Evaluation and Comparison of LBP and HAAR Algorithms for Face Recognition

Neha P[*1] **Mohammed Faizan Bari**[*2] **Vindhya P Malagi**[*3]
[*1,2,3]Assistant Professor, Dept of CSE
[*1,2,3]Dayananda Sagar College of Engineering Bangalore, India

***Abstract*— Face detection for various purposes like fraud detection in passports, voting, audience attendance marking system etc face many challenges. In this paper we throw light on the face detection algorithms of LBP and HAAR with respect to face recognition.Based on the surveyed techniques, the proposed work evaluates the most promising algorithm and its associated cascades. Accordingly, here the LBP and HAAR algorithms along with their cascades have been explored in detail and the best strategy to go about this situation is identified.**

Keywords*— Face detection, face recognition, cascade classifiers, features, visual descriptor.

I. INTRODUCTION

Face recognition is a technology the world needs in various fields. The face recognition has been able to solve various pressing issues like - fraudulent passports, identification of criminals, prevention of fraudulent voting, banks and many more such applications.

Face recognition can be termed as a process which recognizes who a person is from the database it maintains.

There are various challenges faced in this field when it comes to real world operations.

The real world scenarios have various challenging aspects like illumination, pose variations, occlusions and expression changes etc.[1]

This paper focuses on the LBP and HAAR face recognition algorithms and their corresponding LBP and HAAR cascades. The performance of these algorithms are compared and also a combination of these two algorithms is also evaluated and reviewed. There is a comparison drawn between the various cascades and there is a suggestion on the appropriate cascade to be used.

The more the training sample, better are the results. Also better conclusions can be can be drawn from such rigorous experiments..

Earlier, various systems like attendance, fraudulent voter detection were handled manually. With face recognition, the emphasis on automating this manual process and bringing about better systems which help build strong applications and invariably make the world a better place.

Features:

Features are in important aspect of the functioning of the algorithm.LBP and HAAR have very diverse approaches in the way the features are extracted.

A. HAAR Features:

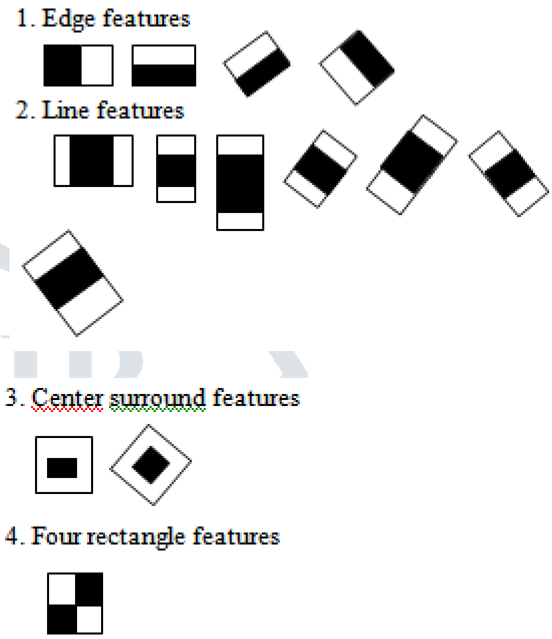

Figure 1: Haar feature representation [2][10]

The images in figure 1 include the region of interest as well as unnecessary portion which needs to be discarded.[2]

The images have both face region, which is the area of interest and a non-face region which needs to be discarded.

The images shown above are various features such as edge features, line features, center-surround & four rectangle features as shown by figure 1,2,3,4. These concern the various portions of the region of interest.

As seen in the above figure, the feature window has a black portion and a white portion.

Each window is placed on the image and the algorithm subtracts the white portion from the black portion to get the feature value.[9]

B. LBP Features

LBP is a visual/texture descrizptor . Descriptors find the connections between pixels contained in a digital image (i.e, binary value of a 2D image) and what human beings recall after seeing an image after some time.

There are many descriptors but the descriptors used for the purpose of this project are: color descriptors, texture descriptors and shape descriptors.

Color descriptors: This is the most basic descriptor. There are various tools in color descriptors. Some tools focus on color distribution and some tools focus on color relation between groups of images.

Texture descriptors: This is an important descriptor used to describe the image. These descriptors characterize a region by observing the region homogeneity and histograms of region borders.

Shape descriptors: Contains information similar to the one which helps humans recognize objects through their shape. This information is extracted through segmentation. This process is quite similar to the one followed in human visual system. Segmentation refers to partitioning of the image into multiple segments (set of pixels) in order to change the representation of the image into something more meaningful that is easier to analyze. But nowadays instead of segmentation there are algorithms with a good amount of approximation which meet the purpose.

The visual features present in images are described by different visual descriptors which may specify the shape, color, texture etc. In this way each visual feature has a unique visual descriptor.

In LBP there are no predefined features like line feature, edge feature, and center-surround feature. Each training image is divided into blocks as shown in the picture below in figure 2.

Figure 2:Division of training images into blocks [2][7]

Each block is a 3x3 window consisting of a total 9 pixels, which are actually 8 pixels compared with respect to the center pixel.[6]

II. ALGORITHMIC DIFFERENCE

This section throws light on the algorithmic difference between LBP and HAAR.

A. *HAAR*

Initially a training set, consisting of positive image (images of faces) and negative images (images without faces),is created.[2]

The HAAR features are then extracted from the image by using the line feature, edge feature, center-surround feature with help of respective windows.

Each window is placed on the picture to calculate a single feature.This feature is a value obtained by subtracting the white part of the feature from the black part of the feature (image)[2][9]

Next, all possible sizes of each window are placed on all possible locations of each image to calculate variety of features of the image.

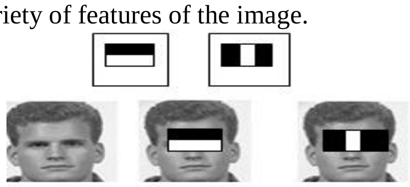

Figure 3: Window positioning representation [2]

Different types of windows are used for feature detection, since it is not efficient to use only one particular feature window at all locations.

For example, the feature window of eye focuses on darker region as the eye region is usually dark and the nose feature focuses on brighter features as the nose region is brighter as shown in figure 2.

The nose window is differentiated from an eye window since a nose window focuses on bright features and an eye window focuses on darker regions.

Therefore using an eye window for detecting nose, cheeks etc are not suitable.

For improving the accuracy we focus on the relevant features only and this technique is called Adaboost [4][8]

Adaboost technique - All the features are applied on the training data and a threshold value and an error rate are calculated for each feature. The features with low error rates are accepted, as they classify a face from a non-face.[4][8]

Another feature of the algorithm is to identify a non face image and discard it immediately, without the need to process it, since in an image most of the region is a non-face image.

From the training set, only relevant features are chosen from a large number of features. The remaining features are discarded as they are irrelevant. Rather than applying all the obtained features on the test set, the features are classified into various stages of classifiers, called cascaded classifiers.[9][10] These are then applied onto the feature window stage by stage, and if at any stage the window fails, it is discarded. The window which passes all the stages, is the face region.

The algorithm uses a variant of adaboost technique to select the best features and to train the classifiers which use them. The algorithm constructs a strong classifier by combining the various weak classifiers [4] as shown in equation 1.

$$h(x)=\mathrm{sgn}\left(\sum_{j=1}^{M} a_j h_j(x)\right) \quad (1)$$

The weak classifiers are threshold functions based on the feature f_j

$$h_j(x)=\begin{cases} -s_j & \text{if } f_j < \Theta_j \\ s_j & \text{Otherwise} \end{cases} \quad (2)$$

The threshold value $-\square_j$ and the polarity $s_j \in \pm 1$ and α_j are determined in training.

Input: Set of N positive and negative training images with their labels (x^i, y^i). If image i is a face $y^i=1$ else $y^i=-1$

1)Initialization: Assign the weight $w_1^i=1/N$ to every image i.

2)For each feature f_j with $j=1,\ldots,M$

i) The weights should be normalized such that they sum up to one.

ii) The feature is applied to every image in the training set, and the optimal threshold and polarity Θ_j, s_j.This minimizes the weighted classification error.That is, as given in equation (3):

$$\Theta_j, s_j=\arg\min_{\Theta,s} \sum_{i=1}^{N} w_j^i \epsilon_j^i \quad (3)$$

where,

$\epsilon_i^j=\{0$ if $y^i=h_j(x^i,\Theta_j,s_j)$, 1 Otherwise$\}$

iii) A weight α_j is assigned to h_j which is inversely proportional to the error rate so as to consider the best classifiers.

iv)In the next iteration, the weights are reduced for the images that are correctly classified

3)Set the final classifier to equation (4):

$h(x)=\mathrm{sgn}(\sum_{j=1}^{M} \alpha_j h_j(x))$ (4)

B. LBP

LBP (Local Binary Pattern) is a visual/texture descriptor. LBP features are extracted from the image and a feature vector is created which classifies a face from a non-face.

Each training image is divided into blocks. Each block consists of a 3x3 window (i.e. 9 pixels). The surrounding 8 pixels are compared with the center pixel. If the value of the surrounding pixel is greater than or equal to the value of the centre pixel, its value is set to 1 otherwise it is set to 0. Then the algorithm reads the updated pixel values in a clockwise manner and a binary number is generated. The binary value is then converted to its decimal equivalent. This decimal number is the value of the center pixel. This process is repeated for every pixel in the block.

LBP code generation:[3][10]

LBP code= $\sum_{n} \mathrm{step\_fun}(I_n - I_{thresh}) * 2^n$,

$\mathrm{step\_fun}(x)=\begin{cases} 1 & \text{if } x\geq 0, \\ 0 & \text{if } x<0 \end{cases}$

The above equation shows how the lbp code is generated. I_{thresh} is the threshold value, I_n are the intensities of the surrounding window pixels (n=0,1,….7). The threshold result is as shown in the figure below. [3] This result is multiplied with a predefined mask which is usually incremental powers of 2. Finally the values are added to obtain an 8 bit LBP code.

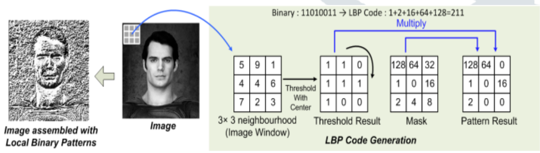

Figure 4: LBP code generation [3][7][8]

The result of LBP processing is shown in the figure below. Divide the LBP based image into k blocks of W_width* W_height pixels (ex: 2×4,4×4,8×8). [3]

Local image descriptors are built by generating local histogram for each block in the image. The local histograms are then concatenated to form a single global histogram as shown in the figure below.

The global histogram expresses information in three different levels:[3]

a) The individual LBP code contains information at the pixel-level

b) The local histograms contain information on a regional level

The concatenated regional histograms contain a global description.

The resulting histogram encodes both local and global characteristics and makes it more robust to object pose and illumination variations.[3]

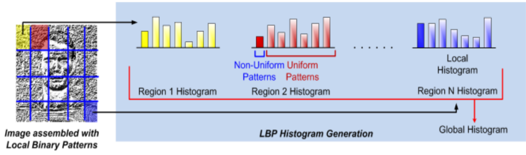

Figure 5: LBP Histogram generation[3]

III. APPLICATION DIFFERENCE

LBP uses integer calculations and avoids floating points calculations and thus has more precision.Therefore LBP is better suited to mobile/embedded systems when compared to haar.

IV. PERFORMANCE DIFFERENCE

The difference in the performances of the two approaches is shown in table 1:

Table 1: Performance difference

HAAR	***LBP***
High detection accuracy	Less accurate
Low false positive rates	High false positive rates
Executes slower when compared to LBP	Execution is faster
Takes longer time to train the data	Takes lesser time to train on images
Under difficult lighting conditions the performance reduces.	Performance is stronger with respect to haar under difficult lighting conditions.
Computation is complex and slow because it deals with floating point numbers	Computation is simple and fast because it deals only with integers and avoids floating point.

V. COMBINED PERFORMANCE

The project conducted for this paper, included a combination of haar and lbp. By combining the advantages of haar and lbp i.e accuracy and speed respectively an improved performance of results was obtained. For face detection haar cascade was used to train the data and to recognize the face, while to recognize the face, lbph face recognizer was used with lbp classifier.

VI. HAAR AND LBP LITERATURE REFERENCE

Both haar and lbp provide various cascade classifiers which provide fairly good results while the eye cascade classifier was used, which gave a high level of accuracy and low false positive rates. As supported by the literature as well as by the experimental results, it was found that a high degree of positive results with high accuracy(80%) and low false positive rates was obtained, by using eye cascade classifiers.

VII. CHALLENGES FACED AND RESULTS

Extensive literature survey to understand how to implement face recognition project using opencv framework by using HAAR and the other is by using LBP. LBP has faster execution rates when compared to HAAR.On the other hand HAAR has a higher accuracy rate when compared to LBP algorithm.

Balancing accuracy vs speed as a trade-off was the challenge.Combining both methods(HAAR and LBP) to obtain higher accuracy for face recognition in a shorter time.

HAAR cascade classifiers were used in face detection and LBPH face recognizer which uses LBP classifier was used for face recognition with the intention of exploiting the strong features of both the algorithms(i.e speed and accuracy).

Experimental results showed that the best cascade classifier to use was the haar-eye-cascade classifier, though there were few challenges. The training data had a window in the background behind the image of a person which made it difficult for the eye cascade classifier to recognize the 'eye' which was the objective of the algorithm. The problem was solved by cropping the unnecessary part of the image that is the background. Another possible solution which was tried out was to use a face cascade classifier to recognize the faces and then use the eye cascade classifier to detect the eye. This approach increased the execution time.

VIII. EXPERIMENTAL RESULT COMPARISON

Rigorous experimentation on various datasets was conducted and experimental results analysed. Table 2 shows some of those results.

Table 2: Experimental results

Cascade classifier used	Face Recognizer	Face Detection time (training data)	Accuracy ratio (test data)	Face recognition accuracy (test data)
Haarcascade_eye.xml	LBPH	1 min 37 sec	39/48	81.25%
Haarcascade_frontalface_alt.xml	LBPH	2 min 37 sec	32/48	66.667%
Haarcascade_frontalface_alt2.xml	LBPH	1 min 59 sec	34/48	70.83%
lbpcascade_frontalface.ml	LBPH	1 min 28 sec	26/48	54.16%

From the above experimental observations it is seen that the cascade haarcascade_eye.xml has the best accuracy and also has a fairly faster rate of face detection.

The experimental snapshots (using haarcascade _eye.xml):

Figure 6: snapshot of successful predictions

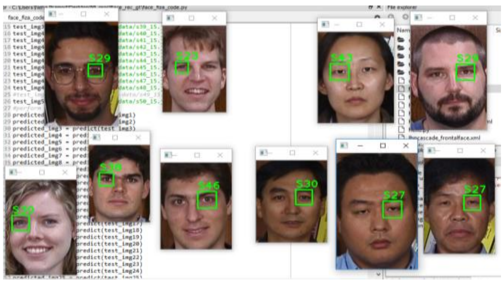

Figure 7: snapshot of unsuccessful predictions

The benchmark dataset used was Georgia Tech Face Database[5] by Georgia Institute of Technology and is primarily used for face recognition.

REFERENCES

[1] M.Hassaballah, Saleh Aly, et al. "Face recognition: challenges, achievements and future directions", 30 July 2015, Vol. 9, pp. 614-626.

[2] Ramiz Raja, articles about "Face detection using opencv and python: A beginner's guide", July 8,2017.

[3] Christos Kyrkou "Object detection using local binary patterns",Nov 1,2017.

[4] Viola Jones "Viola-Jones object detection framework",2001

[5] Georgia Tech Face Database, Georgia Institute of Technology-contains images of 50 people taken in two or three sessions between 06/01/99 and 11/15/99 at the Center for Signal and Image Processing at Georgia Institute of Technology.

[6] Ramiz Raja, articles about "Face recognition using opencv and python: A beginner's guide", August 3,2017.

[7] Vladimir Petruk, Prof. Igor Spiridonov, et al. "Application of Local Binary Patterns to Face Recognition Problem Solving", April 7,2011, pp. 7-9.

[8] Faizan Ahmad , Aaima Najam, et al. "Image-based Face Detection and Recognition: State of the Art" IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 6, No 1, November 2012 ISSN (Online): 1694-0814, pp. 170.

[9] Sander Soo, "Object detection using Haar-cascade Classifier", Institute of Computer Science, University of Tartu, pp. 2-3.

[10] Juliano E. C. Cruza1 , Elcio H. Shiguemorib, et al."A comparison of Haar-like, LBP and HOG approaches to concrete and asphalt runway detection in high resolution imagery", J. Comp. Int. Sci. (2015) 6(3):123-125.