

An Intelligent Personal Assistant Relevance

Students: Beshoy Nady Adly, Tomas Azer Ewnan, Menna Mohammed Hagrss, EL shimaa Mohammed Slime, Marian Nady, Mahmoud Hesham , Norhan Mohammed and Shihab Raafat
Email: 418180523@sha.edu.eg

Supervisor: Assoc. Prof. Emad S. Othman

Senior Member IEEE - Region 8, High Institute for Computers and Information Systems, E-mail: *AL-Shorouk*
Academy, Cairo – Egypt, PH- 0020-010-25830256. Email: emad_othman@sha.edu.eg

Abstract- Despite the great development that has occurred in deep learning field, the problem still persists, and this negates the main reason for creating software, which is to facilitate business, trade, transactions, communication, and others. Deep learning methods are computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction. In this paper, we implemented internet searching on Google and YouTube using voice commands then displaying results, recognizing faces and fingerprints for users to improve security, gathering information, answering questions and reading them to the user, fully controlling all files and folders on the computer and carrying out all common operations on them such as (add, delete, move, rename, create).The system uses Speech Recognition API introduced by Google, Local Binary Patterns Histogram (LBPH) to perform face/object recognition. The evaluation is performed using Histogram analysis and the results show superiority of the model and the efficiency of LBHP algorithm in various applications for facial recognition.

Keywords: Deep Learning, Face Recognition, Object Recognition, Local Binary Patterns Histogram (LBPH).

1. INTRODUCTION

In daily life, we are affiliated with many tasks and have many needs, which lead us to rely on using applications in order to overcome such issue .For example ,we use Uber application in case transportation is needed .We use some commercial application in ordering food, goods and delivering it. These applications have proven their efficiency and power in dealing with these problems.Over the last decade, the use of Deep Learning in many applications produced results that are comparable to and in some cases surpassing human expert

performance. The application domains include diagnosing diseases, finance, agriculture, search engines, robot vision, and many others. Deep learning methods have dramatically improved the state-of-the-art speech recognition, visual object recognition, object detection and many other domains such as drug discovery and genomics. Deep learning discovers intricate structure in large data sets by using the back propagation algorithm to indicate how a machine should change its internal parameters that are used to compute the representation in each layer from the representation in the previous layer [1-3].

Deep convolutional networks have made breakthroughs in processing images, video, speech and audio, while redundant networks shed light on sequential data such as text and speech .For example, nowadays, there is no need to learn to write on the keyboard quickly or even understand how the computer works .Using speech recognition, it is more sufficient to do this by directing voice commands only . It helps the disabled to deal with computers easily without any problems [4].

2.ALGORITHM

A. Speech Recognition

It is the main part of this application. It can listen to users' commands, understand them, and can give answer about users' questions and requests. We used Python programming language during implementation because Python has a lot of libraries and APIs. We used Google API called "Speech Recognition". This API can convert speech to text and text to speech. The first step is to "Initialize the Recognizer" with this command (`r = sr.Recognizer()`) and listen to the audio via the source (`audio = r.listen`

(source, 5,5) and "convert audio to text" by (voice_data = r.recognize_google (voice)). This API sometimes has some failures. The first error is "Error: the recognizer doesn't understand". The app overcomes this error and sends this message: "engine speak ("I didn't understand that")". It will convert this error message to speech. The second error is "Error: Recognizer is offline". The app overcomes this error and sends this message: (engine speak("Sorry, the service is down")).It will convert this error message to speech. We used this method because it is easy to understand by ordinary people [5-7].

B. Speech recognition results

We have three types of results; the first type is audio: the app accepts the audio and converts it into actions like "open chrome" or "open YouTube". The second type is text like the user asking the app about something. In this case, we have three types of results: (text) which is the answer to a user's question, (action) the process of the application to find the answer, and (audio) when that answer is read [8].

C. Face Recognition

After a long-time thinking about making a personal assistant on the computer to control the cable through voice commands to facilitate communication for ordinary people and people inexperienced in the field of computers, we had to invent or use a new method to secure user data and with the widespread use of face ID in smartphones. It was easy to use for computer users. We used face recognition by applying three steps, each of which plays a role in the face recognition process. The first step is to use the Open_CV library to use the computer camera to get 30 user photos by opening a video to capture the images and keep them in the dataset. The second step is to train the model on that data, extract the characteristics of those images to identify them in the third step by using the Local Binary Patterns Histogram algorithm (LBPH). It works on the pixels from the middle and converts them all into numbers and then into an array Binary, then store and refer to it for analysis and to know the extent of congruence [9-11].

A. Local Binary Patterns Histogram (LBPH)

An approach to achieving this goal is to use the LBP texture descriptors to create abundant local facial descriptions and combine them into a global description. These local characteristics are primarily based on ways in which there are many robust causal or illumination variations compared to traditional. The essential methodology for LBP is based primarily on the facial description proposed by Ahonen et al.. It starts by dividing the facial image into local areas and extracting the LBP texture descriptors separately from each area. Then the descriptors are joined together to describe the face globally. Effectively, this histogram features a face description on three completely different location levels: The LBP histogram labels contain data on unit pixel-level patterns, labels are added over a small region to integrate data at a regional level, and local histograms are also joined together to provide a global facial description. In face identification method, it has been marked into 3 stages. Initially, a new picture is used for illustrating their properties, namely, "Integral Image". Then, an easy AdaBoost classifier is utilized to choose a tiny variety of crucial visual highlights from a really massive set of effective properties. In the end, it is important to collaborate with the classifiers during a cascade to extract the background regions of the picture.

Local Binary Patterns (LBP) is a perceptible descriptor style used in the classification of computer vision. LBP is the specific case of the 1990 proposed Texture Spectrum model. In 1994, LBP was represented for the first time. Since then, it has been found as a solid element for classifying texture. More specifically, once LBP is combined with the descriptor histogram of oriented gradients (HOG). It obviously improves the execution of identification on some datasets. Figure 1 illustrates the LBPH algorithm flowchart diagram [12].

The image is divided into cells (4 x 4 pixels) for the encoding of features. It is contrasted by using a clockwise or counter-clockwise bearing of surrounding pixel values. The value of each neighbor's intensity is compared to the central pixel. The location is assigned a 1 or a 0 depending on the difference whether it is higher or lower than 0. The

result gives a single cell an 8-bit value. Figure 2 shows the matrix calculation comparing the value of the middle element of the matrix with the neighboring elements [13].

If the illumination condition of the image is changed, the final result is equivalent to the previous result. Histograms are used in larger cells as well as the frequency of values that make system robust. Edges can be identified as the quality changes by dissecting the results in the cell. It is possible to obtain feature vectors by calculating the values of all cells and connecting the histograms. Pictures can be grouped by ID-connected handling methods. Input pictures are classified using the same procedure and the data set is contrasted and separation is obtained. It is very well recognized by setting a limit value whether it is a known or obscure face. Figure 3 represents the values of the matrix when the light intensity is wavering

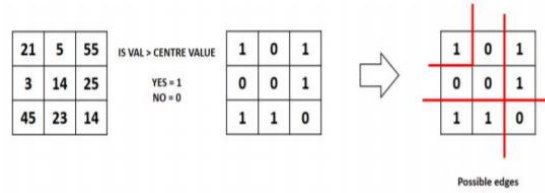


Figure 2: LBPH creating an 8-bit number

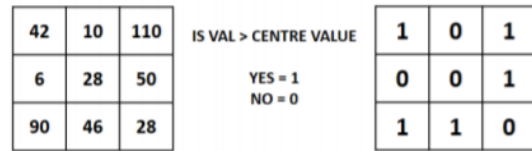


Figure 3 : If the brightness changes, the results will be the same.

D. Methodology

The proposed face recognition approach has four main steps: module for image acquisition, module for extraction of features, module for training classifier database, and module for classification. Initially, the face datasets are collected by the image acquisition module. Then, a series of salient features are extracted by applying feature extraction module. These facial features are used to analyze face landmarks which represent human identity information. In the next process, the classifier is trained for recognizing the face. In the last module, the system recognizes face image and fetches information about the person from the MYSQL database. The system flow chart is shown in Figure. 4.

3.FACE DETECTION

For face detection, OpenCV is used that introduces a Haar cascade classifier .The classifier of the Haar cascade uses the AdaBoost algorithm to locate numerous image facial features. Initially, it takes an input image using the camera and converts that color image into a grayscale image. After that, it loads Haar cascade classifier for determining whether the image contains any faces in the frame or not. When any face is detected, other facial features are checked and a square frame is drawn on the face. Otherwise, it starts reading other pictures. Figure 5 shows the flow chart of the detection process [14].

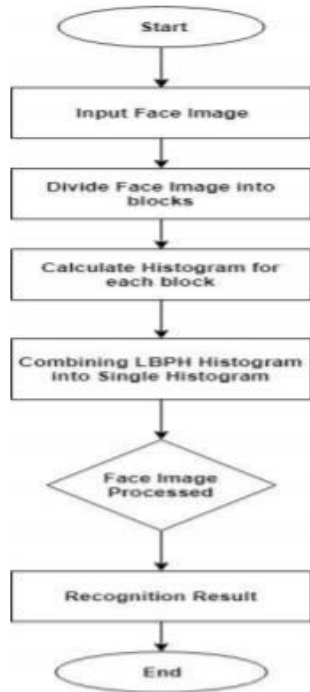


Figure 1 : LBPH Algorithm Flowchart

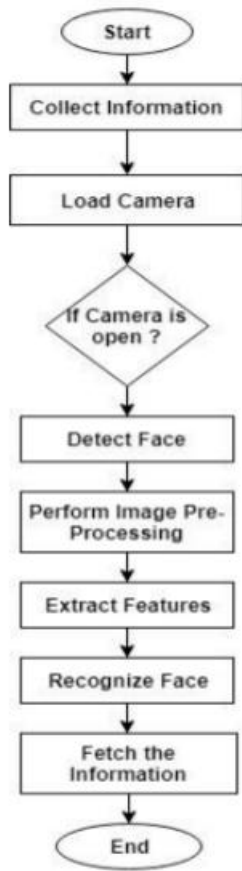


Figure 4 : Face recognition system workflow

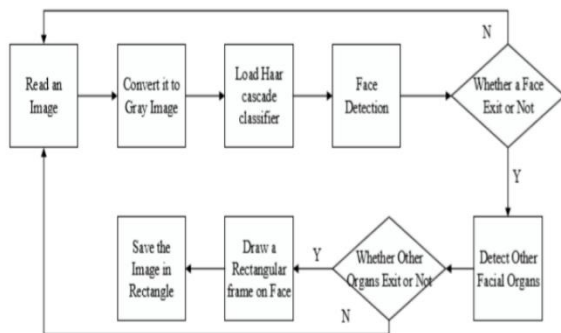


Figure 5 : Face detection flow chart.

A. Feature Extraction

For extracting the facial features from image, the LBP operation is used that compares the intensity value of every component with the 8 nearest neighbour pixels values. If the value of the neighboring pixel is greater than the value of the centered pixel, it will assign 1 to its neighboring pixel, otherwise it

will assign 0. For each pixel, this task provides an 8-bit string. A decimal value of an 8-bit pixel string determines the LBP value. Fig. 6 shows this operation.

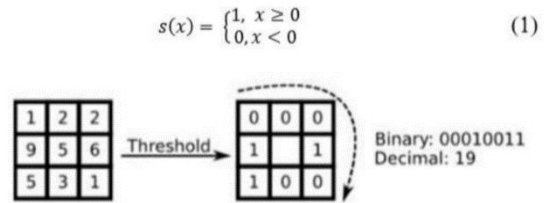


Figure 6 : Extraction of LBP feature from the image

The input image is divided into many small sub-images after the application of the LBP operation and the histograms of the LBP value of each sub-images are extracted. Then all histograms are linked to make an image-representing feature vector and used to train a facial recognition classifier [15-16].

B. Dataset

We used a benchmark dataset, with 30 individual photos. Throughout the image acquisition process, face images are cropped and converted into gray images, then these images are saved in the same folder to make face databases for extraction tasks. After this, the standardization technique is applied to all images to reduce noise and set the correct image scaling position to quickly obtain the result of recognition. Dataset images are shown in Figure 7

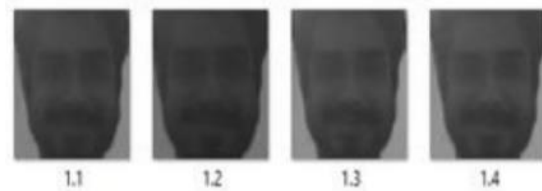


Figure 7 : Dataset images

C. Face Recognition

For the face recognition process, Local Binary Pattern Histogram algorithm is applied. The LBP operator uses local binary patterns to reduce the local spatial distribution of a face image. The LBP operator

is a collection of binary pixel value ratios in the center at regular pixel intervals and is around 8 pixels.

It is shown in the below equation

$$LBP(x_c, y_c) = \sum_{n=0}^7 S(i_n - i_c)2^n \quad (2)$$

Where i_c represents the index value of the middle pixel and (x_c, y_c) shows 8 close surrounding pixels data.

D. Experimental Setup

We developed the system using python programming language. OpenCV (3.0) library and other tools Visual Studio Code and MYSQL are used in this system. Visual Studio Code is an integrated development Environment (IDE) employed for creating computer programs, specifically for the Python language. MYSQL is an RDBMS contained in a C programming library. As compared to other database management systems. The whole project is divided into three parts facial detection, training face images and last the recognition of the faces. These all 3 operations are performed in Visual Studio Code and information regarding a person like a name, age, gender is kept in the MYQL database

E. Results and Discussion

To develop a face recognition system, first we have to prepare the face dataset for training. To develop face dataset, we used Face detection method that detects the face in real time camera and captured face images. The captured images are saved into dataset folder for feature extraction and training processes. In the initial phase, system asks information about the person like name, age, gender and then the camera will open that captures 30 images of a person on different face position and expression. The information about person is saved into MYSQL database and captured images are saved into dataset folder with same unique id.



Figure 8 : Face detection

In the next process, the system extracts the LBP texture features from every input dataset image and train them with the Haar Cascade classifier

TABLE I. Training Images Statistic

Total Images	Trained Images	Training Time
30	30	0.6sec

Finally, the system performed face recognition process. First it detects all faces in the image, then it extracts facial feature from the input test face image. After this input feature vector is compared with the trained image dataset model using Haar Cascade Classifier. If the input test feature vector is matched with the trained model, it recognizes the face and fetches the information about that image from the MYSQL database. The system can detect multiple faces in the image if any new face image comes into the camera frame system recognize the face on base on facial features



Figure 9 : Recognizing face images.

F. (Face & Object) Recognition Result

We have two types of results, when App accepts audio command like "face recognize" after user say that we have two cases. The first case is: if app identified the user so the result will be (Action) when app open Came and "Show face Recognize Result " and set border on face user and display second type of result, which is (Text) through displaying username above the border. The second case is: if App couldn't identify the user, then it will use open cam and set border on user face and set text above the border "Unknown" and app will Ask user if him like to create Face ID if user Accepts that, the app will recognize face and save some images for user face in dataset and training on all dataset and ask user about him name to save it, but if user not accepts to create Face ID the app will close this window and ask user how can I help you ...?

The Stages of Classification and How the LBP-Descriptor Works are shown in Figure 9 and 10 respectively.

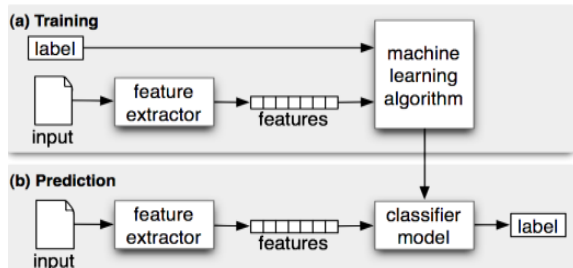


Figure 9: Stages of Classification

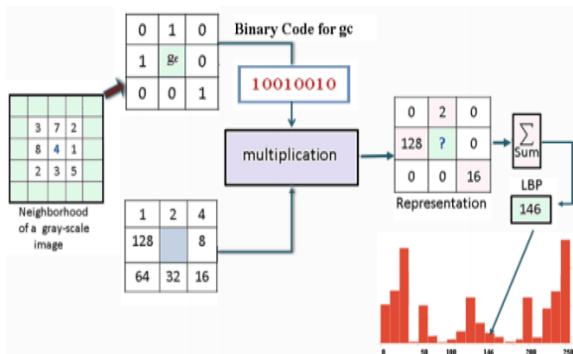


Figure 10: How the LBP- Descriptor Works

4. CONCLUSION

After a lot of experiments on the application and after making sure that it meets the need for which it

was made, which is to facilitate the process of dealing between ordinary people and computers in general. It can be said that it has many capabilities more than similar applications in the market such as (Google Assistant, Siri) which is limited to a course only on smart phones, but this application (Software Assistant Robot) can fully control the computer and perform all the operations that can be performed by the user on files and folders, access the Internet, browse YouTube, search for questions and find the answers and the creation of research that consists of many pages that talk on one of the topics and can add a special section for safety to the data where it can create fingerprints for the faces to increase the safety of use and identify things to add entertainment and learning for children and it can be after some time and after increasing work on it Interfering with finer details, such as adding the voice fingerprint as an alternative to the face print to increase the security of the application and tracking and categorizing user searches that can be exploited to give the user alerts by notifications to anyone.

The new developments that occur in the world about the things that the user cares about, such as the (Programming Language) field, in which the application can display new events by means of notifications on the computer for those interested in this, and this is only an example of how this feature works, after a lot of time and continuous work on This slowdown is because humans no longer need to do anything with their mouse and keyboard anymore they can just say whatever they want and it will happen.

REFERENCES

- 1) HU Liqiao, Q. Runhe. Face recognition based on adaptive weighted HOG. Computer Enigeering and Applications, 2017, 53(3): 164-168.
- 2) CHAO W L, D. J J, L. J Z. Facial expression recognition based on improved local binary pattern and class-regularized locality preserving projection. Signal Processing, 2015
- 3) HU Liqiao, Q. Runhe. Face recognition based on adaptive weighted HOG. Computer Enigeering and Applications, 2017

- 4) Yu yan JIANG, P. LI, Q. WANG. Labeled LDA model based on shared background topic. *Acta Electronica Sinica*, 2015, 2013
- 5) WU Qi, W. Tang-hong, L. Zhan-li. Improved face recognition algorithm based on Gabor feature and collaborative representation, *Computer Engineering and Design*, 2016
- 6) HU Liqiao, QIU Runhe. Face recognition based on adaptive weighted HOG. *Computer Engineering and Applications*, 2017
- 7) Olshausen B A, Field D J. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 1996
- 8) Ahonen, A. Hadid, and M. Pietikäinen. Face recognition with local binary patterns. In *European Conference on Computer Vision (ECCV)*, pages 469–481, 2000.
- 9) Ahonen, T., Hadid, A, and Pietikäinen, M. (2006), Face Description with Local Binary Patterns: Application to Face Recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence* 28(12):2037-2041
- 10) Viola, Paul, and Michael J. Jones. "Robust real-time face detection." *International journal of computer vision* 57.2 (2004): 137-154.
- 11) Varun Garg, Kritika Garg, Face Recognition Using Haar Cascade Classifier, *Journal of Emerging Technologies and Innovative Research (JETI)*, December 2016
- 12) Md. Abdur Rahim, Md. Najmul Hossain, T. Wahid, Md. Shafiul Azam, "Face recognition using Local Binary Patterns (LBP)", *Global Journal of Computer Science and Technology Graphics & Vision*, Global Journals Inc. (USA), Volume 13 Issue 4 Version 1.0 Year 2013
- 13) Tejas Rao C, Mohammed Zainuddin, Shrishail M Patil, Shashank G, Nimrita Koul, Real Time Person Detection and Classification using YOLO, 2019
- 14) Farhadi, J. Redmon, "YOLO9000: Better, Faster, Stronger," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- 15) Benenson, M. Omran, J. Hosang, B. Schiele, S. Zhang, "How Far Are We From Solving Pedestrian Detection?," *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1259-1267, 2016.
- 16) Zhang, P. Wang, Z. Hong, "Pedestrian Detection Based on YOLO-D Network," in *The IEEE 9th International Conference on Software Engineering and Service Science*, 2018