A Comparative Study of LBPH, SIFT and SURF Algorithms for Face Recognition Task

Parneet Kaur[1], Pratham Mittal[2]

[1]B.E. Student, Dept. of Computer Engineering, Thapar Institute of Engineering and Technology, Punjab, India
[2]B.E. Student, Dept. of Computer Engineering, Thapar Institute of Engineering and Technology, Punjab, India

---***---

Abstract - *Face recognition is a task which human beings do daily, without any effort. Large number of available powerful and cheaper computers and embedded systems has opened ways for great interest in image and video processing in several fields, including machine-human interaction, biometric verifications, and surveillance. This gave rise to intense research and development in the field of face recognition. In this paper, we implemented various algorithms of face recognition like LBPH, SIFT and SURF. For face detection, we used Haar Cascade. We did the training on the similar dataset for each method. We have observed some insights, using which we have tried to evaluate which algorithms performs the best. Comparisons of the algorithms have been done along with discussions of their workings. In the end, comparisons are illustrated in the form of tables for easier understanding.*

***Keywords*: LBPH, Haar-cascade, SIFT, SURF, Face Recognition, Face Detection**

1. INTRODUCTION

Face recognition is one of the most challenging sector of computer vision. Research in this area is inspired not only by the challenges it poses but also by various practical implementations where face identification is required. However, there is still a lot of scope of improvements to reach its highest potential. Though the biometric technologies like fingerprint and iris scanners are way more robust and accurate, but they do not show much scope of progress as of face recognition [1]. Facial recognition can be implemented to a great extent in real time videos for larger number of people [2]. In the paper, we have tried to distinguish between the recognition algorithms implementations in OpenCV. The reasons for using OpenCV include:

Pace: MATLAB utilizes a combination of C++ and JAVA. Thus, when MATLAB is used, the code gets turned into JAVA, C++ and then implements the script. Whereas OpenCV is based on C++/C libraries which assists in speedy execution because the machine code is provided to the computer directly. OpenCV lessens the overhead due to the interpretation as well as provides better usage of resources and time which is needed for image processing [3].

Cost: OpenCV is free to utilize as it works on BSD license, whereas MATLAB needs a paid license for working.

Portability: OpenCV gives the coders, the freedom to run it on any platform which supports the C language execution. It can be utilized on machines having Windows, MacOS or Linux [3].

2. METHODOLOGY

The initial step of face recognition is detection of faces. For that, we will go through face detection using Haar Cascade first and then study the recognition algorithms.

2.1 Haar Cascade

Haar Cascade is an object detection approach in machine learning field. Many image inputs are used in training the cascade classifier function. Paul Viola and Michael Jones proposed it in 2001 and is referred to as the most powerful and useful method of object detection [4].

The algorithm is required to be trained with both negative (without face) and positive image (with face) inputs before doing any operations on the dataset [5]. After the training step, feature extraction is performed. Following this, the size of the goal image is chosen, which is generally smaller than the trained ones. It is put on the target image to calculate the average values of the pixels in each segment. Afterwards, if the average threshold values do pass then it is considered to be a match [6]. Many classifiers are utilised to train the input images as a single classifier is not enough and accurate too [7]. Different features like edge, line, centre-surround are shown in the figure 1.

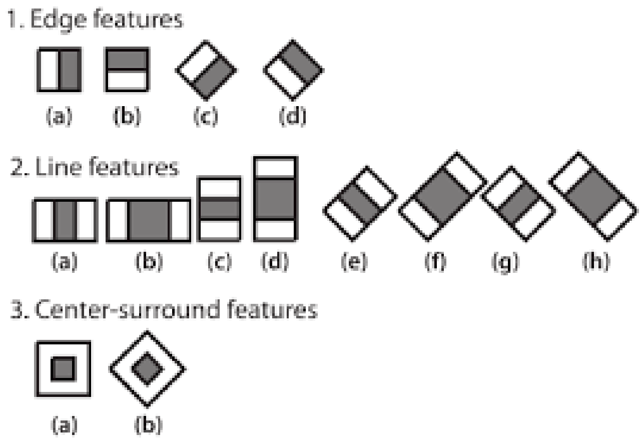
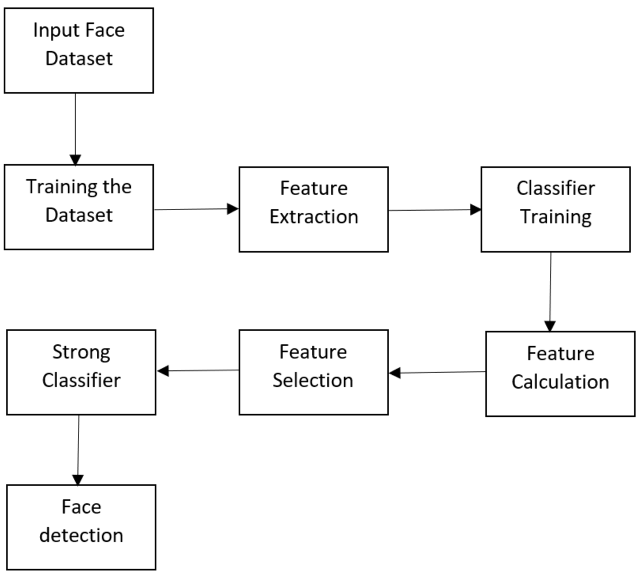


Figure -1: Haar Features

We tried to make the classifier better by utilising many other classifiers. ADA BOOST is an algorithm in machine learning that selects the most suitable match for the gaol image, based on testing on chosen images of many classifiers. It may even revert the process so as to obtain better results [6].



Flowchart -1: Haar Cascade

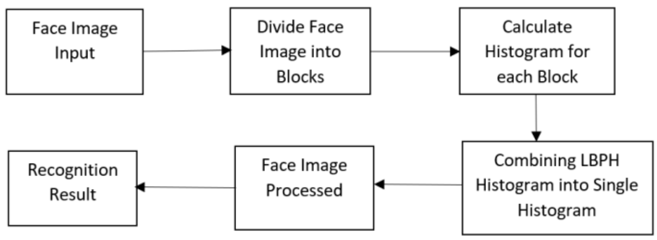
For visual recognition, rectangular combination is utilised. But these rectangles cannot be called true Haar wavelets, so we can call them as the Haar features. We got the existence of Haar features by subtracting the average of light and dark region. Haar feature is there if the computed difference is above the limiting value. The process of Haar Cascade is shown in the flowchart 1.

Jones and Viola utilised a technique known as integral image. Its purpose is to verify if thousands of Haar features are present or absent in the image. After the training step, based on integral image we can detect if there is a face or not in the goal image.

2.2 Linear Binary Pattern Histogram

LBP is an effective texture operator. It holds the threshold value of every pixel in the neighbourhood with the centre pixel value. It considers the output in the form of a binary number [7]. LBP is a popular approach in different applications owing to its simplicity and discriminative power. LBP was described first in 1994 and since then it became a powerful and efficient algorithm in the texture classification [6]. Lately, it was observed that when LBP is combined with the histogram of oriented gradient descriptor, its performance improves on the similar dataset. LBP has comparatively more features such as computational simplicity and monotonic grey-scale changes which makes it possible for real time application by analysing images.

There are many steps required for LBPH algorithm. The sequence of LBPH is demonstrated in flowchart 2. There are four parameters which includes neighbours, radius, Grid Y and Grid X [8]. For finding out the centre of the image, radius is utilised. For more calculations, we will say it a local binary pattern which is generally set to 1. The neighbours are several sample points which are used in building a circular binary pattern and they do not comprise more than 8 pixels as it may need more computational cost. Grid X is basically the number of cells in the horizontal side and Grid Y is the number of cells in the vertical side. Both the grids are set to 8, considering the computational cost. Next, the dataset is to be trained using the algorithm. Every user posses a unique identity and the images are saved by the unique identity value [9].



Flowchart -2: LBPH

Following this is the computational step which comprises the creation of an intermediate image that describes the real image in a better way by utilising the facial characteristics. The LBPH utilises a sliding window which is based on the parameters like neighbours and radius. Consider a facial image in the grayscale. We can have a part of the image in the 3x3 matrix form. Its intensity ranges from 0 to 255-per pixel. The central value of the matrix is the threshold, which is utilised to define new values from 8 neighbours [7]. For the neighbours, the new binary value is provided by comparing the threshold values. If the neighbour's value is more than then threshold one, it is set to 1, else 0. Now the matrix will comprise only the 0 and 1. Then all the binary values are concatenated into one (for example 100001101) as given in the figure 2. After that, the binary value is converted into the decimal and set to the central value of the real image. We obtain a new image that represents improved characteristics of the original [11].

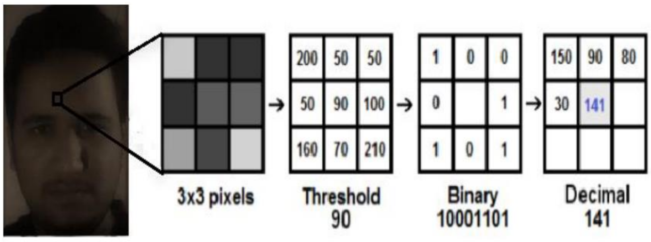


Figure -2: Greyscale value calculation for submatrix

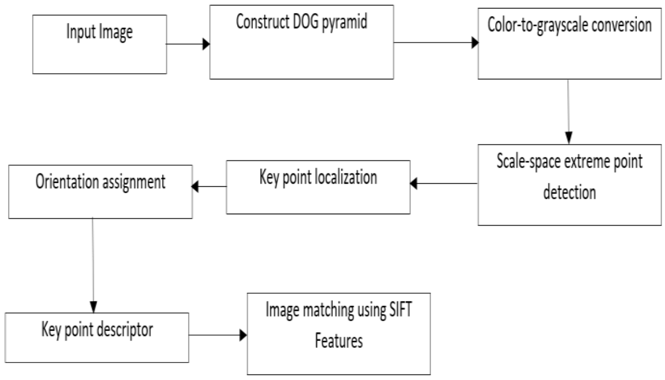
Further, we divide the new image into multiple grids using Grid Y and Grid X parameters. As the image is greyscale, every histogram (from every grid) will have 256 positions, each of which is representing pixel intensity occurrences. A bigger and new histogram is created by concatenating every histogram. Like if there is an 8x8 grid, the final one will posses 8x8x256 = 16,384 positions. The last step is to do face recognition, the algorithm is trained already here. From the trained data, every image is represented with a histogram. For every new image, all the previous steps are implemented to create a corresponding histogram, representing it. All the histograms are compared and the one that have the nearest values are considered as a match. The image represented with the matched histogram is said to be the result. For comparing histograms (calculating distances between two histograms) there are various ways like chi-square, absolute value, Euclidean distance etc. Here, we used Euclidean distance which is based on the following formula number (1)

$$D = \sqrt{\sum_{i=1}^{n} (Hist1_i - Hist2_i)^2} \qquad (1)$$

The resultant of the algorithm is the image with the closest matched histogram. The algorithms also give the confidence value which tells the distance calculated [9]. The lesser the confidence value, the better the result is.
Moreover, setting a threshold value in the algorithm helps to find out if it has accurately recognized an image or not.

2.3 SIFT

(SIFT) Scale-Invariant Feature Transform features extract features from images thus helping in genuine matching between different orientations of the same face or object [10]. The features that are extracted from images are uniform to orientation, scale, and are extremely unique to the image. As described in [11], these features are extracted in four steps in this algorithm. The flowchart 3 is presenting the process of SIFT that are done. Let us see them one-by-one.



Flowchart -3: SIFT

Scale-space Extrema Detection: Scale-space filtering is used to detect larger corners. SIFT algorithm uses Difference of Gaussians (DoG) measure which is an estimation of Laplacian of Gaussian (LoG) used for detecting edges that appear at different image scales. In this step the locations of potential interest points are determined in the image by searching the local maxima and minima over scale and space. For example, one pixel in image is analyzed by comparing with its 8 near-by pixels as well as with 9 in next and 9 in previous scales. If it is found to be a local extremum, it is a potential key point. We can say that this key point is best represented in that scale.

Key point Localization: Once the locations of potential key points are found, they are sifted to generate more accurate results. Taylor series expansion of scale space is used to generate accurate location of extrema. Points of low contrast are discarded if the intensity at this extremum is less than a threshold value. Edges need to be removed as DoG has higher response for edges. For edges, one eigen value is larger than the other. So, a simple method is used i.e., key point is discarded if eigen value ratio is greater than a threshold.

So, what remains is strong interest points after discarding low-contrast key points and edge key points.

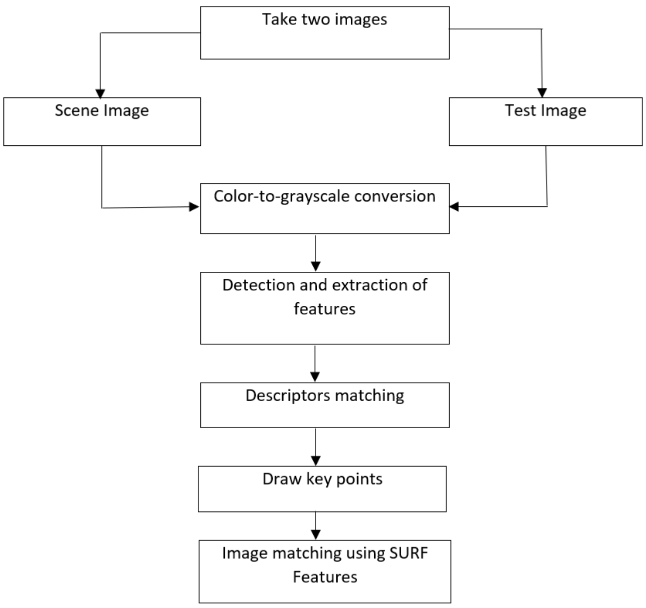
Orientation Assignment: After that, an orientation is assigned to each key point to provide orientation invariance. The magnitude of gradient and direction are calculated by considering a neighbor around that key point location depending upon the scale. A histogram with covering 360 degrees is created called orientation histogram. The highest peak in this histogram is considered and any peak above 80% is considered for calculation of orientation. Key points with same location and scale are created from this histogram thus contributing to stability of matching.

Key point Descriptor and Matching: Now it is time to create key point descriptor. A neighborhood of dimension 16x16 is considered around the key point which is further divided into 16 sub-blocks of size 4x4. Orientation histogram of 8 bins is created for each sub-block. So, these 128 bin values are represented as a vector forming key point descriptor. Key points of two images are compared by recognizing their nearest neighbors. There can be cases due to noise such that second closest match may be close to the first. In this situation, ratio of first closest-distance to second-closest distance is calculated. Key points are rejected if ratio is found to be greater than 0.

2.4 SURF

SURF (Speeded Up Robust Features) [12] proposed by computer vision laboratory of Zurich in 2006 is a fast and powerful algorithm for description of local invariant and comparison of different images. The main purpose of the SURF approach lies in its rapid computation thus allowing real-time applications such as tracking, object detection and recognition.

The flowchart 4 is illustrating the steps of SURF algorithm. Features are extracted in mainly three steps in this algorithm [13][14][15].



Flowchart -4: SURF

Feature Detection: Feature detection means detection of points which lie in corner, spot and T-junction. The most valuable performance of this step is repeatability and reproducibility which shows that the detector can find the points of interest in different viewpoints. SIFT [16] generate Gaussian pyramid firstly by making use of Gaussian and integral image and then make use of the Difference of Gaussians (DOG) and integral image to generate Gaussian Pyramid and construct scale space sharply. The cost time of calculating is too long for interest point detection in SIFT, so in order to fasten it, SURF obtains the maxima by calculating the Hessian matrix.

Feature Description: Feature descriptor in SURF describes the distribution density of its fields. As discussed above SIFT divides 16x16 region into 4*4sub-regions creating orientation histogram of 8 bins for each sub-block and then obtains 4*4*8=128 bin values or descriptor (128-D vector).

SURF unlike SIFT extracts the descriptor by finding Haar wavelet response. The first step is assigning orientation. The estimation of dominant orientation is done by calculating the sum of vertical and horizontal wavelet responses in the scanning area of size $\pi/3$. A local orientation vector is yielded by summing up of these two responses. The longest such vector over all scanning area defines the orientation of the interest point.

The second step is descriptor extraction. For the extraction, the first step includes construction of a square region centered around the key point along the orientation selected in previous section. This region is further split up into smaller 4*4 square sub-regions of length 64 (64-D vector) but in case of SIFT, our descriptor is the 128-D vector, thus showing that SURF is faster and robust than SIFT). SURF descriptor is lighting invariant due to property of invariance to contrast of Haar response.

Feature Matching: Feature matching in SURF is based on the distance between vectors and calculation of the distance of interest points among the images (for example Manhattan distance, Euclidean distance and so on) to check matching of the points.

3. RESULTS

After implementation we prepared the following tables. These tables provide the accuracy and time efficiency after testing for the three algorithms. We compared the number of images rightly recognized with the total number of images tested as shown in figure 3 and 4. SURF and SIFT are applied on original image with different types of images and we found that features points matching is better in case of SURF than SIFT and in case of recognizing. Table 1 and 2 show that LBH leads the race, both in terms of time efficiency and accuracy with SURF at second and then SIFT at the last.

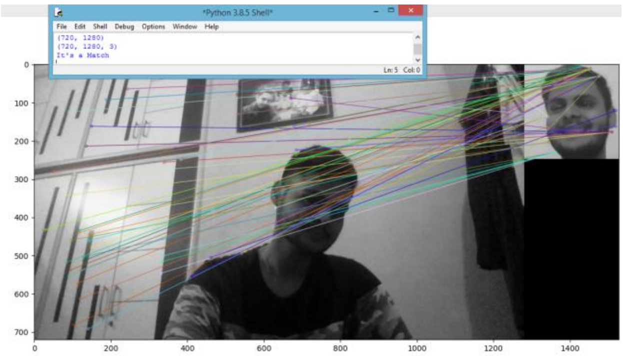


Figure -3: Face recognition using SIFT

Figure -4: Face recognition using LBPH

TABLE -1: Time Efficiency

Methods	LBPH	SIFT	SURF
Time (in sec)	0.06	0.89	0.51

TABLE -2: Accuracy

	4 subjects		
Methods	Total 40 training images (10 pics per subject)		
	correct	error	Result
LBPH	40	0	100%
SIFT	34	6	85%
SURF	38	2	95%

4. CONCLUSIONS

In this paper a comparison of performances between LBPH, SIFT and SURF algorithm in the term of matching rate of detected points have been observed. We have concluded that when we use LBPH with Haar cascade we get the best result from all three algorithms. If we do not use them together, they will not produce the result with similar accuracy. SIFT and SURF, both are rapid and powerful methods for feature detection and matching and are invariant in nature. SURF gives better result in feature matching and consumes less time than SIFT [17]. Both have same effect in detecting feature points in good lightening conditions.

5. REFERENCES

[1] Roja Ghasemi, and Maryam Ahmady, "Facial Expression Recognition Using Facial Effective Areas And Fuzzy Logic," 2014 Iranian Conference on Intelligent Systems (ICIS), IEEE, April 2014.

[2] Shervin EMAMI, and Valentin Petruț SUCIU, "Facial Recognition using OpenCV," Journal of Mobile, Embedded and Distributed Systems, vol. IV, no. 1, 2012.

[3] Kruti Goyal, Kartikey Agarwal, and Rishi Kumar, "Face Detection and Tracking Using OpenCV," International Conference on Electronics, Communication and Aerospace Technology ICECA 2017.

[4] Jalled, Fares, and Ilia Voronkov. "Object detection using image processing." *arXiv preprint arXiv:1611.07791* (2016).

[5] Shalince Dominic, Mahesh Mohan, Aparna C, Ajeesh M S, Aswin S Nath, and Anil Antony, "A REVIEW OF FACE DETECTION SYSTEM," International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT) – 2016, IEEE, November 2016.

[6] Lahiru Dinalankara, "Face Detection & Face Recognition Using Open Computer Vision Classifies," ResearchGate, August 2017.

[7] Kushsairy Kadir, Mohd Khairi Kamaruddin, Haidawati Nasir, Zulkifli Abdul Kadir Bakti, and Sairul Safie, "A Comparative Study between LBP and Haar-like features for Face Detection Using OpenCV," 4th International Conference on Engineering Technology and Technopreneuship (ICE2T), IEEE, January 2015.

[8] Jia-Jing Lin, and Shih-Chang Huang, "The Implementation of the Visitor Access Control System for the Senior Citizen Based on the LBP Face Recognition," 2017 International Conference on Fuzzy Theory and Its Applications (iFUZZY), IEEE, March 2018.

[9] Nasim, A. S. M., Md Hasib, and Nazmul Hasan Nahid. "Face Recognition Based on LBPH Algorithm." (2019).

[10] David G. Lowe. Distinctive image features from scale-invariant key points. Inter-national journal of computer vision, 60, 2004.

[11] D.G. Lowe, “Distinctive image features from scale-invariant key-points,”International Journal of Computer Vision, vol. 60, pp. 91–110,2004.

[12] H Bay, A Ess, T Tuytelaars, et al. SURF: Speeded-up-robust features. [J]. Computer Vision and Image Understanding, 2008, 110: 346-359.

[13] Qiao Yongjun, Xie Xiaofang, Li Dedong, Sun Tao. Research on the speed-up method of SURF feature matching by blocking. LASER & INFRARED, 2011, 41(6): 691-696.

[14] Zhang Huijuan, Hu Qiong. Fast Image Matching Based-on Improved SURF Algorithm.[C] Electronics, Communications and Control International Conference, 2011, 1460-1463.

[15] Shi Lei, Xie Xiaojun, Qiao Yongjun. Research on a Face Tracking Technology Based on SURF Algorithm.[J] Computer Simulation, 2010, 27(12): 227-231.

[16] Zhou Zhiming, Yu Songyu, Zhang Rui, Yang Xiaokang. An algorithm for Face Recognition Based on SIFT Descripor.[J] Journal of Image and Graphics, 2008, 13(10): 1882-1885.

[17] F. Qi, X. Weihong, and L. Qiang, “Research of Image Matching Based on Improved SURF Algorithm,” TELKOMNIKA Indones. J. Electr. Eng., vol. 12, no. 2, pp. 1395–1402, 2014