ISSN 2395-1621

Occluded face Detection in ATM system using LBPH Algorithm

Apurva Sathe, Akshay Dhanmane, Pranita Karle, Devashish Bakare
Prof. A. V. Markad

Department of information Technology,

Savitribai Phule Pune University, Pune, India.

ABSTRACT

Now days we seen that the robbery had been increasing in tremendous manner in ATM, so that we are creating a system that identify the thief by using facial recognition technique and algorithm such as HAAR cascade and LBPH. The implementation and execution with results are explained.

Keywords: LBPH, Image Processing, Face detetion.

ARTICLE INFO

Article History

Received: 8th June 2020

Received in revised form :

8th June 2020

Accepted: 12th June 2020

Published online :

12th June 2020

I. INTRODUCTION

A face recognition is nothing but a technique in which we stored a features of a face in mathematically manner to identify a persons, it may be contains biometrically or different techniques and algorithms.

The face is the most crucial entity for human identity. The feature best distinguishes a person. And for the very same reasons, Face Recognition is an important technique. Face recognition is an interesting and challenging problem and impacts important applications in many areas such as identification for law enforcement, authentication for banking and security system access, and personal identification among others. Face recognition is an easy task for humans but it is an entirely different task for a computer. A very little is known about human recognition to date on How do we analyze an image and how does the brain encode it and Are inner features (eyes, nose, mouth) or outer features (head shape, hairline) used for successful face recognition? Neurophysiologist David Hubel and Torsten Wiesel have shown that our brain has specialized nerve cells responding to specific local features of a scene, such as lines, edges, angles or movement. Since we don’t see the world as scattered pieces, our visual cortex must somehow combine the different sources of information into useful patterns. Automatic face recognition is all about extracting those meaningful features from an image, putting them into a useful representation and performing some classifications on them.

In face Recognition there are a lots of studies were happens in past and been happens now days but the main issues in face recognition is that no one gives the accuracy to identifications.

Now there is many technology and algorithms which can solve these problem in our studies we were used a LBPH (local binary pattern histogram) algorithm.

II. LITERATURE SURVEYS

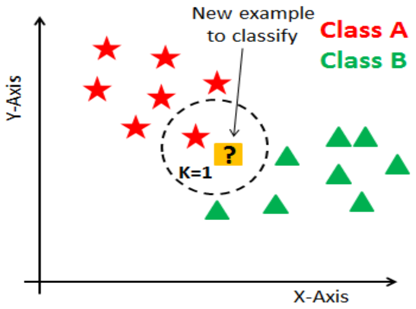
In literature, we were studies two algorithms

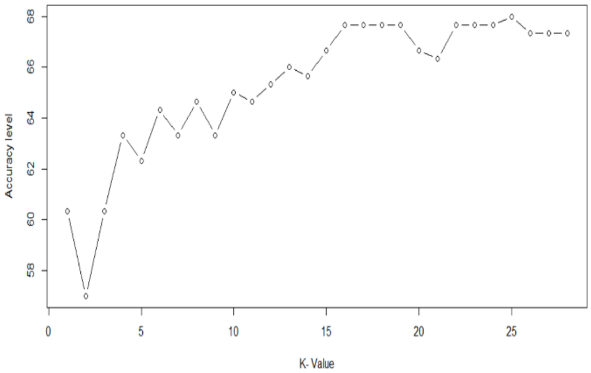
1) KNN algorithm

KNN algorithmic program is among one of the only algorithmic program for regression and classification in supervised learning. KNN is non-parametric which suggests it doesn't create any assumptions however bases on the model structure generated from the data

Through this pepper, a very basic form of face recognition has been implemented using the Haar Cascades Classifier, openCV & K-Nearest Neighbors Algorithm.

It works by finding the distances between a query and all the examples in the data, selecting the specified number examples (K) closest to the query, then votes for the most frequent label (in the case of classification) or averages the labels (in the case of regression).





The accuracy rate of KNN is around 67 – 68 %.

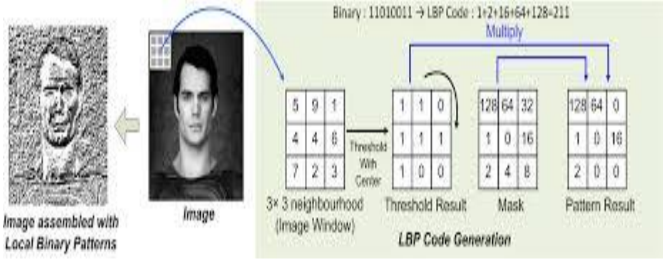
LBPH algorithm

Local Binary Pattern (LBP) is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number. LBP was first described in 1994 and has since been found to be a powerful feature for texture classification. It has further been determined that when LBP is combined with histograms of oriented gradients (HOG) descriptor, it improves the detection performance considerably on some datasets. Using the LBP combined with histograms we can represent the face images with a simple data vector

Four parameters of LBPH

• Radius: the radius is used to build the circular local binary pattern and represents the radius around the central pixel. It is usually set to 1

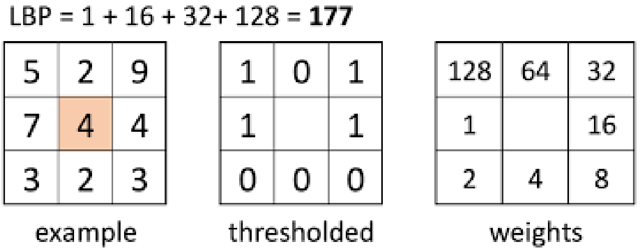
• Neighbors: the number of sample points to build the circular local binary pattern. Keep in mind: the more sample points you include, the higher the computational cost. It is usually set to 8.



• Grid X: the number of cells in the horizontal direction. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector. It is usually set to 8.

• Grid Y: the number of cells in the vertical direction. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector. It is usually set to 8.

As above shown the working and principles of the LBPH algorithms.



The accuracy rate of LBPH is around 75 – 80 %.

III. Comparisons in between LBPH Vs KNN

	Recognition Rate (%)					
Method	50% 50%		70% 30%		90% 10%	
LBP +KNN Using:	Yale	ORL	Yale	ORL	Yale	ORL
Euclidean Distance	91.1	93	95.6	93.3	100	94
Correlation Distance	90	94	97.8	94.5	100	95
Canberra Distance	91.1	95	100	96	93.3	98
Manhattan Distance	94.4	96.5	95.6	97.3	100	99
Mahalanobis Distance	94.4	97	100	97.5	100	98

Hence above shown the LBPH algorithm is best way for facial Recognition

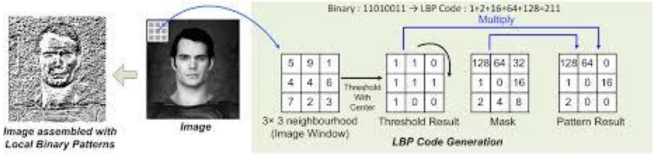
IV. RESULTS AND IMPLEMENTATIONS

Training the Algorithm

First, we need to train the algorithm. To do so, we need to use a dataset with the facial images of the people we want to recognize. We need to also set an ID (it may be a number or the name of the person) for each image, so the algorithm will use this information to recognize an input image and give you an output. Images of the same person must have the same ID. With the training set already constructed, lets see the LBPH computational steps.

Applying the LBP operation

The first computational step of the LBPH is to create an intermediate image that describes the original image in a better way, by highlighting the facial characteristics. To do so, the algorithm uses a concept of a sliding window, based on the parameters-radius and neighbors. The image below shows this procedure


Binary : 11010011 → LBP Code : 1+2+16+64+128=211
Multiply
Threshold With Center
3× 3 neighbourhood (Image Window)
Threshold Result
Mask
Pattern Result
LBP Code Generation
Image
Image assembled with Local Binary Patterns

5	9	1
4	4	6
7	2	3

1	1	0
1	1	1
1	0	0

128	64	32
1	0	16
2	4	8

128	64	0
1	0	16
2	0	0

Based on the image above, lets break it into several small steps so we can understand it easily:

• Suppose we have a facial image in grayscale.

• We can get part of this image as a window of 3x3 pixels.

• we can also be represented as a 3x3 matrix containing the intensity of each pixel (0 255).

• we need to take the central value of the matrix to be used as the threshold.

• This value will be used to define the new values from the 8 neighbors.

• For each neighbor of the central value (threshold), we set a new binary value. We set 1 for values equal or higher than the threshold and 0 for values lower than the threshold.

• Now, the matrix will contain only binary values (ignoring the central value). We need to concatenate each binary value from each position from the matrix line by line into a new binary value (e.g. 10001101). Note: some authors use other approaches to concatenate the binary values (e.g. clockwise direction), but the final result will be the same.

• Then, we convert this binary value to a decimal value and set it to the central value of the matrix, which is actually a pixel from the original image.

• At the end of this procedure (LBP procedure), we have a new image which represents better the characteristics of the original image.

• It can be done by using bilinear interpolation. If some data point is between the pixels, it uses the values from the 4 nearest pixels (2x2) to estimate the value of the new data point.

Based on the image above, we can extract the histogram of each region as follows

• As we have an image in grayscale, each histogram (from each grid) will contain only 256 positions (0 255) representing the occurrences of each pixel intensity.

• Then, we need to concatenate each histogram to create a new and big- ger histogram. Supposing we have 8x8 grids, we will have 8x8x256=16.384 positions in the final histogram.

The final histogram represents the characteristics of the image original image. The LBPH algorithm is pretty much it.

V. PERFORMING THE FACE RECOGNITION

To find the image that matches the input image we just need to compare two histograms and return the image with the closest histogram.

• We can use various approaches to compare the histograms (calculate the dis- trance between two histograms), for example: Euclidean distance, chi-square, absolute value, etc. In this example, we can use the Euclidean distance (which is quite known) based on the following formula

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

• So the algorithm output is the ID from the image with the closest histogram. The algorithm should also return the calculated distance, which can be used as a confidence measurement.

VI. CONCLUSION

Occlusion face detection is an active research area and the technology has come a long way since the survey of Viola Jones. Some of the best algorithms dealing with complex environments are computationally expensive for real-time processing. However, along with the improvements in computer hardware, this situation is likely to change.

Therefore, future work can be done in the direction of implementing an algorithm capable of tracking and recognizing human faces in different background with accurate rate. Parallel algorithms, which speeds up the computation, can be adopted.

The face detection is a heated topic in the last decade, there has been many problems with that, and occlusion is a major drawback in this issue. Many places where face detection is required majorly in security.

In addition, it is very difficult to recognize the individual when faces are occluded. Therefore, research in this field is very important and now when occlusion is detected and still the person can be found it even in the occluded image that is where the problem is solved.

REFERENCES

[1] Patel, T. B.,& Patel, J. T. (2017). Occlusion detection and recognizing human face using neural network. 2017 International Conference on Intelligent

[2] Computing and Control (I2C2). Wang, H., Gu, X., Li, X., Li, Z., & Ni, J. (2015). Occluded Face Detection Based on Adaboost Technology. 2015 Eighth International Conference on Inter-net Computing for Science and Engineering (ICICSE).