 6 Process Flow Diagram

3.2 Data Flow Diagram

(i) Main Screen Process

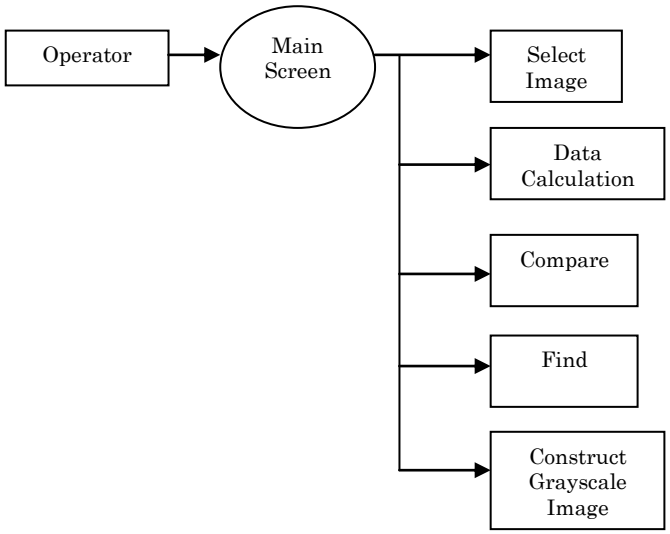

Fig 1the Main Screen Process

This process mainly explains the different options that are available for the operator. Here the selection of any option depends on the operator and he can select whatever screen he wants to. The different options that are available are Browse Image, Calculate data, access database, comparing two images and constructing the grayscale image of original image. [2]

(ii) Data Calculation

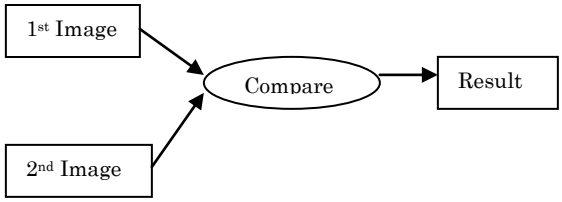

Fig 2 Calculation of data

The interface expects you to browse an image and convert it into the binary image by first converting that image into byte array and then further into bit array.

The image that is selected as input should be in JPEG format. Then this process will encode that image and extract all the pixels that constitute it.

(iii) Construct Grayscale Image

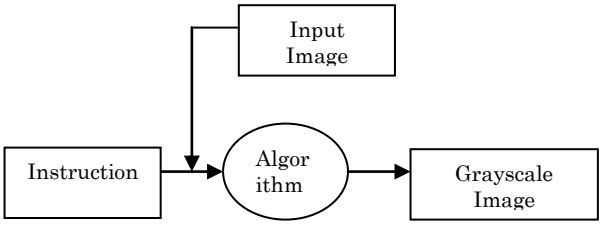

Fig 3 Construction of grayscale image

When the byte array is formed instruction may be given at any moment to implement the algorithm on input and hence the result is given out in Black and White Form i.e. a grayscale image is obtained at the end of the algorithm.
This is the smoothen form of input image.

(*iv) Accessing Database*

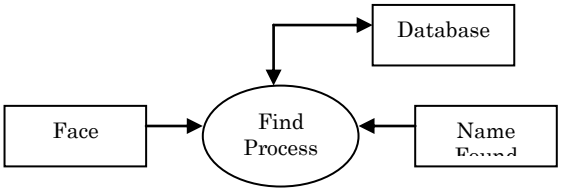

Fig4 Accession of database

A database is made which consist of only the name and path of any image available. When the FIND procedure is applied to the given image/ face, the database is scanned the name for the given image is searched for.
If the name is found then it is displayed on the main screen as a result.

(v) Comparing Images

When the data of both the images has been calculated and the algorithm has been implemented then the grayscale image obtained by algorithm in the end are compared.

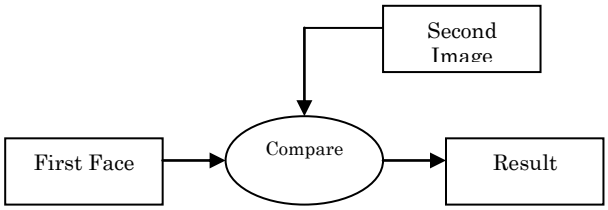

Fig 5 Comparison of grayscale images

If they are similar and match at least 80 percent then the result of comparing is that the both input faces are Lookalike otherwise not matched message is displayed.

3.3 Conceptual Design

(i) Working of Median Filter for the system

Median filter is a spatial filtering operation, so it uses a 2-D mask that is functional to each pixel in the input image. To relate the mask means to centre it in a pixel, evaluating the covered pixel brightness es and determining which brightness value is the median value. Figure presents the concept of spatial filtering supported on a 3x3 mask, where 'I' is the input image and 'O' is the output image. [5]

The median value is calculated by placing the brightness in ascending order and selecting the centre value. The acquired median value will be the value for that pixel in the output image. Figure shows an instance of the median filter application, as in this case, habitually a 3x3 median filter is used[10].

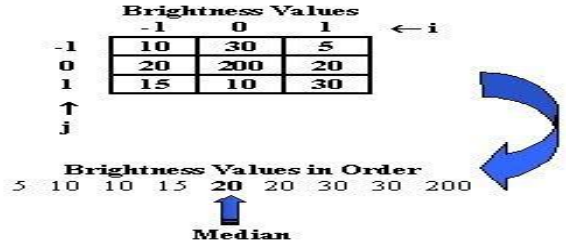

Fig 7 Filtering Example

The median filter considers every one pixel in the image in turn and looks in close proximity neighbors to decide whether or not it is representative of its surroundings. Instead of simply replacing the pixel value with the *mean* of neighboring pixel values, it replaces it with the *median* of those values. The median is intended by first sorting all the pixel values from the surrounding neighborhood into numerical order and then replacing the pixel being measured with the middle pixel value. (If the neighborhood under consideration contains an even number of pixels, the average of the two middle pixel values is used.) Figure 1 illustrates an example calculation.

As can be seen, the central pixel value of 150 is rather deceiving of the surrounding pixels and is replaced with the median value: 124. A 3×3 square neighborhood is used here --- larger neighborhoods will make stricter smoothing. [6]

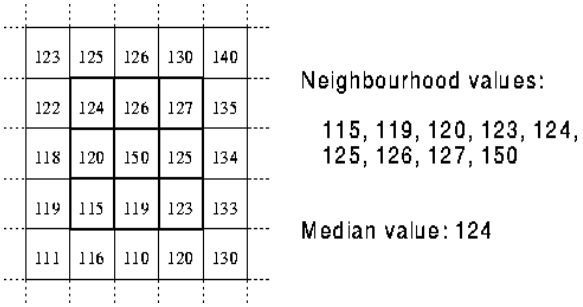

Fig 8 Calculating the median value of a pixel neighborhood[10].

By scheming the median value of a neighborhood rather than the mean filter, the median filter has two main compensations over the mean filter:

The median is a further robust average than the mean and so a single very unrepresentative pixel in a neighborhood will not affect the median value significantly.

Since the median value must actually be the value of one of the pixels in the neighborhood, the median filter does not create new unrealistic pixel values when the filter straddles an edge. For this reason the median filter is much better at conserving sharp edges than the mean filter.

Unlike the mean filter, the median filter is non-linear. This means that for two images *A(x)* and *B(x)*:

$$median[A(x) + B(x]) \neq median[A(x)] + median[B(x)]$$

Fig 9 Original image and its Grayscale version

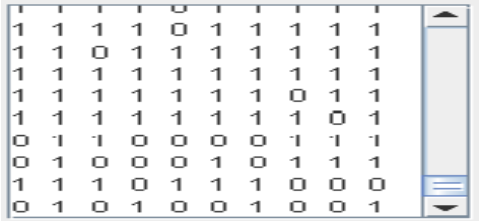

Fig 10 Byte array conversion of fig…

3.4 Overview of the Algorithm

To convert the original image in 2-D byte array in grayscale form following figure shows the steps involved in the median filter algorithm

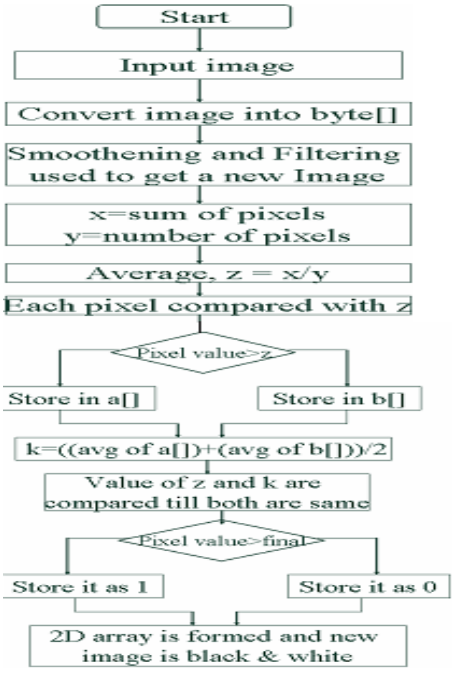

Fig 11 Steps involved in the Median Filter

IV. CONCLUSION

The project aimed at distinguishing the faces of the identical twins. This work was accomplished using the biometric and the median filter algorithm technique. An overall accuracy of 85% has been achieved. This efficiency is much more than that achieved by using other techniques for the discussed task. This work would be highly beneficial in high security zones where entry is to be maintained restricted. The project implements the identification process on still images only. This is an issue that still remains to be tackled. The aim now is to apply the same concept real-time. The idea is to capture real time videos, convert them into images and make comparisons. All this should be achieved in only few seconds, only then this will cater to the security needs of that area.

REFERENCES

[1] Lin, Kwan-Ho; Lam, Kin-Man; Siu, Wan-Chi "A new approach using modified hausdorff distances with eigenface for human face recognition"Proceedings of the 7th International Conference on Control, Automation, Robotics and Vision, ICARCV 2002 Kenichi;

[2] Machida, Kazuyuki; Matsuura, Shingo; Akamatsu, Shigeru "Comparison of racial effect in face identification systems based on eigenface and GaborJet Tanaka", Proceedings of the SICE Annual Conference, SICE Annual Conference 2004, 2004, p 2835-2840, Compendex.

[3] Sengupta K.; Lee P.; Ohya J. "Face posture estimation using eigen analysis on an IBR (image based rendered) database" Pattern Recognition, January 2002, vol. 35, no. 1, pp. 103-117(15), Ingenta.

[4] Kim, Hyun-Chul; Kim, Daijin; Bang, Sung Yang; Lee, Sang-Youn "Face recognition using the second-order mixture-of-eigenfaces method" Pattern Recognition, v 37, n 2, February, 2004, p 337-349, Compendex.

[5] ICGST International Journal on Graphics, Vision and Image Processing (GVIP)-http://www.icgst.com/GVIP.html

[6] On relation of image compression and image association. (English. English summary) Skarbek, Wadysaw Fund. Inform. 32 (1997), no. 3-4, 359--371, MathSciNet.

[7] Repetition priming for face speech images: speech-reading primes face identification.- http://www.highbeam.com/doc/1G1-20824406.html

[8] Shakunaga, Takeshi; Sakaue, Fumihiko; Shigenari, Kazuma "Robust face recognition by combining projection-based image correction and decomposed eigenface" Proceedings - Sixth IEEE International Conference on Automatic Face and Gesture Recognition, Proceedings - Sixth IEEE International Conference on Automatic Face and Gesture Recognition FGR 2004, 2004, p 241-247, Compendex.

[9] The development of face-identification skills: what lies behind the face module?

[10] http://homepages.inf.ed.ac.uk/rbf/HIPR2/median.htm

