Project: Face Detection and Recognition

Course: ECE-GY 6143 Machine Learning

Professor: Pei Liu

Team Members	NetId
Divya Gupta	dg3483
Swathi Jain	sp6180

QUAD CHART

Topic Introduction:

Facial recognition is a technology that is capable of recognizing a person based on their face. It employs machine learning algorithms which find, capture, store and analyze facial features in order to match them with images of individuals in a pre-existing database. To recognize face is the easiest ways to distinguish the individual identity of each person. Mainly, human face recognition system has two phases- one is to detect face, and another is to recognize face. Detection of face means to identify if the images contain any face or not and recognition of face means to identify the name of the detecting face. For this, CNN (Convolutional Neural Network) plays a major role. There are a lot of neural network libraries like TFlearn and Keras. This report is mainly based on TensorFlow and TFlearn i.e., a model is created using these libraries. To create model, dataset (labelled data) is required which is generated using OpenCV and for visualization of output, another dataset (without labelling) is generated and this time, a model predicted the label (i.e., name) of the images of the user to be authenticated. The accuracy is found to be very high i.e., it almost predicts all images correctly.

Motivation:

The main motivation behind this project is using face detection for authenticating users. There might have been number of situation where it is necessary to recognize face or simply detect face. The traditional methods of lock/unlock are very inefficient. There may be possibility of losing keys or breaching of codes/passwords. So, we propose a face recognition system which can recognize face with good accuracy. The objectives of this system are as follows:

- Detect faces.
- Match detected faces to the images previously captured and recognize them.
- Provides accurate information about them (e.g. their names).

Why use Machine Learning to solve this problem?

Face recognition is the problem of recognizing and checking people in a photograph by their face. It is a task that is performed by humans, even in diverging light and when faces are changed by age or interfered with accessories and facial hair. Nonetheless, it is remained a challenging computer vision obstacle for decades until recently. Machine learning algorithms have become central to human efforts in resolving complex computational challenges. Deep learning systems can leverage very large datasets of faces and learn rich and condensed representations of faces, providing modern models to first perform as-well and later to outperform the face recognition abilities of humans.

Tools and Technologies:

Tools used

1. Camera integrated system

We can use either internal or external camera for this project. The internal camera can be our webcam that is already present in our system and the external camera can be joined externally to our system (e.g. Logitech webcam).

2. Google Colab

We used Google Colab as a web-IDE (integrated development environment) for this project. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education. It is very easy to use and flexible.

Libraries used

1. OpenCV

OpenCV is an open source computer vision and machine learning free software library. This library has more than 2500 optimized algorithms, which can be used to detect and recognize faces and to generate dataset.

2. TensorFlow:

TensorFlow is Google's Open Source Machine Learning Framework for dataflow programming across a range of tasks[1]. This library can be used with TFlearn to train the model and to make prediction.

3. TFlearn

TFlearn is a modular library in python that is built on top of core TensorFlow[2]. It is a deep learning library. We can use either Keras or TFlearn as a TensorFlow framework.

Contribution:

Among the many possible approaches, we have decided to use Haar-Like features algorithm for face detection part and neural network approach for face recognition part. Live captured images are converted to digital data for performing image-processing computations. These captured images are sent to face detection algorithm. Initially, a dataset is generated where human face of

authorized users is detected and cropped from a video. Then the picture is transformed from RGB to Grayscale because it is easy to detect faces in the grayscale. After that, the image manipulation used, in which the resizing, cropping, blurring and sharpening of the images done if needed. The next step is image segmentation, which is used for contour detection or segments the multiple objects in a single image so that the classifier can quickly detect the objects and faces in the picture. The next step is to use Haar-Like features algorithm for face detection.

The haar-like algorithm is also used for feature selection or feature extraction for an object in an image, with the help of edge detection, line detection, centre detection for detecting eyes, nose, mouth, etc. in the picture. It is used to select the essential features in an image and extract these features for face detection. The next step is to give the coordinates of x, y, w, h which makes a rectangle box in the picture to show the location of the face or we can say that to show the region of interest in the image. After this, it can make a rectangle box in the area of interest where it detects the face.

This report proposes a new face recognition method where local features are given as the input to the neural network. First, the face region is extracted from the image by applying various pre-processing activities. The method of locating the face region is known as face localization. The local features such as eyes and mouth are extracted from the face region. The distance between the eye balls and the distance between the mouth end points are calculated using the distance calculation algorithm. Then the distance values between the left eye and the left mouth end point, the right eye and the right mouth end point, the left eye and the right mouth end point, the right eye and the left mouth end point are calculated. These values are given as the inputs to the neural network. Back propagation algorithm is used for training the values. There is a hidden layer between input layer and the output layer which performs non-linear transformations of the input entered into the neural network. The result from the output layer is considered as the recognition result.

Accuracy and Loss:

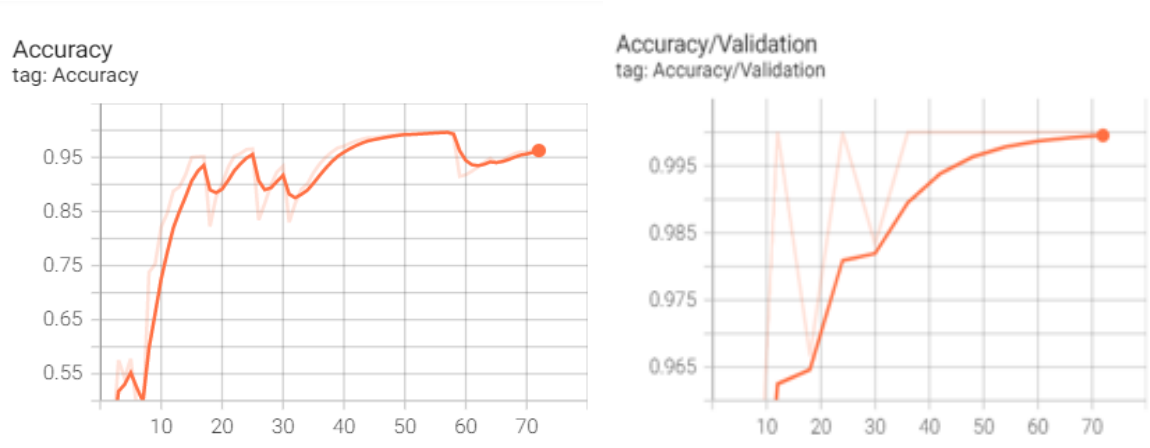

Figure 1: Training Accuracy and Test Accuracy

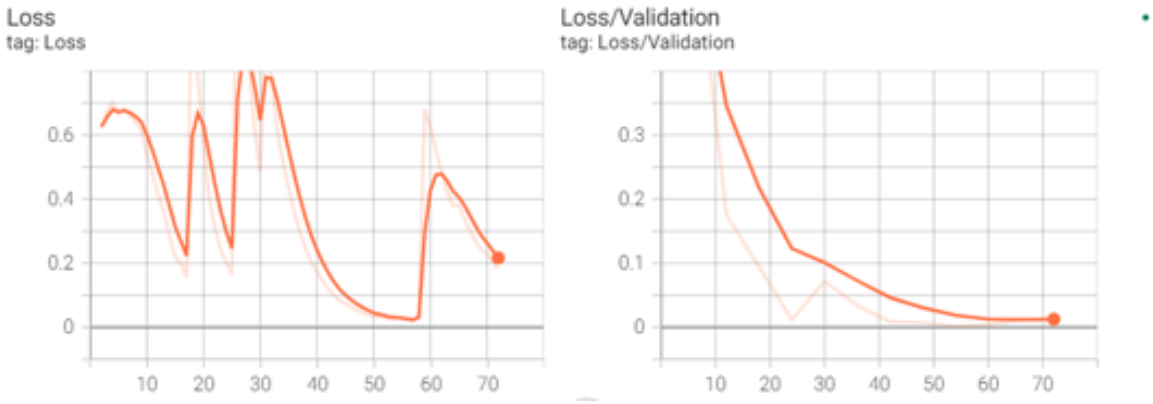

Figure 2: Training Loss and Test Loss

As you can see, with increase iterations, the loss has reduced to almost zero and we are getting almost accurate results. The dataset can be updated in time to maintain the accuracy. The accuracy is found to be very high i.e., it almost predicts all images correctly.

Future Work:

Still few faces are there which are really tough to distinguish, for e.g. twin siblings. Building a system that can distinguish between the two look-alikes can be a challenge. Moreover, we want our system to recognize someone, even though there is only one picture of that person on file, and it was taken at different angle, in different lighting, or they were wearing sunglasses as like human being. May be in near future, we can imagine our system to identify more challenging images.

Thus, this face recognition system can be preserved for future use. The dataset can be updated in time to maintain the accuracy. With the help of GUI, it doesn't take too much time to update the dataset.

REPORT

Introduction

Facial recognition is a technology that is capable of recognizing a person based on their face. It employs machine learning algorithms which find, capture, store and analyze facial features in order to match them with images of individuals in a pre-existing database. To recognize face is the easiest ways to distinguish the individual identity of each person. Mainly, human face recognition system has two phases- one is to detect face, and another is to recognize face. Detection of face means to identify if the images contain any face or not and recognition of face means to identify the name of the detecting face. For this, CNN (Convolutional Neural Network) plays a major role. There are a lot of neural network libraries like TFlearn and Keras. This report is mainly based on TensorFlow and TFlearn i.e., a model is created using these libraries. To create model, dataset (labelled data) is required which is generated using OpenCV and for visualization of output, another dataset (without labelling) is generated and this time, a model predicted the label (i.e., name) of that images. The accuracy is found to be very high i.e., it almost predicts all images correctly.

1.1. Background

Object detection is the process of finding real-world object instances like car, TV and humans in images or videos. It allows for the detection, recognition, and localization of object in an image. It is commonly used in applications such as image retrieval, security purpose system and even for the purpose of counting number of people in a crowd. Object detection can be done in various ways like Feature-based object detection and deep learning object detection [1].

Face Recognition is the task of identifying an already detected object as a known or unknown face. Two major libraries used for this project are TensorFlow and TFlearn. TensorFlow is Google's Open-Source Machine Learning Framework for dataflow programming across a range of tasks [1] and TFlearn is a modular library in python that is built on top of core TensorFlow [2]. The main aim of this project is to predict the images with their name. For any face recognition project, generating dataset is very important. So, for this, OpenCV library comes to an existence. OpenCV is an open-source computer vision and machine learning software library. The library has more than 2500 optimized algorithms, which can be used to detect and recognize faces and to generate dataset. To detect and recognize face may not be so much accurate than by using deep learning. So, in this project OpenCV is used only for generating dataset which is very much easy and efficient using this library.

Facial recognition systems are built on computer programs that analyze images of human faces for the purpose of identifying them. The programs take a facial image, measure characteristics such as distance between the eyes, the length of the nose, the angle of the jaw, and create a unique file and using that file, it then compares the image with another image and produces a score that measures how similar the images are to each other[3].

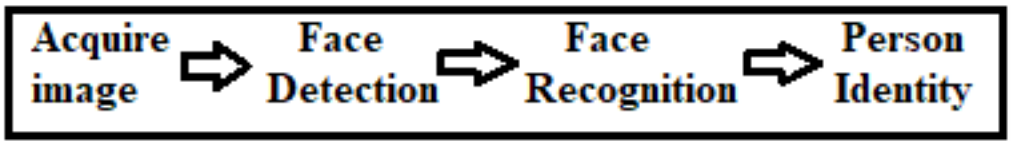

Figure 1.1: Steps of face recognition system applications

1.2. Problem Statement

There might have been number of situation where it is necessary to recognize face or simply detect face. The traditional methods of lock/unlock are very inefficient. There may be possibility of losing keys or breaching of codes/passwords. So, we propose a face recognition system which can be able to recognize face with maximum accuracy as possible.

1.3. Objectives

The objectives of this system are as follows:

- Detect faces.
- Match detected faces to the images previously captured and recognize them.
- Provides accurate information about them (e.g. their names).

1.4. Future scope

- Our current recognition system acquires images from file located on folder and from webcam. Scanner support can be implemented for greater flexibility.
- Currently, our system fails under the vastly varying conditions which we can solve in the future.

1.5. Limitation

Some of the limitations of this system are listed below:

- Face recognition system doesn't work well under poor lighting, sunglasses, long hair, low resolution images or another object partially covering the subject's face.
- This system can't tell the difference between identical twins.

Requirement analysis and Feasibility analysis

2.1. Literature review

The innate way for face recognition is to look at the main features of a captured face and compare them to features on other faces that have been stored in the database. The brief description of the related works and history has been presented in chronological order. The first attempt at face recognition began in the 1960's with a semi-automated system. At that time, marks were made on the photographs to locate the main features. It used features such as mouths, ears, noses and eyes. Then, the distances and ratios were computed from these marks to a common reference point and compared to reference data.

The fully automated face recognition was developed by Kenade in 1973 on a computer system involving geometric parameters extracting 16 facial features. The average successful identification matched accuracy above 50%. In 1986, L. Sirovich and M. Kirby[4] introduced Eigen faces in image processing based on Principal Component Analysis. The central idea was to reconstruct images from lower dimensions without any loss in information. With further advancement in technologies, implementing face recognition to identify criminal came into substance. In 1980's several techniques were improved measuring more enhanced subjective facial features and developed algorithms based on artificial neural networks. In 1991, Turk and Pentland[5] expanded upon the Eigen face approach by discovering how to detect faces within images. This led to the first instances of automatic face recognition. Their approach was constrained by technological and environmental factors, but it was a significant breakthrough in proving the feasibility of automatic facial recognition.

The Defense Advanced Research Projects Agency (DARPA) and the National Institute of Standards and Technology rolled out the Face Recognition Technology (FERET) program beginning in the 1990s in order to encourage the commercial face recognition market. The project involved creating a database of facial images. The database was updated in 2003 to include high-resolution 24-bit color versions of images. Included in the test set were 2,413 still facial images representing 856 people. The hope was that a large database of test images for facial recognition would be able to inspire innovation, that might result in more powerful facial recognition technology.

Beginning in 2010, Facebook began implementing facial recognition functionality that helped identify people whose faces may be featured in the photos that Facebook users update daily. While the feature was instantly controversial with the news media, sparking a slew of privacy-related articles, Facebook users at large did not seem to mind. Having no apparent negative impact on the website's usage or popularity, more than 350 million photos are uploaded and tagged using face recognition each day.

In 2012 the paper published in AJCSIT by Uttam Mande[6], binary clustering and classification techniques have been used to analyze the criminal data. The paper aims to potentially identify a

criminal based on the facial evidence obtained through the CCT cameras or the identification based on witness/clue at the crime spot using a Generalized Gaussian Mixture model.

Then in the report published in Image Processing on Line on 2014 about the Analysis of the Viola-Jones Face Detection Algorithm proposed a complete algorithmic description, a learning code and a learned face detector that can be applied to any color image. Since the Viola-Jones algorithm typically gives multiple detections, a post-processing step is also proposed to reduce detection redundancy using a robustness argument. Similarly, in the paper In March, 2018 a new technique to increase the speed of processing using haar feature-based cascade classifier was proposed by Indian students which was published in IJARCCE. Concepts of Viola-Jones framework and Open CV for detecting face in real time have been elaborated in it.

There are various techniques used in face detection. According to a paper on a "Review of Person Recognition Based on Face Model" by Shakir F. Kak & Firas Mahmood Mustafa & Pedro Valente published in September 1, 2018 [7] face recognition techniques are presented by details. Appearance-Based Model, Model-Based, Hybrid Based, Classification, Distance Measurements, Face Databases, Face Recognition

Face recognition has caught an increased interest from the scientific community in recent times because of its discrete nature. It holds several advantages over other biometric techniques especially for being natural, non-intrusive, and relatively easy to use in varied scenarios.

2.2. Requirement analysis

Requirement analysis is a process of precisely identifying, defining, and documenting the various requirements that are related to a particular business objective. Requirements gathering helps in clearly understanding the needs of the customer, defining the scope of the project, and resources required to complete it. The functional, non-functional, and technical requirements for this project are:

2.2.1. Functional requirements:

The functional requirement refers to "any requirement which specifies what the system should do"[8]. The functional requirements for this project are mentioned below:

- It should be able to handle 'png' and 'jpeg' images.
- It should generate the dataset properly.
- It should be able to predict the authorized users with high accuracy.

2.2.2. Non-functional requirements:

The non-functional requirement refers to “any requirement that specifies how the system performs a certain function”[8]. They are the characteristics or attributes of the system that can judge its operation. The non-functional requirements for this project are mentioned below:

- The GUI of the system will be user friendly.
- The system will be flexible to changes, e.g. an authorized user can be added at any time.
- Efficiency and effectiveness of the system will be made sure.

2.2.3. Technical requirements:

The technical requirements for this project are mentioned below:

a. Hardware Requirements:

- camera integrated system

b. Software Requirements:

- Windows operating system
- TensorFlow version 1.14
- OpenCV

2.3. Feasibility study

A feasibility study is an analysis of how successfully a project can be completed. It also serves as a solid foundation for developing our business plan. This study includes following three factors:

2.3.1. Technical feasibility

Technical feasibility defines the feasibility that is concerned with specifying equipment and software that will successfully satisfy the user requirement. For this project, the software and hardware requirement for the development of this system are already available as free as open source, so this project is technically feasible.

2.3.2. Economic feasibility

Economic feasibility means whether a business or a project is feasible cost wise or logistically. For this project, all the requirements are downloaded via internet and there are no any charges of any sorts associated with this system, so this project is also economically feasible.

2.3.3. Operational feasibility

Operational feasibility refers to the measure of solving problems with the help of new proposed system. In this project, the system solves the problem of losing keys or breaching of codes/passwords as mentioned in the problem statement, so this project is also operationally feasible.

2.4. Structuring System Requirements

2.4.1. Data modeling

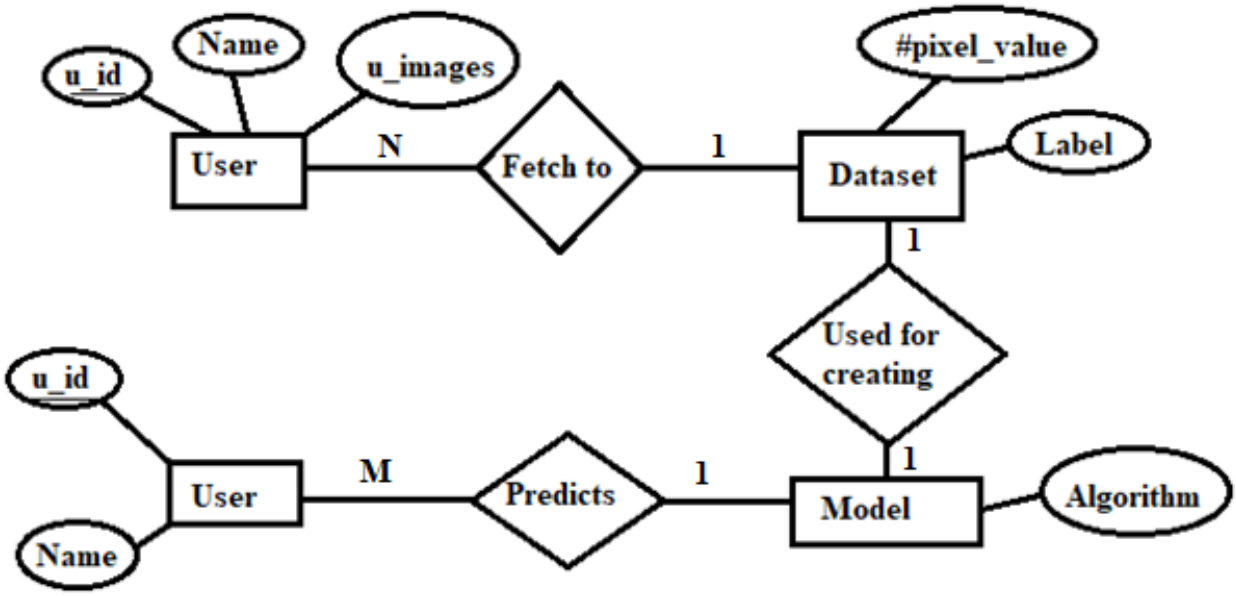

Figure 2.4.1: ER Diagram of Face Recognition System

2.4.2. Process modeling

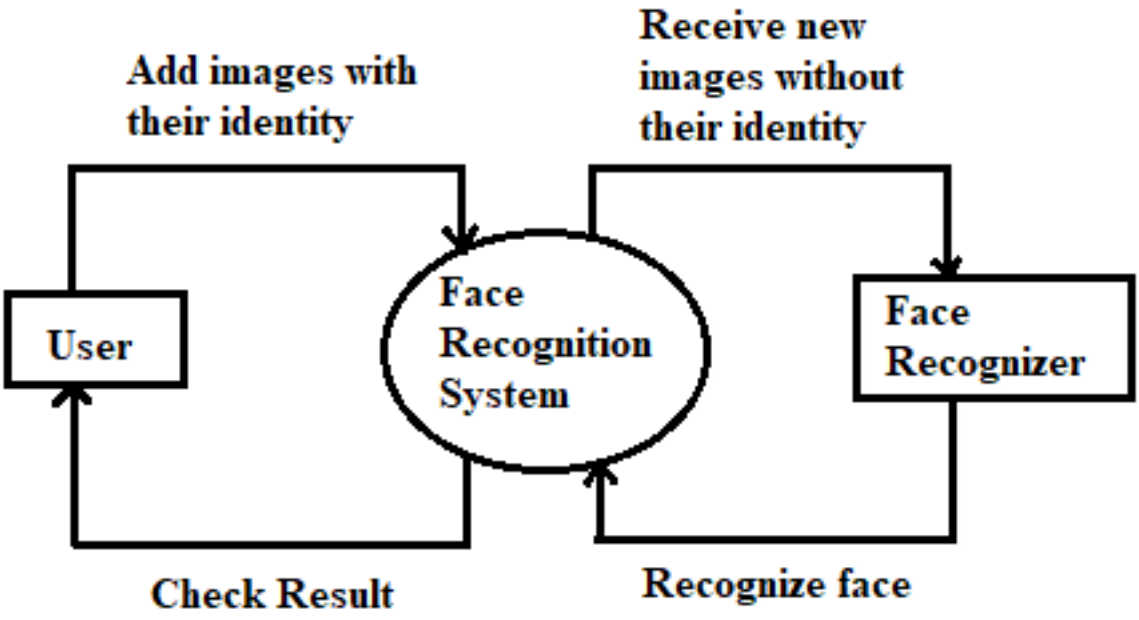

Figure 2.4.2(a): Data flow diagram (level 0)

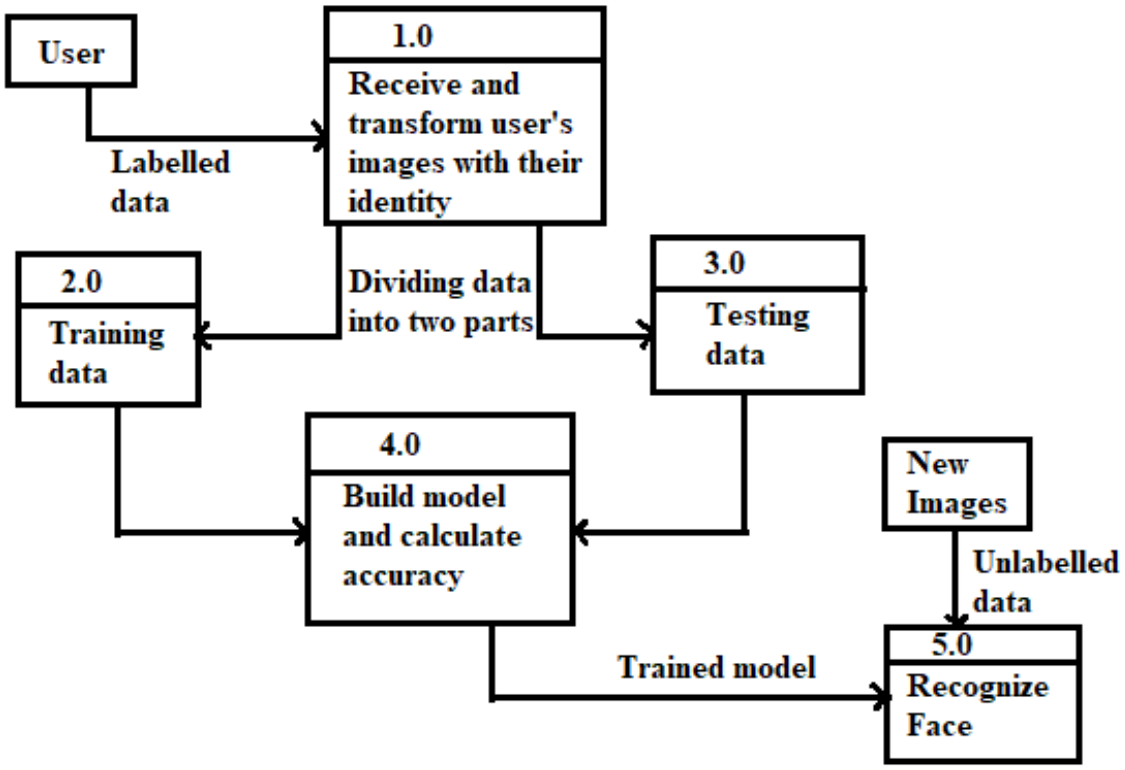

Figure 2.4.2(b): Data Flow Diagram (level 1)

Design of Face Recognition System

This system is mainly based on face detection and face recognition. Among the many possible approaches, we have decided to use Haar-Like features algorithm for face detection part and neural network approach for face recognition part.

When user opens GUI, there is button to capture images where the user captures images from camera. After image capturing, pre-processing of that image is done and then face detection and recognition is accomplished and result is shown to the user.

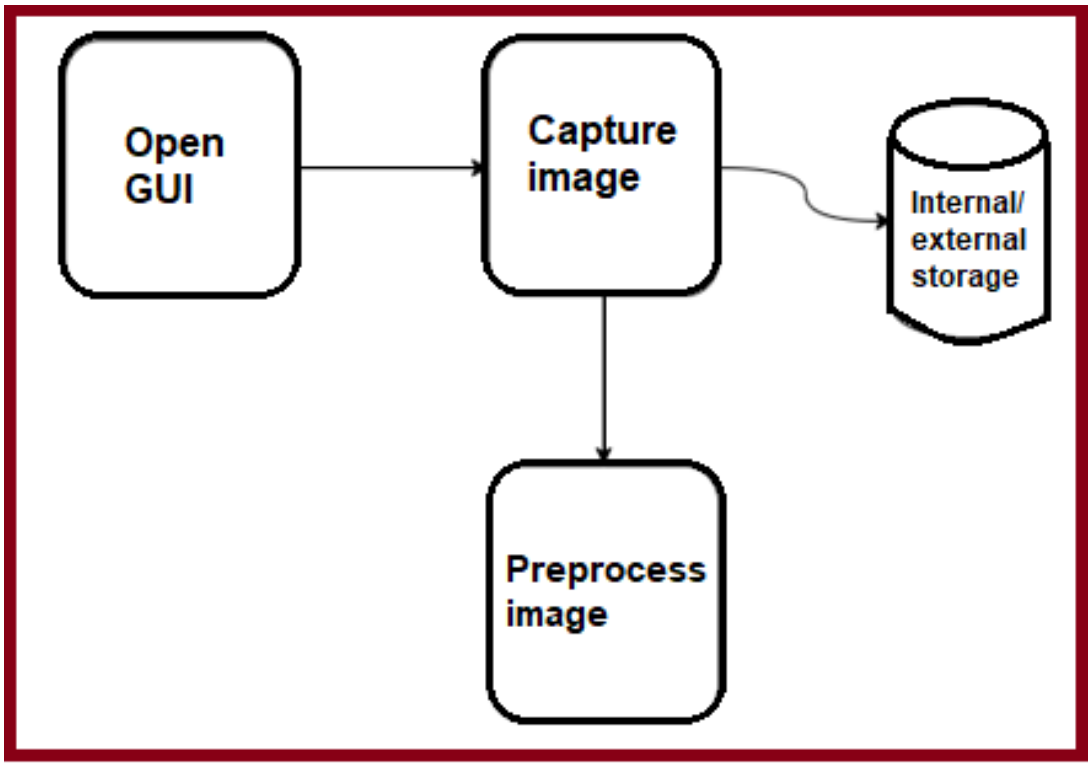

Figure 3: System architecture for Face Recognition

3.1. Face Recognition Approach

The face recognition system approach is illustrated in below figure 3.1:

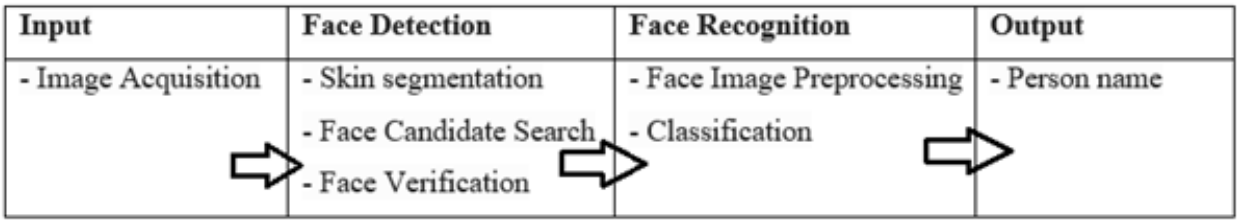

Figure 3.1: Face Recognition Approach

3.1.1 Input part

Input part is prerequisite for face recognition system. Image acquisition operation is performed in this part. Live captured images are converted to digital data for performing image-processing computations. These captured images are sent to face detection algorithm.

3.1.2 Face Detection Part

Face detection is the process of identifying human face in an image. The process of face detection is given below:

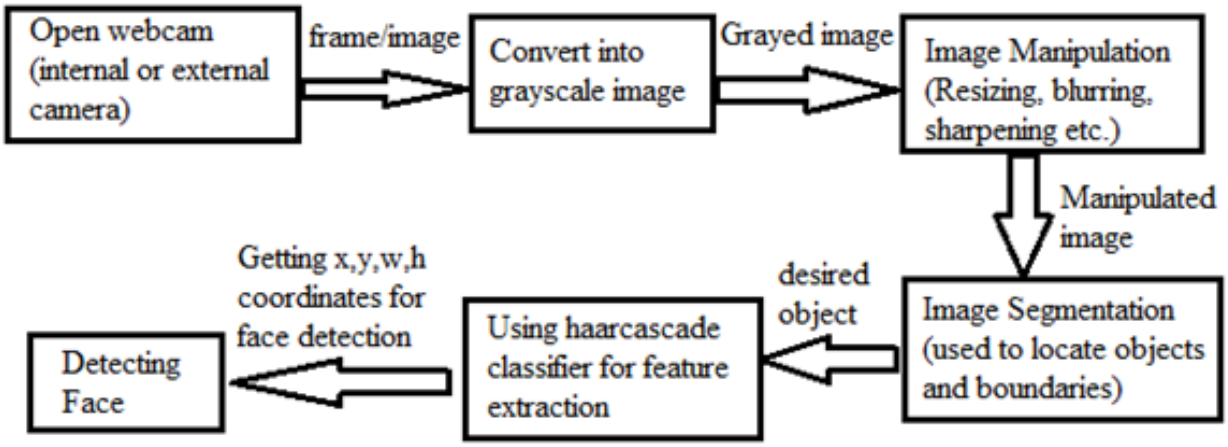

Figure 3.1.2: The framework of face detection process

Firstly, the image is imported by providing the location of the image. Then the picture is transformed from RGB to Grayscale because it is easy to detect faces in the grayscale. After that, the image manipulation used, in which the resizing, cropping, blurring and sharpening of the images done if needed. The next step is image segmentation, which is used for contour detection or segments the multiple objects in a single image so that the classifier can quickly detect the objects and faces in the picture. The next step is to use Haar-Like features algorithm, which is proposed by Voila and Jones for face detection. This algorithm used for finding the location of the human faces in a frame or image. All human faces share some universal properties of the human

face like the eye's region is darker than its neighbor pixels and nose region is brighter than eye region.

The haar-like algorithm is also used for feature selection or feature extraction for an object in an image, with the help of edge detection, line detection, centre detection for detecting eyes, nose, mouth, etc. in the picture. It is used to select the essential features in an image and extract these features for face detection. The next step is to give the coordinates of x, y, w, h which makes a rectangle box in the picture to show the location of the face or we can say that to show the region of interest in the image. After this, it can make a rectangle box in the area of interest where it detects the face[10].

3.1.3 Face Recognition Part

Face recognition is a process of recognizing name of detected face. The process of face recognition is shown below:

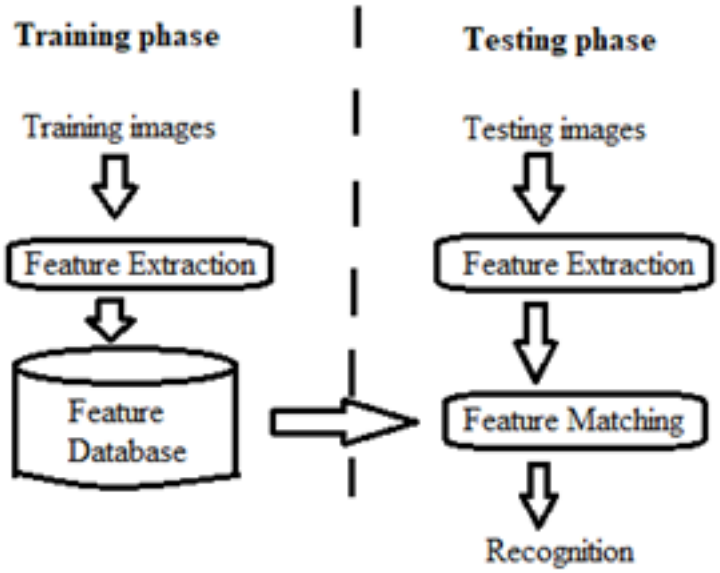

Figure 3.1.3(a): The framework of face recognition system

This report proposes a new face recognition method where local features are given as the input to the neural network. First, the face region is extracted from the image by applying various pre-processing activities. The method of locating the face region is known as face localization. The local features such as eyes and mouth are extracted from the face region. The distance between the eye balls and the distance between the mouth end points are calculated using the distance calculation algorithm. Then the distance values between the left eye and the left mouth end point, the right eye and the right mouth end point, the left eye and the right mouth end point, the right eye and the left mouth end point are calculated. These values are given as the inputs to the neural network. Back propagation algorithm is used for training the values. There is a hidden layer between input layer and the output layer which performs non-linear transformations of the input

entered into the neural network. The result from the output layer is considered as the recognition result.

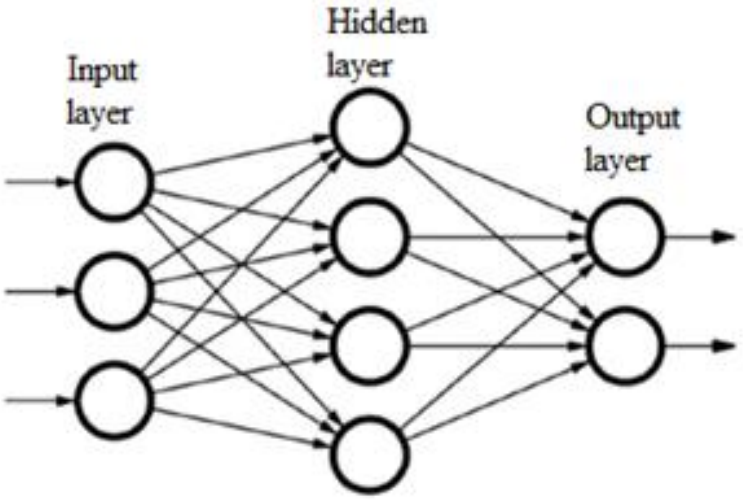

Figure 3.1.3(b): Multi-layer network structure

The activation functions help in deciding what is to be fired to the next neuron. Common activation functions are Linear, Sigmoid, SoftMax and ReLU function.

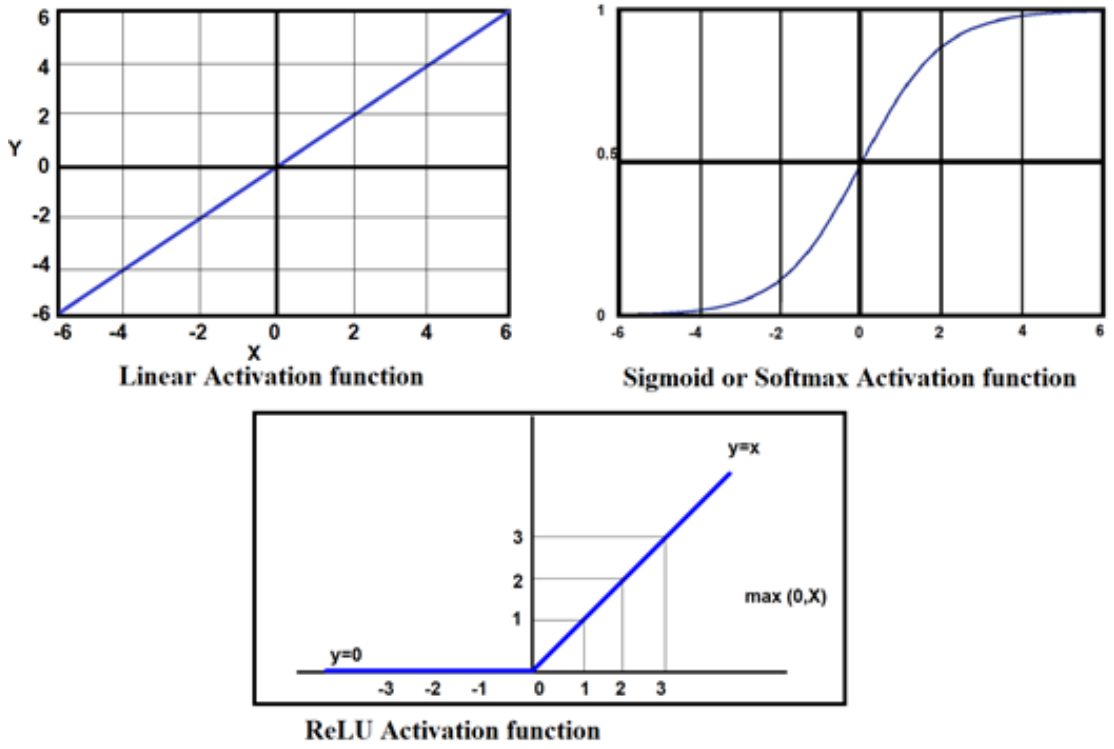

Figure 3.1.3(c): Activation functions used in neural network

3.1.4 Output part

This part is final step of face recognition system. Person name is determined with respect to output of face recognition. Output vector of neural network is used to identify person name[9].

Implementation and Testing

4.1. Implementation

The implementation phase involves putting the project plan into action. The code is written in Google colab as an integrated development environment (IDE). This emulator helped to implement the project in real-like environment.

4.1.1 Tools used

1. Camera integrated system

We can use either internal or external camera for this project. The internal camera can be our webcam that is already presented in our system and the external camera can be joined externally to our system (e.g. Logitech webcam).

2. Google Collab

We used Google Colab as a web-IDE (integrated development environment) for this project. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education. It is very easy to use and flexible.

4.1.2 Libraries used

1. OpenCV

OpenCV is an open source computer vision and machine learning free software library. This library has more than 2500 optimized algorithms, which can be used to detect and recognize faces and to generate dataset.

2. TensorFlow:

TensorFlow is Google's Open Source Machine Learning Framework for dataflow programming across a range of tasks[1]. This library can be used with TFlearn to train the model and to make prediction.

3. TFlearn

TFlearn is a modular library in python that is built on top of core TensorFlow[2]. It is a deep learning library. We can use either Keras or TFlearn as a TensorFlow framework.

4.1.3 Data Collection

Here, the data is collected in the form of Images. For data collection, we used the python openCV. It combines the best qualities of OpenCV, C++, API and Python language. OpenCV supports a lot of algorithms related to Computer Vision and Machine Learning. We used the haarcascade_frontalface_default.xml file for detecting the frontal faces and cropping it to the size we required and saved in the folder which is used later on to train the model.

Detection = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")

4.1.4 CNN Recognition Algorithm

CNNs are a type of feed-forward neural networks made up of many layers. CNNs consists of filters or kernels or neurons that have learnable weights or parameters and biases. Each filter takes some inputs, performs convolution and optionally follows it with a non-linearity. The structure of CNN contains Convolutional, Pooling, Rectified Linear Unit (ReLU), and Fully Connected Layers.

In our project, the algorithm is mainly carried out in three steps as below:

a. We have first resized the input image and then is passed to the different conv layers and the pooling layers.

b. We have used 5x5 filter and the number of filters used can differ from layer to layer of convolution.

c. Finally, we get flattened image which is used later on to recognize the face.

Conclusion and Future work

5.1. Conclusion:

Face recognition system recognize the face of authorized users very easily.

The main steps of this project are concluded below:

Step 1: To generate the dataset of authorized users

Step 2: Use that dataset to train the model

Step 3: Calculate the accuracy

Step 4: Use that trained model to predict detected faces

Step 5: Representing the project into GUI and also can be able to listen the name of authorized users.

All the above-mentioned steps are accomplished successfully. It met our initial aims and objectives and as mentioned in the limitation, we are working to deal with this too.

5.2. Future work:

Still few faces are there which are really tough to distinguish, for e.g. twin siblings. Building a system that can distinguish between the two look-alikes can be a challenge. Moreover, we want our system to recognize someone, even though there is only one picture of that person on file, and it was taken at different angle, in different lighting, or they were wearing sunglasses as like human being. May be in near future, we can imagine our system to identify more challenging images.

Thus, this face recognition system can be preserved for future use. The dataset can be updated in time to maintain the accuracy. With the help of GUI, it doesn't take too much time to update the dataset.

References:

[1] "Object Detection Tutorial using TensorFlow | Real-Time Object Detection | Edureka."

[2] "How TFLearn makes building TensorFlow models easier | Packt Hub." https://hub.packtpub.com/how-tflearn-makes-building-tensorflow-models-easier/

[3] X. Wang, "Face Identification," *Comput. Vis.*, pp. 279–285, 2014, doi: 10.1007/978-0-387-31439-6_354.

[4] M. Kirby and L. Sirovich, "Application of the Karhunen-Loéve Procedure for the Characterization of Human Faces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 1, pp. 103–108, 1990, doi: 10.1109/34.41390.

[5] S. Zhou, C. Chen, G. Han, and X. Hou, "Double additive margin softmax loss for face recognition," *Applied Sciences (Switzerland)*, vol. 10, no. 1. 2020, doi: 10.3390/app10010060.

[6] U. Mande, Y. Srinivas, and J. V. R. Murthy, “Criminal Identification System Based on Facial Recognition Using Generalized Gaussian Mixture Model.,” *Asian J. Comput. Sci. Inf. Technol.*, vol. 2, no. 6, 2012.

[7] S. F. Kak and F. M. Mustafa, “A Review of Person Recognition Based on Face Model,” *Eurasian J. Sci. Eng.*, vol. 4, no. 1, 2018, doi: 10.23918/eajse.v4i1sip157.

[8] “Functional vs Non-Functional Requirements.” https://reqtest.com/requirements-blog/functional-vs-non-functional-requirements/

[9] https://github.com/theAIGuysCode/colab-webcam

[10] D. Dwivedi, “Face Detection For Beginners.” https://towardsdatascience.com/face-detection-for-beginners-e58e8f21aad9