Grenze International Journal of Engineering and Technology, Jan Issue

A Resultant of Local Binary Pattern Histogram (LBPH) and Speeded Up Robust Feature (SURF) Algorithm for Face Recognition

Durgesh Pandey[1], Nitin Kushwaha[2] and Kamal Srivastava[3]
[1-3]Department of Computer Engineering, Ambalika Institute of Management and Technology, Lucknow, INDIA
durgeshpandey733@gmail.com, strangenk66@gmail.com , 2007.srivastava@gmail.com

***Abstract*—In the time of increasing crime Face recognition is very significant with regards to computer vision, security, monitoring, reconnaissance, pattern checking, neural network, real-time video processing, etc. Face is non- interfering solid Biometrics for distinguishing people and subsequently goons consistently attempt to cover their faces by various fake methods, for example, plastic surgery, mask, etc. The accessibility of an exhaustive face database is vital to test the working of these algorithms of face recognition. Though, we-tested freely accessible face databases as well as self-created one containing face pictures with a wide assortment of postures, light, signals and face impediments. The contribution of this research paper is Comparing Local Binary Pattern Histogram (LBPH) and Speeded Up Robust Feature (SURF) algorithms and finding the best among them.**

***Index Terms*— Introduction, LBPH, SURF, Experimental results, Conclusion & References.**

I. INTRODUCTION

At present, in computer vision the most significant subject is the Face recognition which has a numerous applications, for example, for monitoring, security, forensics, banking, IOT etc.
The efficiency of the most of the current systems are adversely affected at only two points:

1. Huge variation in facial appearance of a person because of changes in posture, age and expression, just as encompassing light condition and camera parameters, and
2. High dimensionality due to small size sample training.

Throughout the years, numerous researchers have created various sorts of face recognition algorithms, including (LBP) Local Binary Pattern Algorithm, (HOG) Histograms of Oriented Gradients Algorithm, (SC) Sparse Coding algorithm, Gabor Feature Algorithm and (LDA) Linear Discriminant Analysis Algorithm with all having accuracy rate b/w 50% to 75%. whereas Local Binary Pattern Histogram Algorithm recognizes front as well as side faces having an accuracy rate of 85% to 90%.
Coming to our next comparison subject Speeded Up Robust Features(SURF), this algorithm consists of mainly three procedures. Initially, T-junctions, corners, and blobs which are also known as ‘interest points’ are marked at different points on the target image. Feature Vectors are used to represent the neighborhood of the each interest point. The descriptor has the following properties: Geometric, Robust to noise, Distinctive, Photo-metric deformations and Error detection. At the end the matching b/w the Descriptor vectors of various images are performed by finding the distance b/w the vectors using Euclidean Distance or the Mahalanobis formula.

Grenze ID: 01.GIJET.7.1.507

II. LOCAL BINARY PATTERN HISTOGRAM ALGORITHM

The first powerful feature for texture classification came in 1994 as LBP. On further research it was found that combining (HOG) Histograms of Oriented Gradients Descriptor with LBP effectively increases the efficiency of detection on various data-sets. The images can be simply represented by data vectors by combining Local Binary Pattern with Histograms. The steps performed in this algorithm are as follows:

A. Parameters: Four parameters are mainly used in LBPH

Grid X: Denotes the no. of Cells present on the horizontal axis.
Grid Y: Denotes the no. of Cells present on the vertical axis.
As the no. of cells increases, grids becomes finer & there occurs an increase in the dimensionality of the Feature vectors. Generally, both are set to 8.
Neighbors: They denote the no. of Sample points which are used for the generation of circular LBP. As we increase the sample points the computational cost also increases. Generally, it is also set to 8.
Radius: It is utilized for the generation of circular LBP by taking the central pixel as a centre. Generally, it is set to 1.

B. Algorithm Training

Initially, the data-set of the target person's facial images is required for the training of the algorithm. Then mark ID (which can be no. or name of that person) for each image, so that the algorithm can recognize the input image and provide the output. A person must have same ID for all their images.

C. Implement the LBP operation

Now on emphasizing the facial features we can generate an Intermediate image that illustrate the actual image of the person in a refined manner. For that the Sliding window idea is employed by the Algorithm which uses neighbors and radius as a parameter.

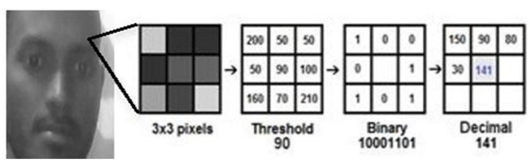

Figure 1.

Based on the image above, let us divide the whole process into small parts for proper understanding:

1. Let the image is in grayscale. Here the image can be considered as a 3*3 matrix window where each pixel has an intensity b/w (0-255).
2. Threshold denotes the central-value of that matrix. It is further utilized to update the neighbors values.
3. For every threshold's neighbour a "new binary value" is assigned. Set 0 for values lower than the threshold and 1 for the rest.
4. Binary values are assigned to the Matrix, except for the central value. Now a 8 bit binary value (i.e. 01001101) is formed by the line by line concatenation of all the binary values of the Matrix.
5. Now, the Binary value is converted into to a Decimal value which is further assigned to the central value of that Matrix. It represents a pixel of actual image.
6. Now the actual image has been transformed into a new unique image which shows upgraded features.
7. **Note**: The Local binary pattern method is extended using **bi-linear interpolation** for variation in the size of neighbors and radius, which is also known as circular Local binary pattern.

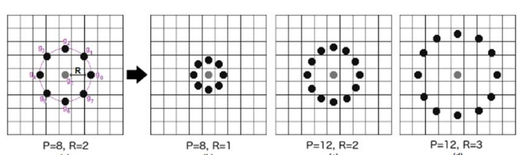

Figure 2.

The values of four closest Pixels is taken to determine the newer data spot when any data spot lies b/w the Pixels.

D. Histogram Extraction

1. The Parameters Grid X & Grid Y are used to split that image among various grids.
2. When the histogram is created from the Greyscale image, it contains 256 locations representing the intensities of every pixels.
3. Unique histogram is generated by concatenation of every other histograms. If its size is 8*8 grids, then there will be 8*8*256 = 16384 locations in that unique Histogram. The final histogram represents the features of the original image.

E. Face recognition

Training of the algorithm is performed. Every image of the training data-set is represented by a unique Histogram. The same steps are performed for the input image to obtain its histogram.

1. So just compare the histograms of the image data-set with the histogram of input image and it returns output image which has the highest resemblance among the histograms.

2. There are various techniques available for the comparison of these histograms, e.g. **Chi-square, Absolute Value, Euclidean Distance**, etc. For example, the formula for the Euclidean Distance is as follows:

$$D = \sqrt{\sum_{i=1}^{n}(hist1_i - hist2_i)^2}$$

3. The identity of the image which has the highest resemblance among the histograms is returned as an output. The confidence measure can also be assessed with the help of distance calculated returned by the algorithm. To assess whether image is recognized accurately or not we can use the values of confidence and threshold. The accuracy can also be calculated using these results.

F. For Accuracy

Let,
a = cv2.face.LBPHFaceRecognizer_create()
//a is a variable
result=a.predict(grey)
//grey is a variable that contains grey scale image.
Accuracy = (100*(1-a.result[1])/300)

III. SPEEDED UP ROBUST FEATURE ALGORITHM

SURF is one of the famous algorithm for image processing system ,though it is patented so, is not available in openCV new versions, for using it one have to build the openCV and extend the features.

- Scale-invariant feature transform (SIFT) is partially included in the SURF.
- Interest points are identified by integer approx. of determinant of Hessian Blob detector that is calculated by integer operations.
- The sum of the Haar wavelet constitutes the Feature Descriptor which are present near the locus of interest.

Some important features of SURF :

A. Detection

For approximation of Gaussian smoothing SURF uses square-shaped filters. The application of integral image robust the process of filtration of images in squared shape:

$$S(x,y) = \sum_{i=0}^{x}\sum_{j=0}^{y} I(i,j)$$

The evaluation of addition of real images inside the rectangular shape can be rapidly done with the application of integral images as it requires to evaluate at the corners of the rectangle only. Hessian Matrix has a Blob

Detector which is used by the SURF for searching all the point of interests. The determinants of Hessian Matrix are used as a measure for local change around the points and only those points are chosen which have the highest determinants.

For the selection of scale the Hessian's Determinant is used in complement to the Schmid's & Mikolajczyk's Hessian-laplacian detector. We can find the Hessian Matrix H(p, σ) on some points like p & σ as a scale where p=(x, y) inside the image I:

$$H(p,\sigma) = \begin{pmatrix} L_{xx}(p,\sigma) & L_{xy}(p,\sigma) \\ L_{yx}(p,\sigma) & L_{yy}(p,\sigma) \end{pmatrix}$$

where $L_{xx}(p,\sigma)$ etc. denotes the Convolution of the Second-Order Derivative of the Gaussian on that point p.

For example, in this image matching key points are represented by different colourful lines. The matching ratio is calculated as follows:

$\mathbf{M_R = M_C / M_T * 100}$

where M_C denotes no. of correct matching pairs & M_T denotes the total no. of matching pairs inside the image.

B. Descriptor

For every point of interest a description is identified previously because mostly the descriptors are calculated locally. Fast and eccentric descriptions must be provided by the descriptors as it is the main objective of an image feature.

The descriptor's dimensionality has a huge impact over the Accuracy, Point matching and the Computational Complexity. Short Descriptors can be fast but provides more false positives and insufficient result in case of variations in appearance.

Initially the orientations that can be reproduced are fixed by collecting info from the areas about Interest points. Then create a squared area aligning the marked orientations and the descriptor of SURF is extracted from it.

C. Matching

The descriptors extracted out of various images can be compared to find the matching pairs. There are different key-points or interest points matching techniques such as Brute Force Matcher and FLANN Matcher. In our Experiment we have used Brute Force Matcher.

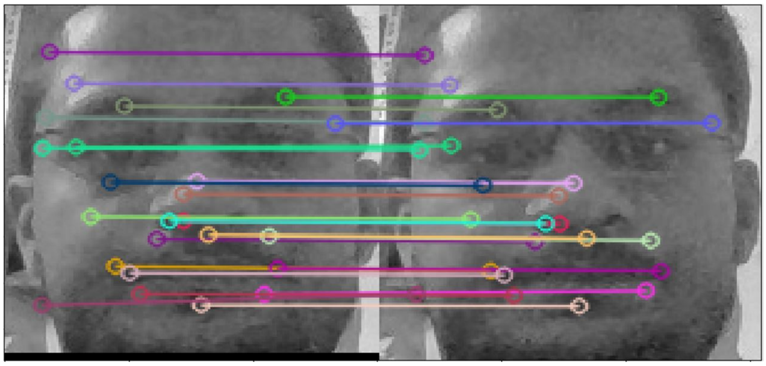

Fig. 4. Representation of keypoints matching

IV. EXPERIMENTAL RESULTS

The performance of these algorithms have been evaluated by performing the experiments, and compared the results with each other. The data sets we used in the experiments are standard Yale Face Database as well as self-created database of our own images having different subsets of Facial Expression.

In the given table the matching ratio or matching percentage or recognition rates of LBPH and SURF are compared.

We have tested it for 10 subjects (people) in which each subject has a subset of 10 images with different expressions.

TABLE I: COMPARISON OF MATCHING RATIO

Subjects	LBPH(%)	SURF(%)
Subj_1	84	77
Subj_2	76	80
Subj_3	77	74
Subj_4	71	68
Subj_5	67	76
Subj_6	74	79
Subj_7	77	81
Subj_8	75	72
Subj_9	77	75
Subj_10	77	77

Average matching ratio for LBPH : 84 %

TABLE II: COMPARISON OF EXECUTION TIME

Subjects	LBPH(sec)	SURF(sec)
Subj_1	0.5368	0.4155
Subj_2	0.4932	0.4213
Subj_3	0.5012	0.4532
Subj_4	0.4889	0.3988
Subj_5	0.5456	0.4011
Subj_6	0.4798	0.4111
Subj_7	0.5234	0.4312
Subj_8	0.5053	0.4456
Subj_9	0.4894	0.4576
Subj_10	0.5131	0.3911

Average matching ratio for SURF : 86 %
Average Execution time for LBPH: 0.5076 seconds.
Average Execution time for SURF: 0.4226 seconds.
**The results may slightly vary on systems with different configurations of hardware* .

V. CONCLUSIONS

After, doing above experiment with respect to Accuracy and Time between LBPH and SURF, both the algorithms are performing good at their place. Both have approximately same accuracy but in execution time, SURF is comparatively faster than LBPH.

Since each one of the algorithms has its own strengths and weaknesses, there isn’t a definite winner, but rather a more suitable choice would be SURF. So, we can say that both algorithms work at satisfactory level in order to recognize faces but the better choice would be SURF. It is expected that combining two or more face recognition algorithms may yield higher accuracy at a higher computational cost.

REFERENCES

[1] Shakhnarovich G, Moghaddam B. Face recognition in subspaces. Handbook of Face Recognition, 2004.

[2] Practical Machine Learning and Image Processing: For Facial Recognition, Object Detection, and Pattern Recognition Using Python. Book by Himanshu Singh.

[3] D. G. Lowe, "Object recognition from local scale-invariant features," International Conference on Computer Vision, Corfu, Greece, 1150–1157 (1999).

[4] Computer Vision: Algorithms and Applications by Richard Szeliski.

[5] Hongshuai Zhang, ZhiyiQuLiping, Yuan GangLi, A Face Recognition Method Based on LBP Feature for CNN, 2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC).

[6] Richard Szeliski, "Computer Vision: Algorithms and Applications" published by Springer.

[7] Introduction to Machine Learning with Python: A Guide for Data Scientists. Book by Andreas C. Müller and Sarah Guido.
[8] Mastering OpenCV 4 with Python: A Practical Guide Covering Topics from Image Processing, Augmented Reality to Deep Learning with OpenCV 4 and Python 3.7.