Identifying Actual Working Hours of Drivers Using Face Recognition

Vinu Reji John Department
of Computer Applications
Amal Jyothi College of Engineering Kanjirappally,
India
vinurejijohn@mca.ajce.in

Shelly Shiju
George
Asst professor Department of Computer
Application. Amal Jyothi College of
Engineering Kottayam,India
shellyshijugeorge@amaljyothi.ac.in

Abstract: Identify the challenges that drivers experience in a car rental system to examine their actual working hours. We provided an improved method for improving human face recognition in this paper using LBPH (Local binary pattern histogram). The key contribution of this paper is about the live training dataset. In the face recognition process, data and feature reduction are critical, and researchers have recently concentrated on the current neural network. As a result, we used a local binary pattern histogram descriptor to show that even with standard approaches, there is room for improvement.

Introduction

[4] Even in today's technologically advanced world, the great majority of businesses still use a manual attendance system in which employees manually record their entry and leave times in attendance papers. This places the burden on each individual employee to manually count the number of working hours for each day, which is not only a time- consuming operation but also frequently does not reflect the actual amount of time that they work. The above-mentioned system can be effective in case of marking general attendance of the employees and to find the amount that he worked. This paper aims on calculate the actual working hours of drivers using face recognition system. The purpose of this article is to ensure that the system recognizes the proper individual, we employ LBP (Local Binary Pattern) to recognize the face for better accuracy. [2] LBP is a basic yet effective texture operator that labels pixels in an image by thresholding each pixel's neighborhood and treating the result as a binary number.[2] It was also discovered that when LBP is paired with histograms of oriented gradient descriptors, the detection performance on specific datasets is improved. We can represent the face image with a simple data vector using LBP and histograms.

Taxi and rental companies have a tough time determining the actual working hours of their drivers. In this system we are trying to perform face recognition of drivers whenever the enter into the car. Driver reporting activities are tracked by the software system. Drivers must scan their faces upon arrival, at which point the system records the time and stores it in the cloud. At returning the system automatically finds that the drivers are out. As a result, the system saves the information of each entry time and leaving time of driver. By combining each data’s, the system admin can determine the actual working hour of drivers and can pay them based on their working hours.

According to Patrick KAMENCAY [5] The process of the face recognition starts with two image sets: the gallery, and the reference set. The gallery consists of frontal face images. The goal is to rank the gallery images based on their similarity to a given probe image.

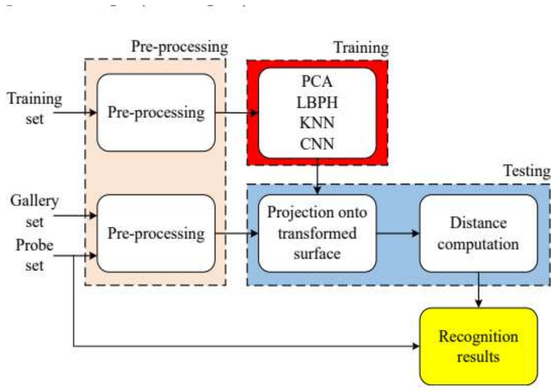


Fig. 1: Example of the face recognition system.

DOI: 10.5281/zenodo.6367449
ISBN: 978-93-5607-317-3@2022 MCA, Amal Jyothi College of Engineering Kanjirappally, Kottayam

The reference-based descriptors for the gallery images are derived after pre-processing by comparing the gallery and reference set feature vectors. The face recognition algorithms must deal with a lot of lighting differences between the gallery and probing photos.

1. Training the system

[5] Because a facial image can be considered as a compilation of micro-texture patterns, the texture descriptor Local Binary Patterns (LBP) can also be used to represent faces. In a nutshell, the technique entails partitioning a facial picture into various areas, from which the LBP features are extracted and concatenated into a feature vector, which will later be utilized as a facial descriptor.

[1] Because of the variety of facial emotions, personal appearances, positions, and lighting, human face identification is a difficult undertaking. The primary area of this system is to train the system to recognize the correct individuals. It is difficult to create a real-time face recognition system due to the unpredictability in lighting intensity and direction, the number of light sources, and the camera's orientation. Because the object identification system demands a large amount of computational space, reducing the image dimension is important to optimize classification processing time. The LBPH method is one of the most widely used traditional methods; it is used for data representation, histograms, and feature reduction.

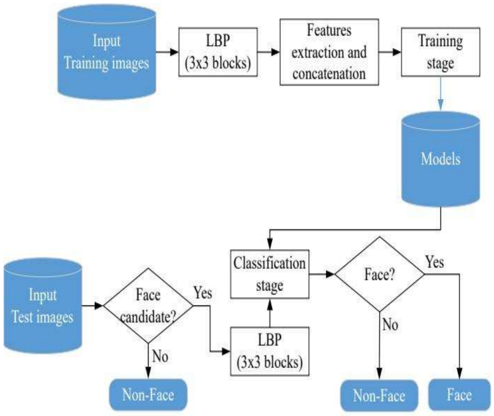
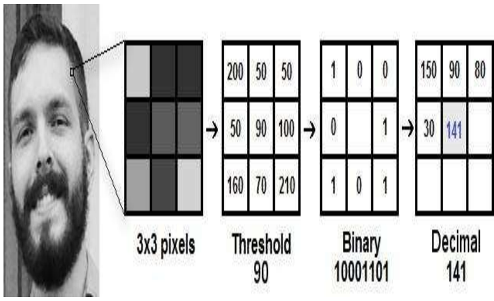


Fig. 4: Block diagram of a LBP algorithm.

[1] The first phase is preprocessing, which includes a variety of processes such as picture registration, scaling, face normalization, background noise reduction, detection, and resizing, all of which have an impact on the accuracy of face recognition.

[1] The second phase is feature extraction, which can be accomplished by employing sophisticated transformation techniques. By keeping important elements, the image dimension can be reduced to a lower size. Image descriptors such as Gabor wavelets and LBP are used in some of the image descriptors.

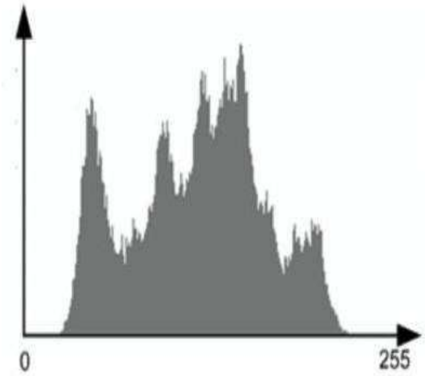
[1] The classification step, which uses sophisticated classifiers such the BPNN and fully connected NN Support Vector Machine (SVM), Euclidean distance classifier MahalaNobis distance classifier Hidden Markov Models, and extreme learning machine, is the last phase.



[1] Each uniform pattern has its own bin, and all non-uniform patterns have their own bin in the histogram. We have 58 uniform patterns in an 8-bit binary number, thus we used 58 bins for those and one bin for all non-uniform patterns. Concatenating all regional histograms yields a global description of the facial image. In a histogram, the overall value of LBPH is represented as:

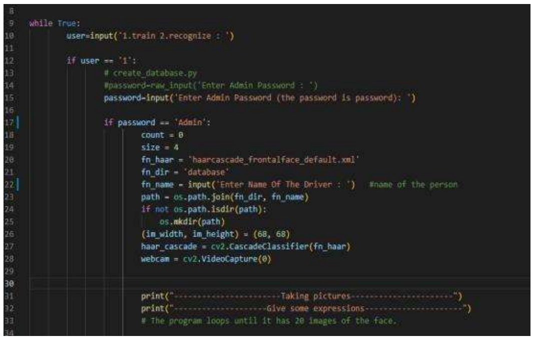
$$H(k)=\sum_{i=0}^{n}\sum_{j=1}^{m} f(LBP_{P,R}(i,j),k),k\in[0,k]$$

Based on this decimal value system generates a histogram.

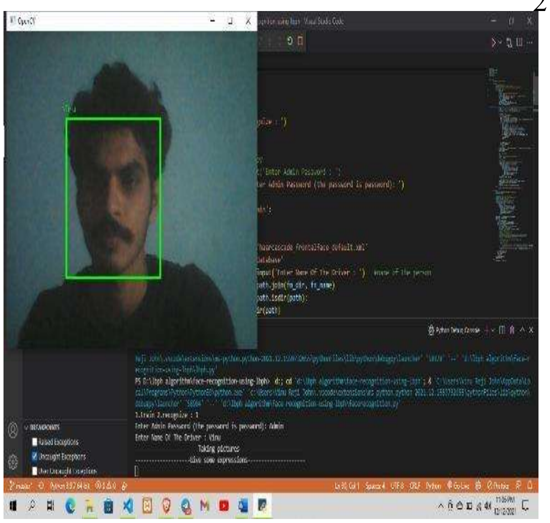


The user must first run facerecognition.py to register his face, then enter his password and name after selecting 1 to train the algorithm. The algorithm collects 64 photos of the user and saves them in a database folder. After that, the programme applies LBPH to the pictures and saves the learned classifier to trainedRec.py.

Given below is the python code for training a dataset:



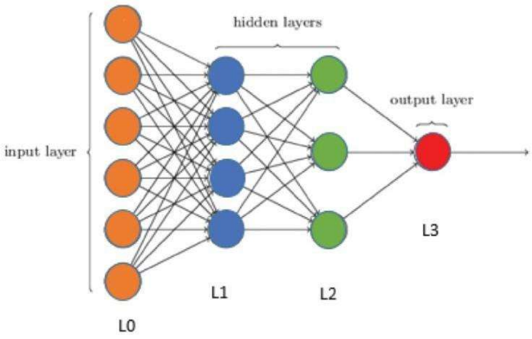
```
while True:
        user=input('1.train 2.recognize : ')

        if user == '1':
                # create_database.py
                #password=raw_input('Enter Admin Password : ')
                password=input('Enter Admin Password (the password is password): ')

                if password == 'Admin':
                        count = 0
                        size = 4
                        fn_haar = 'haarcascade_frontalface_default.xml'
                        fn_dir = 'database'
                        fn_name = input('Enter Name Of The Driver : ')   #name of the person
                        path = os.path.join(fn_dir, fn_name)
                        if not os.path.isdir(path):
                            os.mkdir(path)
                        (im_width, im_height) = (68, 68)
                        haar_cascade = cv2.CascadeClassifier(fn_haar)
                        webcam = cv2.VideoCapture(0)


                        print("-----------------------Taking pictures----------------------")
                        print("--------------------Give some expressions---------------------")
                        # The program loops until it has 20 images of the face.
```



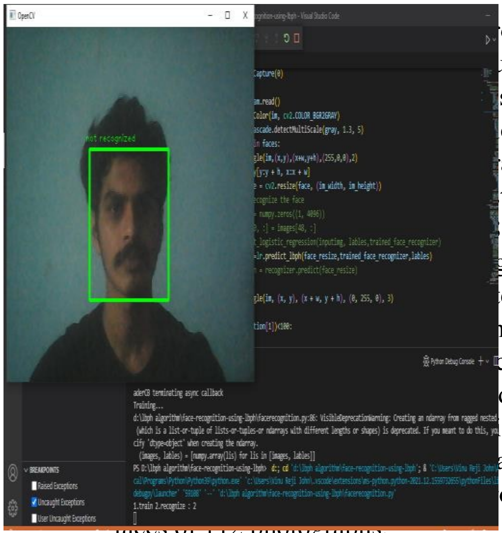
```
1.train 2.recognize : 1
Enter Admin Password (the password is password): Admin
Enter Name Of The Driver : Vinu
                    Taking pictures
-----------------Give some expressions------------------
```

2. Recognizing

[1] To attain a high recognition system rate with little computation time and resources, computer vision requires robust categorization methods. Because BPNN is simple, efficient at computing gradient descent, and easy to implement, it is commonly used for training neural networks. It is difficult to determine the size of the NN, the number of samples, and the weights, but it is necessary to suit the NN output. As indicated in the diagram, the BPNN is separated into three levels: the input layer, one or more hidden layers, and the predictable output layer.

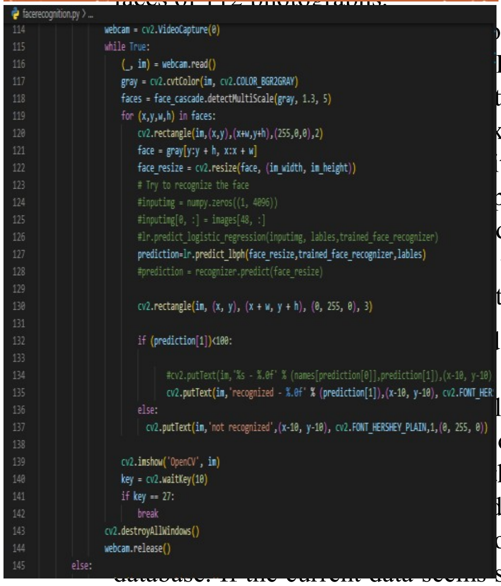


DOI: 10.5281/zenodo.6367449
ISBN: 978-93-5607-317-3@2022 MCA, Amal Jyothi College of Engineering Kanjirappally, Kottayam



```
Capture(0)

am.read()
Color(im, cv2.COLOR_BGR2GRAY)
ascade.detectMultiScale(gray, 1.3, 5)
in faces:
gle(im,(x,y),(x+w,y+h),(255,0,0),2)
y[y:y + h, x:x + w]
e = cv2.resize(face, (im_width, im_height))
ecognize the face
= numpy.zeros((1, 4096))
0, :] = images[48, :]
t_logistic_regression(inputimg, lables,trained_face_recognizer)
=lr.predict_lbph(face_resize,trained_face_recognizer,lables)
n = recognizer.predict(face_resize)

gle(im, (x, y), (x + w, y + h), (0, 255, 0), 3)

tion[1])<100:
```

```
aderCB terminating async callback
Training...
d:\lbph algorithm\face-recognition-using-lbph\facerecognition.py:86: VisibleDeprecationWarning: Creating an ndarray from ragged nested
 (which is a list-or-tuple of lists-or-tuples-or ndarrays with different lengths or shapes) is deprecated. If you meant to do this, yo
cify 'dtype=object' when creating the ndarray.
  (images, lables) = [numpy.array(lis) for lis in [images, lables]]
PS D:\lbph algorithm\face-recognition-using-lbph> d:; cd 'd:\lbph algorithm\face-recognition-using-lbph'; & 'C:\Users\Vinu Reji John\
cal\Programs\Python\Python39\python.exe' 'c:\Users\Vinu Reji John\.vscode\extensions\ms-python.python-2021.12.1559732655\pythonFiles\li
debugpy\launcher' '59108' '--' 'd:\lbph algorithm\face-recognition-using-lbph\facerecognition.py'
1.train 2.recognize : 2
```



```
            webcam = cv2.VideoCapture(0)
            while True:
                (_, im) = webcam.read()
                gray = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
                faces = face_cascade.detectMultiScale(gray, 1.3, 5)
                for (x,y,w,h) in faces:
                    cv2.rectangle(im,(x,y),(x+w,y+h),(255,0,0),2)
                    face = gray[y:y + h, x:x + w]
                    face_resize = cv2.resize(face, (im_width, im_height))
                    # Try to recognize the face
                    #inputimg = numpy.zeros((1, 4096))
                    #inputimg[0, :] = images[48, :]
                    #lr.predict_logistic_regression(inputimg, lables,trained_face_recognizer)
                    prediction=lr.predict_lbph(face_resize,trained_face_recognizer,lables)
                    #prediction = recognizer.predict(face_resize)

                    cv2.rectangle(im, (x, y), (x + w, y + h), (0, 255, 0), 3)

                    if (prediction[1])<100:

                        #cv2.putText(im,'%s - %.0f' % (names[prediction[0]],prediction[1]),(x-10, y-10)
                        cv2.putText(im,'recognized - %.0f' % (prediction[1]),(x-10, y-10), cv2.FONT_HER
                    else:
                      cv2.putText(im,'not recognized',(x-10, y-10), cv2.FONT_HERSHEY_PLAIN,1,(0, 255, 0))

                cv2.imshow('OpenCV', im)
                key = cv2.waitKey(10)
                if key == 27:
                    break
            cv2.destroyAllWindows()
            webcam.release()
        else:
```

rocess of
ultimedia
sing any
c graph
randomly
tic graph
7 people
g various
ernation,
hed. The
compute;
d objects
ers takes
aked the
ont view
faces of 112 photographs.

on Local
PH) are
t stage of
xtract the
ithm. The
pattern is
ds. After
to detect
tographs.

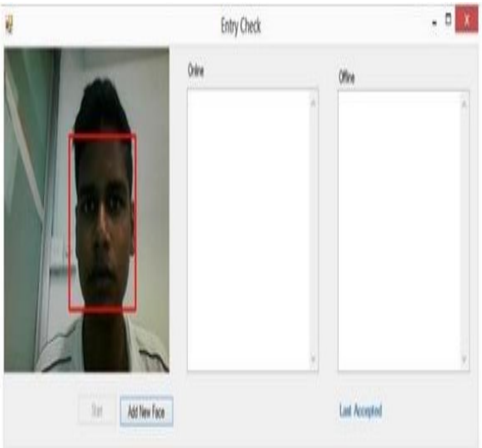
l using local
and face
la jones haar
of the system
he data. The
data with the
ored in the
database. If the current data seems similar to the any of the datasets in the database, then the system will give output.as recognized.

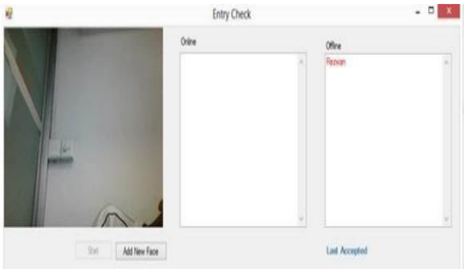
The user can choose recognize to see if he is an authorized person. To increase the sensitivity of face recognition, the threshold posed by the condition if (prediction [1])
<100 needs to be altered.

The working of recognition is done by comparing the decimal values and histogram of the datasets and the current data. The system calculate plot the histogram of the current data and search for the same in the database. By analyzing those values, the system identifies the data.

The mild AdaBoost algorithm and Haar feature techniques are used to train the classifiers.[4] It requires two sets of photos, one of which does not contain the object to be detected, in this example a face characteristic. The negative images are the set of images that make up a photograph. The positive images, on the other hand, include one or more occurrences of the face. The picture name, the upper left pixel, and the object's height and width are used to specify the location of the objects within the positive photographs. The positive set of photos travels through all stages of the cascade classifiers, yielding a face region.

[4] The faces of drivers are recognized at their workstations in the proposed system by recognizing Haar-like features in order to evaluate their presence. When a face is discovered and thus detected, a red square box is drawn around it to indicate that it has been detected.





By the study of Md. Zahangir Alom [4] For the detection of detected faces from grey photos, the Principal Component Analysis (PCA) algorithm is used. By concatenating each row (or column) into a long thin vector, a 1-D vector can represent a 2-D facial image. Assume that a set of sampled images is represented by M vectors of size N (= rows of image x columns of image). The eigenvectors generated by PCA are called eigenfaces because they are in the direction of the training vectors' highest variation.

Each eigenface can be thought of as a distinct feature. When a face is projected onto the face space, its vector represents the relative importance of each of the features in the face. The eigenface coefficients (or weights) of a face are stated in face space. The small weight vector in the face space can be used to manage a big input vector, such as a facial image. Because the dimensionality of image space is substantially higher than that of face space, the original face is reconstructed with some mistakes.

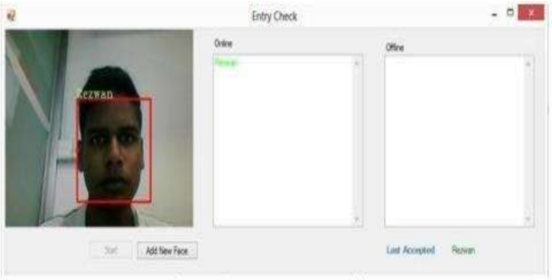
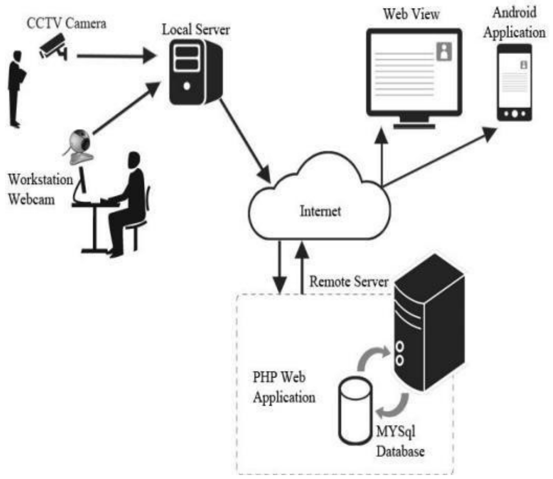


Figure 8: Face recognition

3. Finding actual working hours

At the entrance to the vehicle documenting driver arrival and departure times. Individual workstation webcams record real-time video of their matching vehicles. [4] The application for processing all of the video feeds supplied from the workstation webcams in order to identify whether an driver is present in his desk or not is executed on the local server at the office campus. The information is then transferred to a distant server, which keeps track of how much time each employee spent at their desk, as well as the timestamps of their arrival at work and leave at the end of the day. Employee profiles can be used to see statistical analysis of employee activity in the office via a server-side web application. Whenever the driver arrives into the vehicle the system scans the drivers face and finds if it is a authorized person or not. If so, the system will store the current date and time into the corresponding database of the driver. Also stores the date and time when the driver leaves from the vehicle. The process will continue whenever the driver enters and leaves the vehicle.



Future works

[5] In addition, the system can be divided into numerous modules. We can implement paying the drivers income based on their working hours in addition to identifying real working hours. The system will automatically calculate the daily wage of the driver based on the actual working hours and deposit it to the driver's account. Also, the system can generate a report and send it to both the admin and the driver.

We can also create an automatic payment system on the user's side as an add-on. The system will detect customers, and when they depart, the appropriate amount will be immediately deducted from the customer's account.

Conclusion

The system is based on LBPH algorithm to recognize face of drivers. After successful recognition of drivers, the data will be stored to drivers' cloud and the daily wage will be automatically paid to drivers' account. The system uses LBPH algorithm to train and recognize drivers, cloud storage to store the data, ClearPoint tool for report generation. The overall results were acquired by varying the number of training and test photos used. Large training datasets result in a significant computing load and memory utilization, requiring substantial processing power to be applied effectively. In our situation, the largest evaluated face dataset (BioID Face Database) has 1521 greyscale images with a resolution of 384 286 pixels

References

1. Mohammad A. Albuneid
   Department of Computer Science and Engineering, University of Bridgeport, Bridgeport, CT, USA.
   Publisher: IEEE
   Published on 10 April 2018

2. Parth Singh
   Understanding face recognition using LBPH
   Published on: July 12, 2021

3. Farah Deeba1, Aftab Ahmed4
   School of Information and Software Engineering, University of Electronic Science and Technology of China
   Chengdu, Sichuan, China

4. Md. Zahangir Alom et al
   School of Engineering and Computer Science, BRAC University, Dhaka, Bangladesh

5. Patrik KAMENCAY et al
   Department of Multimedia and Information-Communication Technologies, Faculty of Electrical Engineering,
   University of Zilina, Slovakia
   Published on 2017