cognizancejournal.com
Grishma Poudel *et al*, Cognizance Journal of Multidisciplinary Studies, Vol.4, Issue.8, August 2024, pg. 67-85
(An Open Accessible, Multidisciplinary, Fully Refereed and Peer Reviewed Journal)
ISSN: 0976-7797
Impact Factor: 4.843
Index Copernicus Value (ICV) = 77.57

Voice Control Digital Assistant with Face Recognition

Grishma Poudel[1]; Dolraj Rijal[2]

[1]Nepal College of Information Technology, Kathmandu, Nepal
[2]York St John University, London, UK

grishpdl59@gmail.com; dolraj.rijal@yorksj.ac.uk

DOI: 10.47760/cognizance.2024.v04i08.004

Abstract: Digital assistants are an ongoing research topic with the potential to transform people's daily lives. Essentially, it is a system that has been designed to do specific tasks for the user. It lowers the need for additional people to complete a task that can be done digitally. The system prioritizes security by requiring authentication at the outset. It can recognize the user via facial recognition. The identified user can issue a specific voice command, which the system accepts as input. speech commands can be used to execute operations such as web queries, mail handling, reminder setting, simple arithmetic computations, and weather checking, with the result or output likewise provided in speech form.

Keywords: Digital Assistant, Face Recognition, Voice Command, Web queries, Mail handling, Weather Checking.

I. INTRODUCTION

Voice Control Digital Assistant with Face Recognition is combination of different software projects as a single integrated unit. Designed to address the need of reliable, confidential system that can perform the basic functions an individual need on regular basis like weather checking, mail handling, addressing simple queries through Google, Wikipedia and other information sources. Facial recognition feature enhances the security of the system. Access to other features is granted only to the user passing this facial recognition feature. Voice controlling feature of the system makes it more easily sophisticated and thus user friendly to operate. It reduces the human effort to give written commands and surf through search engines for simple Queries.

1.1 Background Introduction

This project is a prototype for a variety of uses. It can assist the user in doing simple activities such as checking the mail inbox, time, and weather conditions as well as complex tasks like computing arithmetic problems, searching Wikipedia and face recognition for security at home.

cognizancejournal.com
Grishma Poudel *et al*, Cognizance Journal of Multidisciplinary Studies, Vol.4, Issue.8, August 2024, pg. 67-85
(An Open Accessible, Multidisciplinary, Fully Refereed and Peer Reviewed Journal)
ISSN: 0976-7797
Impact Factor: 4.843
Index Copernicus Value (ICV) = 77.57

In contrast to Siri, Cortana and Google Assistant this project includes both face recognition and personal assistance. Any authorized person who have access to the device can access these functions. Voice Control Digital Assistant first recognizes the face and identify if the person is authorized or not. If the face of the person matches the facial database of the device then the person can access it. It is used as a single functioned machine so the system is dedicated to perform a single task. The objective of the project is to make a standalone personal assistant that can be interacted solely through the user's voice. The user calls the system by speaking a keyword through the microphone, to the system that he wants to accomplish. The user speaks the voice command and the system gives the output through the speakers attached to the system. The project's fundamental principles are speech-to-text conversion (to interpret user input) and text-to-speech conversion (to provide output to the user). The project includes aspects related to information, entertainment, and security. Using only voice instructions, the user may check his Gmail for unread messages, find out the current time, and solve basic arithmetic problems. In addition, the system can search Wikipedia for any phrase that the user enters and provide results via voice only. The user can also entertain himself by telling the system to play music from his playlist in both normal and shuffled manner for security, the system may take a photo using the webcam and recognize the face in the image. It further processes the face and tells whether the person is a friend or stranger. It achieves this by comparing the face to a collection of faces that have previously been saved and labeled as friends in the system. OpenCV and LBPH has been used for face detection and matching.

1.2 Motivation

We chose to work on this project as we wanted to make a system to support human productivity, to provide infotainment and provide security of the system. When we began working on it, it stroked us that this can be a great tool for visually impaired to connect normally with the World. After the voice controlled personal assistant part is finished, we think that the system is not secured and everybody can use the data of the system. So, the system is not secured and cannot used to store private data. So, in order to keep the private data secured we introduce face detection and recognition part so that the system is secured. As a result, interest grew as the project developed.

1.3 Problem Definition

It is not always possible to have human as personal assistant. So digital assistant can replace human assistant. Digital assistant is very cheap as compare to human assistant. In case of blind or disable person, digital assistant is very suitable.

1.4 Objectives

The main objectives of our project are as follows:

- To develop personal digital assistant.
- To search anything in internet through voice command and also obtain output in audio form as well as text form.

- To help blind as well as disable person to searching information without key typing.
- To avoid unwanted user by using face recognition which may control misuse of digital assistant.

1.5 Scope and Applications

The scope and application of voice controlled digital assistant with face recognition are as follows:

1. It helps to playing back simple information through voice command.
2. It can be used in office as receptionist.
3. It can be used to check mail, weather conditions etc.
4. It can be used as helper for blind person in study.

1.6 Limitation

The main limitations of our project are as follows:

- The voice recognition cannot work properly on background noise interference.
- Current face recognition still often misidentifies people which can sometimes lead to controversy.
- It does not work well in poor lighting, sunglasses, hats, beards, long hair, make-up etc.
- It is less effective when facial expression varies.

II. REQUIREDMENT ANALYSIS

Our project is software-based project and written in python code so different module are required to perform required task. Different software and required module are describe as follows:

2.1 Software Requirement

Python

Python has a dynamic type system, automated memory management, and supports a variety of programming paradigms, including object-oriented, imperative, functional, and procedural approaches. It boasts a vast and extensive standard library.

Python is extensively used, and interpreters for many operating systems are available, making it possible to run Python code on a broad range of platforms. The standard version of Python, CPython, is open source software with a community-based development process, as are virtually all of its variant implementations. The Python Software Foundation, a non-profit organization, manages CPython. [6]

OpenCV

We will require the OpenCV module in Python to identify and recognize faces. We utilize the OpenCV module version 3.2.0. OpenCV (Open Source Computer Vision Library) is a free and open-source library for computer vision and machine learning. The collection contains over 2500 optimized algorithms, including a diverse range of traditional and cutting-edge computer vision and machine learning techniques. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track

ISSN: 0976-7797
Impact Factor: 4.843
Index Copernicus Value (ICV) = 77.57

moving objects, extract 3D models of objects, generate 3D point clouds from stereo cameras, stitch images together to create a high-resolution image of an entire scene, and more. [7]

Google Speech to Text

Input is taking through voice in our project so Google Speech to Text module is required to covert voice input into text form for further process. Google Speech to Text module is only work when there is continuous internet access in PC.

Pocketsphinx

Pocketsphinx module is also use for convert audio signal into text form. It is offline based module which does not required internet connection. We are using pocket Sphinx library inside CMU Sphinx API. Pocket Sphinx is a library that depends on another library called SphinxBase which provides common functionality across all CMUSphinx projects. To install Pocketsphinx, we need to install both Pocketsphinx and SphinxBase. It's possible to use Pocket sphinx both in Linux, Windows, on MacOS, iPhone and Android. [1]

Beautiful Soup

Beautiful Soup is a Python package for extracting data from HTML and XML files. It integrates with our preferred parser to give idiomatic methods for traversing, finding, and altering the parse tree. It often saves programmers hours or days of effort. We may be searching for documentation for Beautiful Soup 3, but it is no longer being developed, and Beautiful Soup 4 is recommended for all new projects. [9]

2.2 Feasibility Study

- **Project Description**

  This project is Voice controlled digital assistant with face recognition .it is designed for the sole purpose of simplifying the otherwise tedious and hectic task cost effectively. This is done by designing a system that takes voice commands instead of manual typing.

- **Technical feasibility**

  Programming is done in python programming language which is easy to learn and apply and hardware requirements for executing are fulfilled by a normal PC.

- **Economic Feasibility**

  This project is implemented on PC but it can also be implemented on raspberry pi with external raspberry pi camera module in either case cost of implementation is reasonable.

- **Legal Feasibility**

  This project does not conflict any existing laws like data protection act or social media act. Parsing from Wikipedia is legal and permission of user is initially confirmed while logging in to email account.

cognizancejournal.com
Grishma Poudel *et al*, Cognizance Journal of Multidisciplinary Studies, Vol.4, Issue.8, August 2024, pg. 67-85
(An Open Accessible, Multidisciplinary, Fully Refereed and Peer Reviewed Journal)
ISSN: 0976-7797
Impact Factor: 4.843
Index Copernicus Value (ICV) = 77.57

- **Operational Feasibility**

Practical application of this system is reasonable. It is one-time investment. It may need timely maintenance but that will cost much less than human assistant. Given appropriate amount of maintenance it is operationally feasible.

III. METHODOLOGY

A system is realized by the combination of different sections. Proper coordination of different sections results in formation of perfect complete system. Voice Controlled Digital Assistant can be realized by combination of different parts which are as follows:

3.1 Face Detection

Face detection, the project's initial phase, was accomplished. It was done with OpenCV, and the algorithm is Haar Cascades. Paul Viola and Michael Jones introduced an effective object identification approach in their work "Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001. It uses Haar feature-based cascade classifiers. It is a machine learning-based strategy in which a cascade function is developed using a large number of positive and negative photos. It is then used to recognize items in other photos. To train the classifier, the method requires a large number of positive pictures (images of faces) as well as negative images. Then we have to extract features from it. For this, the haar characteristics illustrated in the graphic below are employed. They are exactly like our convolutional kernel. Each feature is a single value calculated by subtracting the total of pixels under the white rectangle from the sum of pixels under the black rectangle.

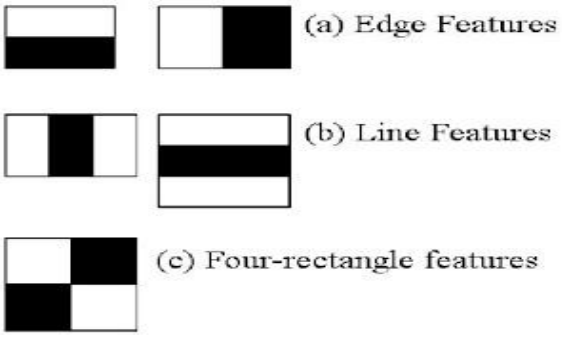

Fig 1: Haar features

Now, all conceivable kernel sizes and locations are employed to calculate a wide range of characteristics. (Can you image how much computing that requires? Even a 24x24 window returns more than 160000 features). For each feature computation, we must find the total of pixels beneath white and black rectangles. To address this, they introduced integrated pictures. It reduces the computation of the sum of pixels, regardless of how many pixels there are, to just four pixels. Nice, isn't it? It speeds things up significantly. However, the vast majority of the traits we computed are irrelevant. Consider the image below. The top row highlights two positive qualities. The first trait chosen appears to focus on the fact that the region of the eyes is frequently darker than the region

cognizancejournal.com
Grishma Poudel *et al*, Cognizance Journal of Multidisciplinary Studies, Vol.4, Issue.8, August 2024, pg. 67-85
(An Open Accessible, Multidisciplinary, Fully Refereed and Peer Reviewed Journal)
ISSN: 0976-7797
Impact Factor: 4.843
Index Copernicus Value (ICV) = 77.57

of the nose and cheekbones. The second picked trait is based on the fact that the eyes are darker than the nasal bridge. However, applying the same windows to the cheeks or anywhere else is irrelevant. So, how do we choose the finest features among 160000+ options? It is accomplished with Adaboost.

Fig 2: Implementation of Haar-features [2]

For this, we apply each and every characteristic to all training photos. It determines the appropriate threshold for each characteristic in order to classify faces as positive or negative. However, mistakes or misclassifications are unavoidable. We choose the features with the lowest error rate, which implies they accurately classify face and non-face photos. (The method is not as straightforward as this. Initially, each image is assigned equal weight. After each categorization, the weights of misclassified photos rise. Then the same procedure is repeated. New error rates are computed. There are also new weights. The method is repeated until the requisite accuracy or error rate is met (or the appropriate number of features are discovered). The final classifier is the weighted sum of several weak classifiers. It is referred to as weak since it cannot categorize a picture on its alone, but when combined with others, it becomes a powerful classifier. According to the report, even 200 characteristics may achieve 95% detection accuracy. Their final arrangement had roughly 6000 features. So now we're going to capture a photograph. Take each 24-by-24 window. Apply 6,000 characteristics to it. Check whether it is a face or not. Isn't this inefficient and time-consuming? Yes, it is. Authors have an excellent solution for this. The non-face area accounts for the majority of a picture. So it's a good idea to have a simple mechanism for determining whether or not a window is a face region. If not, discard it in one shot. Do not process it again. Instead, concentrate on the area where there may be a face. This gives us more time to investigate a probable facial area.
They introduced the notion of Classifier Cascade. Instead of applying all 6000 characteristics to a single window, split them into distinct stages of classifiers and apply them one at a time. Discard any windows that fail the first step. We do not evaluate the remaining features on it. If it passes, apply the second level of features and repeat the procedure. The window that goes through all stages is known as a face area.

Haar-cascade Detection in OpenCV

OpenCV includes both a trainer and a detector. We may use OpenCV to train our own classifiers for any item, such as cars and planes.

cognizancejournal.com
Grishma Poudel *et al*, Cognizance Journal of Multidisciplinary Studies, Vol.4, Issue.8, August 2024, pg. 67-85
(An Open Accessible, Multidisciplinary, Fully Refereed and Peer Reviewed Journal)
ISSN: 0976-7797
Impact Factor: 4.843
Index Copernicus Value (ICV) = 77.57

This section focuses on detection. OpenCV currently has a large number of pre-trained classifiers for faces, eyes, and smiles. These XML files are located in the opencv/data/haarcascades/ subdirectory.

Let's create face and eye detector with OpenCV.

- Load the XML classifiers first.
- Load the input image/video in grayscale mode.
- Then we find the faces in the image.
- Detected faces' locations are returned as Rect (x, y, w, h).
- After obtaining these locations, we can generate a ROI for the face and use eye detection.

3.2 Face Recognition

Face recognition, the second component of the project, was finished. It was done with the OpenCV (Open Source Computer Vision) library. The OpenCV version we are presently utilizing is 2.4.

OpenCV is a famous computer vision library founded by Intel in 1999. The cross-platform library focuses on real-time image processing and contains patent-free implementations of the most recent computer vision methods. Willow Garage took over support in 2008, and OpenCV 2.3.1 currently includes programming interfaces for C, C++, Python, and Android. OpenCV is distributed under the BSD license, thus it may be used in both academic and commercial projects.

OpenCV now supports the following algorithms:

- Eigen faces,
- Fisher faces, and
- Local Binary Patterns Histograms.

We implemented Local Binary Pattern Histogram (LBPH) in our project.

Local Binary Pattern Histogram

A human being's face provides a lot of information about his or her identity and emotions. Face recognition is a fascinating and demanding topic that has vital applications in many domains, including law enforcement identification, authentication for banking and security system access, and personal identity, among others. In our project, we employ the Local Binary Patterns Histograms (LBPH) technique to detect faces. LBPH work is divided into three parts: face representation, feature extraction, and classification. Face representation defines how to model a face and the subsequent detection and recognition methods. The feature extraction step extracts the most valuable and distinctive characteristics from the facial picture. During categorization, the facial image is compared to photographs from the database. The facial area is initially broken into discrete sections from which Local Binary Patterns (LBP) and histograms are retrieved and combined to form a single feature vector. This feature vector represents the face efficiently and is used to measure picture similarities. Face representation is the initial job, which involves modeling a face.

cognizancejournal.com
Grishma Poudel *et al*, Cognizance Journal of Multidisciplinary Studies, Vol.4, Issue.8, August 2024, pg. 67-85
(An Open Accessible, Multidisciplinary, Fully Refereed and Peer Reviewed Journal)
ISSN: 0976-7797
Impact Factor: 4.843
Index Copernicus Value (ICV) = 77.57

The methods used to depict a face determine the subsequent detection and identification algorithms. For entry-level recognition (determining whether or not the provided image represents a face), the image is altered (scaled and rotated) such that it has the same 'position' as the photos in the database. During the feature extraction step, the most useful and distinct characteristics (properties) of the facial picture are extracted. The collected characteristics are used to compare the facial image to the photographs in the database. This is done during the categorization process. The classification component returns the identification of a face picture from the database with the greatest matching score, indicating the least variations from the input face image. A threshold value can also be used to evaluate whether the differences are sufficiently minimal. After all, it is possible that a certain face is not in the database at all.

These characteristics are binary patterns that characterize the surrounds of pixels in areas. The gathered features from the areas are concatenated into a single feature histogram, which represents the picture. Images may be compared by calculating the similarity (distance) between their histograms. According to various research [2, 3, 4], face recognition employing the LBP approach produces excellent results in terms of speed and discrimination performance. Because of how the texture and form of images are defined, the approach appears to be extremely resistant against face photographs with various facial expressions, lighting circumstances, image rotation, and human aging. Ojala et al. introduced the initial LBP operator.

This operator operates with a pixel's eight neighbors, using the Centre pixel's value as a threshold. If a neighbor pixel has a greater grey value than the Centre pixel (or the same grey value), it is allocated a one, otherwise a zero. The Original LBP Operator Later, the LBP operator was extended to include neighborhoods of varying sizes. In this scenario, a circle with radius R is drawn from the center pixel. P sample points are obtained along the periphery of this circle and compared to the value of the center pixel. To obtain the values of all sampling points in the vicinity for any radius and number of pixels, (bilinear) interpolation is required. For neighborhoods, the notation (P, R) is used.

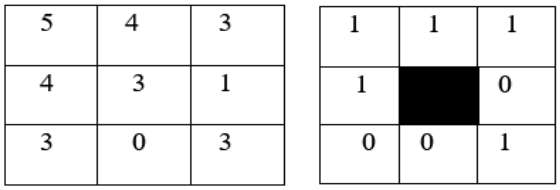

Fig 3: LBPH Module

Circularly neighboring sets for three distinct P and R values. If the coordinates of the center pixel are (xc, yc), the coordinates of his P neighbors (xp, yp) on the edge of the circle of radius R may be computed using the sine and cosine functions:

$Xp = xc + R\cos(p/P)$ ……. (1)
$Yp = yc + R\sin(p/P)$ …….. (2)

Once the Local Binary Pattern for each pixel has been determined, the image's feature vector may be created. To get an effective representation of the face, the picture is first segmented into K2 areas. For each region, a histogram with all potential labels is created. This implies that each bin in a histogram reflects a pattern and the

cognizancejournal.com
Grishma Poudel *et al*, Cognizance Journal of Multidisciplinary Studies, Vol.4, Issue.8, August 2024, pg. 67-85
(An Open Accessible, Multidisciplinary, Fully Refereed and Peer Reviewed Journal)
ISSN: 0976-7797
Impact Factor: 4.843
Index Copernicus Value (ICV) = 77.57

number of times it appears in the region. The feature vector is then created by concatenating the regional histograms into a single huge histogram.

To implement facial recognition in this project, we offered the Local Binary Patterns approach. Local Binary Pattern works on local features and employs the LBP operator to summarize the local special structure of a face picture. LBP is described as an ordered series of binary comparisons of pixel intensities between the center pixels and the eight pixels around them. Local Binary Pattern do this comparison by applying the following formula:

LBP (Xc, yc) = ∑7n=0S (in-ic) 2n

Where ic corresponds to the value of the center pixel (Xc, Yc), in to the value of eight surrounding pixels. It is used to determine the local features in the face and also works by using basic LBP operator. Feature extracted matrix originally of size 3 x 3, the values are compared by the value of the center pixel, then binary pattern code is produced and also LBP code is obtained by converting the binary code into decimal one. [3]

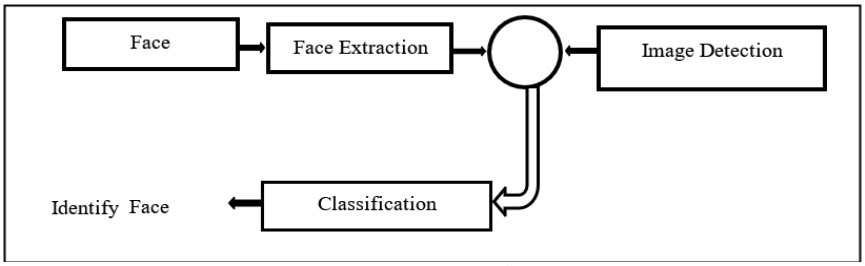

Fig 4: Block Diagram of LPBH

The Face Recognition Algorithm

Input: Training Image set.

Output: Feature extracted from face image and compared with center pixel and recognition with unknown face image.

- Initialize temp = 0
- FOR each image I in the training image set
- Initialize the pattern histogram, H = 0
- FOR each center pixel tc € I 5. Compute the pattern label of tc, LBP
- Increase the corresponding bin by 1.
- END FOR
- Find the highest LBP feature for each face image and combined into single vector
- Compare with test face image.
- If it matches it most similar face in database then successfully recognized.

cognizancejournal.com
Grishma Poudel *et al*, Cognizance Journal of Multidisciplinary Studies, Vol.4, Issue.8, August 2024, pg. 67-85
(An Open Accessible, Multidisciplinary, Fully Refereed and Peer Reviewed Journal)
ISSN: 0976-7797
Impact Factor: 4.843
Index Copernicus Value (ICV) = 77.57

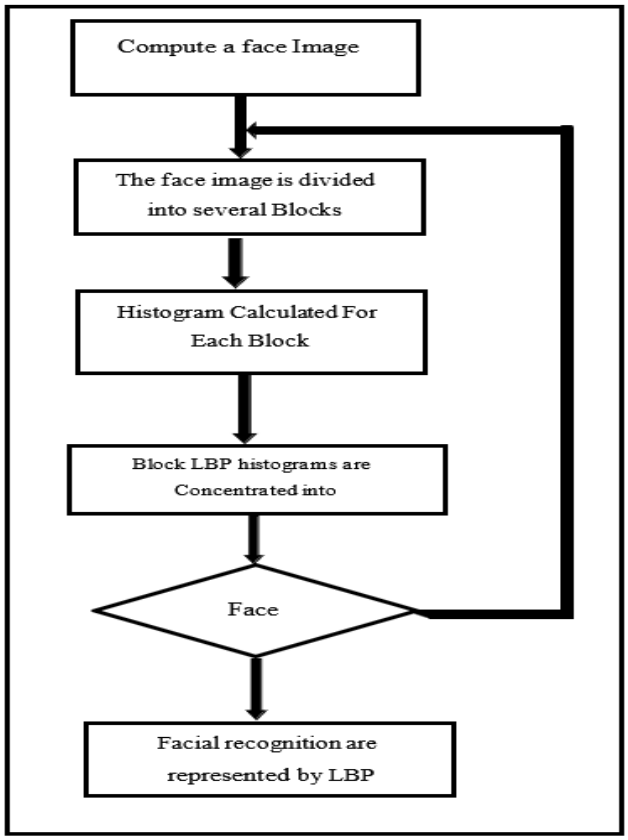

Fig 5: Face Recognition Algorithm

3.3 Speech Recognition and Conversion

The terms "voice recognition" and "speech recognition" are used interchangeably to describe software/hardware systems capable of transliterating spoken words into digital text and utilizing voice commands to control computer activities. Speech recognition is a complicated and dynamic categorization problem. It focuses on digitizing the human voice and translating it into a computer-readable format. It is the process of obtaining a written transcription or some meaning from oral input. Speech recognition is far more difficult than synthesis. Speech recognition is the technique of automatically extracting and identifying linguistic information supplied by a speech wave via computers or electrical circuits. The phrases "voice recognition" and "speech recognition" are used interchangeably to describe software/hardware systems that can transliterate spoken words into digital text and employ voice commands to control computer operations. Speech recognition is a complex, dynamic classification issue. It focuses on digitizing and converting human voices into computer-readable formats. It is the process of producing a written transcription or meaning from spoken information. Speech recognition is far more difficult than synthesis.

Speech recognition is the process of automatically extracting and recognizing linguistic information contained in a speech wave using computers or electrical circuits.

ISSN: 0976-7797
Impact Factor: 4.843
Index Copernicus Value (ICV) = 77.57

It simply understands that they are words, and the technology used to do the task is voice recognition. Automatic voice Recognition (ASR) is the technique by which a computer translates an acoustic voice stream to text. ASR provides a solution called directed speech-to-text, which converts speech into text. This method of speech recognition is carried out using ASR technology. [12]. The goal of voice recognition is to identify the message being said. It is natural and intuitive. Speech recognition enables users to provide input to an application using their voice. Speech recognition allows us to offer input to an application by talking, just like clicking with a mouse, typing with a keyboard, or pressing a key on a phone keypad does. In the desktop environment, we'll need a microphone to perform this. Speech may also be synthesized from plain text in a technique known as text-to-speech. This allows voice-processing software to read from textual databases. The quality of synthetic speech is judged by its intelligibility and naturalness. Fluency of verbal output refers to the capacity to create messages with a variety of languages, emphasis, intonation, and speed. The complexity of the synthesis hardware is evaluated in terms of both storage and compute. An ideal synthesis system generates comprehensible and natural speech with fluency at a cheap cost. Currently, there is no such practical system with performance comparable to the ideal system. Technically, speech synthesis is a straightforward concatenation system that keeps a vocabulary of prerecorded and digitally coded words and phrases. The user's action sends a request to the concatenation device for a certain sequence of words and phrases. The coded versions of the lexical items are retrieved from storage. The vocabulary pieces for the message are concatenated. The final findings are given to the decoder, which reproduces the analogue speech. In a complete Text to Speech (TTS) system, the message is any ASCII string. The text string is translated into a series of phonetic symbols accompanied by prosody indicators that indicate speech speed, intonation, and word stress.

Text-to-sound/prosody conversion entails linguistic analysis, including the use of dictionaries to search up word pronunciation rules for exceptions and exceptional circumstances. Converting speech waveforms or letters into sounds Message Text (ASCII Alphabetical characters Assemble units and calculate prosody. Continuous parameters. Phonetic symbols and prosody markers Synthesizer Store for sound units. Dictionary and Rules Text-to-Speech Synthesis System We discuss an algorithm for creating word length, voice, pitch, and loudness contours. Once the phonetic symbols and prosody markers have been discovered, speech units are built. Speech pitch and duration contours are calculated by storing fundamental sound. Text-to-speech is a technology that automatically generates new phrases or words. This is a way for making a computer talk.3.1.4 Text to Speech Conversion

Likewise, we are giving voice command as input, it is appropriate to use voice as output for Voice Controlled Intelligent Assistant. So, for speech to text conversion different API is used. So in this project we are using eSpeak API as speech to text conversion engine.

eSpeak

eSpeak is an open-source software voice synthesizer for English and other languages, available for Linux and Windows.

cognizancejournal.com
Grishma Poudel *et al*, Cognizance Journal of Multidisciplinary Studies, Vol.4, Issue.8, August 2024, pg. 67-85
(An Open Accessible, Multidisciplinary, Fully Refereed and Peer Reviewed Journal)
ISSN: 0976-7797
Impact Factor: 4.843
Index Copernicus Value (ICV) = 77.57

eSpeak employs a "formant synthesis" approach. This enables numerous languages to be delivered in a short package. The voice is crisp and fast-paced, but it lacks the naturalness and smoothness of bigger synthesizers based on genuine speech recordings. [13]

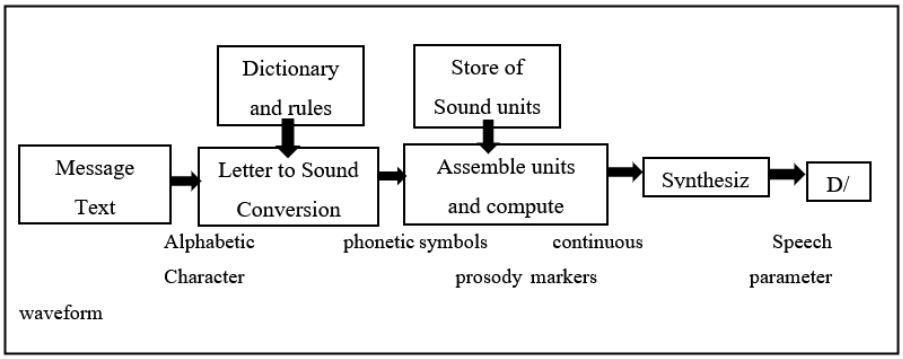

Fig 6: Block Diagram of Text to Speech Conversion

The eSpeak speech synthesizer supports several languages, however in many cases these are initial drafts and need more work to improve them. [4]

Speech to Text Conversion

For this project, speech is employed as a command input. So, this voice must be converted to text and analyzed so that the Voice Controlled Intelligent Assistant can choose which job to perform. We intend to create this conversion application using Python programming language. Python has several libraries for converting voice to text. There are both cloud-based and offline conversion engines. Speech recognition library that supports several engines and APIs, both online and offline. Speech recognition engine/API support:

- CMU Sphinx (works offline)
- Google Speech Recognition
- Google Cloud Speech API
- Wit.ai
- Microsoft Bing Voice Recognition
- Houndify API

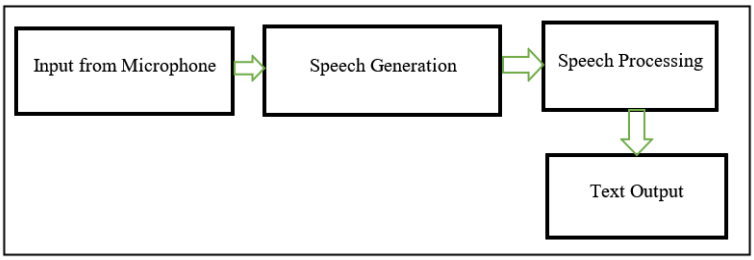

Fig 7: Block Diagram of Speech to text conversion

ISSN: 0976-7797
Impact Factor: 4.843
Index Copernicus Value (ICV) = 77.57

For this project, we'd want to work on CMU Sphinx, which runs offline. This is a regularly used API that is incredibly adaptable, allowing us to tailor it to meet our needs. CMUSphinx is a premier voice recognition toolkit that includes a variety of tools for developing speech applications. The CMU Sphinx toolbox includes a variety of packages for various tasks and applications. Making a decision might be difficult at times.

To clean up, here is the list:

- Pocket sphinx — lightweight recognizer library written in C.
- Sphinx base — support library required by Pocket sphinx
- Sphinx4 — adjustable, modifiable recognizer written in Java
- Sphinx train — acoustic model training tools

Pocket sphinx

We're utilizing the pocket sphinx library with the CMU Sphinx API. Pocket Sphinx is a library that relies on another library named Sphinx Base, which offers common functionality to all CMUSphinx applications. To install Pocket Sphinx, we must also install the Sphinx foundation. Pocket sphinx is compatible with Linux, Windows, MacOS, iPhone, and Android.

Google API

The Google Cloud voice API allows developers to easily integrate Google voice recognition technology into their applications. The Speech API enables you to transmit audio and get text transcriptions from the service. The Google Speech API makes use of the gcloud command-line tool, which is included with the Google Cloud Platform Cloud SDK.

Web scraping

In this project, we may issue commands to our Assistant, such as browsing the internet for information. It should figure out what information to retrieve. The information might be about some individuals, the weather, or some questions. It then searches for this information and provides a summary of it. As a result, it must sift this data and choose just the relevant stuff. In this instance, we need to use Web Scrapping and Parsing. We are using Beautiful Soup, a Python module for extracting data from HTML and XML files.

3.4 System Block Diagram

In our system, first of all the system open the camera which is used to detect the face of the person. If the face is detected the system recognize whether the face is authorized person or not. If that is not of the authorized person then the system exits and the process goes to the initial state. If the face of the person is recognized then the system says “please say something I will try to understand” and user gives the voice command. Voice command may be like check my mail, send my mail, what is weather, any simple mathematical calculation, etc. The output

cognizancejournal.com
Grishma Poudel *et al*, Cognizance Journal of Multidisciplinary Studies, Vol.4, Issue.8, August 2024, pg. 67-85
(An Open Accessible, Multidisciplinary, Fully Refereed and Peer Reviewed Journal)
ISSN: 0976-7797
Impact Factor: 4.843
Index Copernicus Value (ICV) = 77.57

of our system is in audio as well as text form. System wait just for 5 second to take voice input, if it does not detect any audio signal It say "please say something I will try to understand" and again we give voice command.

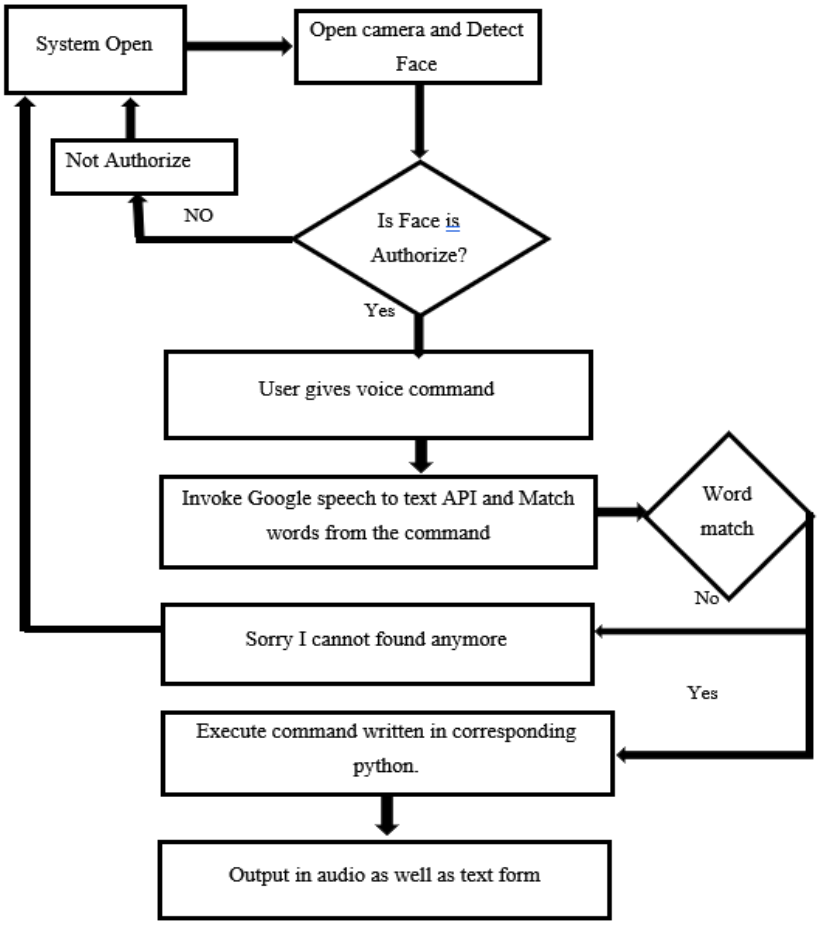

Fig 8 System Block Diagram

IV. IMPLEMENTATION DETAILS

This project was completed implementing Python as programming language. This project includes some functions which is interpreted by Python 2.7.11 and some are interpreted by Python 3.5.11. Speech recognition, Web Scraping, Sending Mail took Python 3.5.11 as interpreter whereas Face Recognition and Mail Reading function took Python 2.7.11 as interpreter. With comparison with other programming language, Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently whereas the other languages use punctuations. It has fewer syntactical constructions than other languages. [5]

cognizancejournal.com
Grishma Poudel *et al*, Cognizance Journal of Multidisciplinary Studies, Vol.4, Issue.8, August 2024, pg. 67-85
(An Open Accessible, Multidisciplinary, Fully Refereed and Peer Reviewed Journal)
ISSN: 0976-7797
Impact Factor: 4.843
Index Copernicus Value (ICV) = 77.57

PyCharm

PyCharm is the programming environment used to build this project. PyCharm is an Integrated Development Environment (IDE) for computer programming, especially Python. It was created by the Czech firm JetBrains. [16] It includes code analysis, a graphical debugger, an integrated unit tester, version control system (VCS) integration, and Django web development support.

PyCharm is available for Windows, macOS, and Linux. The Community Edition is published under the Apache License [17], while the Professional Edition is released under a proprietary license, which includes additional functionality.

Python Packages

The Python packages used in this project is listed below:

Pip is a package management system used to install and manage software packages written in Python. Many packages can be found in the Python Package Index (PyPI) [6]. Python 2.7.9 and later, and Python 3.4 and later include pip (pip3 for Python 3) by default. 'pip' is a recursive acronym that can stand for either "Pip Installs Packages" or "Pip Installs Python"

OpenCV

OpenCV is a famous computer vision library founded by Intel in 1999. The cross-platform library focuses on real-time image processing and contains patent-free implementations of the latest computer vision methods.The Opencv library is utilized in this project's permission portion, which utilizes face recognition. This library's functions, such as VideoCapture(), allow the computer to access a webcam and receive video input, resulting in a high number of picture frames.Training it to detect Haar-like characteristics. OpenCV is used to recognize faces in the input. The opencv_contribute library includes numerous classes. One of them is implemented here. For face recognition, the LBPH, Fisherface, and Eigen face methods are available, with the Local Binary Pattern Histogram being employed. [6].

Speech Recognition

We implemented speech recognition for taking command as input. When the face recognition system finds out the requesting person as user then the user will be able to access speech recognition system and give voice commands. We have used two speech recognition engines as primary recognition engine and secondary recognition engine. The primary recognition engine is Google speech-to-text API. It is online API which receives audio as input which Google processes, converts and sends the text data via web. Google speech recognition is a paid service. In contrast with other speech recognition system, we found Google's offering to be of highest quality and we have 50 calls per day free with our developer API key.

For secondary speech recognition engine we used PocketSphinx package which is based on CMUSphinx. It is offline speech recognition engine. So whenever the internet connection is fast and reliable the speech-to-text

conversion takes place via Google speech recognition. But when the internet connection is interrupted or the data transfer is very low, it automatically switches to PocketSphinx. Sphinx voice recognition was very poor as compared to Google speech recognition though still tended to produce useful recognized words in the middle and end of the phase.

PyAudio

PyAudio offers Python bindings for PortAudio, a cross-platform audio I/O library. PyAudio allows you to simply use Python to play and record audio on a range of platforms. PyAudio was inspired by pyPortAudio/fastaudio, which provides Python bindings for the PortAudio v18 API.Snack: A cross-platform sound toolkit for Tcl/Tk and Python.

In this project, we utilized PyAudio to capture audio from a microphone and play it over a speaker. PyAudio is compatible with a variety of systems, including GNU/Linux, Microsoft Windows, and Apple Mac OS X / macOS.

Google 1.9.3

Google 1.9.3 is a Python tool that allows you to search Google from within Python. [19] We combined this program with speech recognition so that the user could search Google using voice commands. The search queries are captured in speech form and then converted to text before being applied to Google's search module. The search results are listed, and web scraping is used to retrieve them, including Wikipedia.

Web Scraping

Web scraping (also known as screen scraping, web data extraction, online harvesting, and so on) is a technique for extracting huge volumes of data from websites, which is then saved to a local file on a computer or to a database in spreadsheet format. [20] Most websites show data that can only be seen using a web browser. They do not provide the option to save a copy of this information for personal use. The only alternative is to manually copy and paste the data, which might take several hours or even days to accomplish. Web Scraping is a way for automating this process, so that instead of manually copying data from websites, the Web Scraping program may complete the same operation in a fraction of time.

We used BeautifulSoup 4 to scrape the web. When the system accesses a Wikipedia page, it scrapes it using BeautifulSoup, displays the first paragraph of content in the console, and generates speech output using the eSpeak API.

SMTP

We utilized SMTP as a module to send messages through Gmail. The smtplib module creates an SMTP client session object that may be used to send email to any Internet system running an SMTP or ESMTP listener daemon. It includes methods that support the complete range of SMTP and ESMTP activities. [21]

cognizancejournal.com
Grishma Poudel *et al*, Cognizance Journal of Multidisciplinary Studies, Vol.4, Issue.8, August 2024, pg. 67-85
(An Open Accessible, Multidisciplinary, Fully Refereed and Peer Reviewed Journal)
ISSN: 0976-7797
Impact Factor: 4.843
Index Copernicus Value (ICV) = 77.57

IMAP

This module defines three classes: IMAP4, IMAP4_SSL, and IMAP4_stream, which represent a connection to an IMAP4 server and implement a major portion of the IMAP4rev1 client protocol as described in RFC 2060.

The imaplib module provides three classes, with IMAP4 as the basic class:

Class imaplib. IMAP4 ([host [, port]].

This class implements the IMAP4 protocol. When the instance is first started, a connection is established and the protocol version (IMAP4 or IMAP4rev1) is determined. If no host is supplied, the local host ('') is used. If port is not specified, the normal IMAP4 port (143) is used.

V. RESULT AND ANALYSIS

The system has been tested on different conditions and environment the output were noted. Mainly face detection and recognition parts were very sensitive towards light and color. Face detection and recognition has been texted on different intensity of light. We found out that the efficiency is higher if testing environment is maintained similar to trained condition.

The result of face recognition is below:

Fig 9: Results of face recognition

When face is detected, it checks that face with trained data sets and finds out whose face is that. It draws rectangle in face and writes name in face. The result of Speech to text conversion was good when accessed through google speech recognition system. But when speech to text conversion done by pocketsphinx due to interruption in internet connection, the result was poor. So the result was better in good connectivity state. The background noise plays vital role in interrupting the result. So the background should be as silent as possible.

cognizancejournal.com
Grishma Poudel *et al*, Cognizance Journal of Multidisciplinary Studies, Vol.4, Issue.8, August 2024, pg. 67-85
(An Open Accessible, Multidisciplinary, Fully Refereed and Peer Reviewed Journal)
ISSN: 0976-7797
Impact Factor: 4.843
Index Copernicus Value (ICV) = 77.57

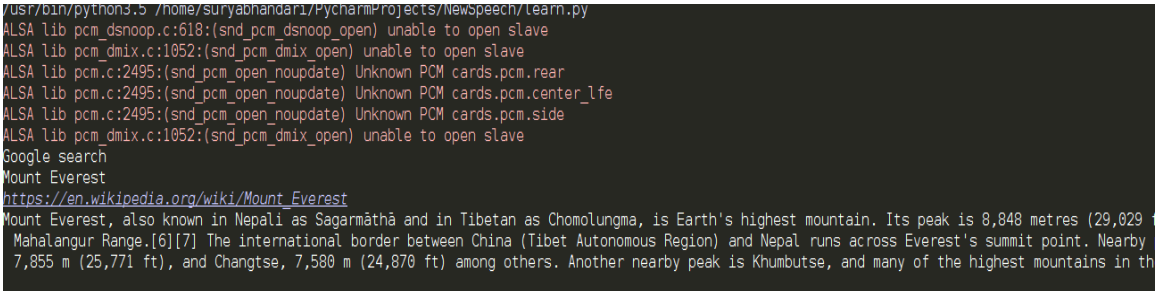

Fig 10: Results of speech to text and web scraping

When we give command to send mail then our system send mail from voice command and results obtained as below:

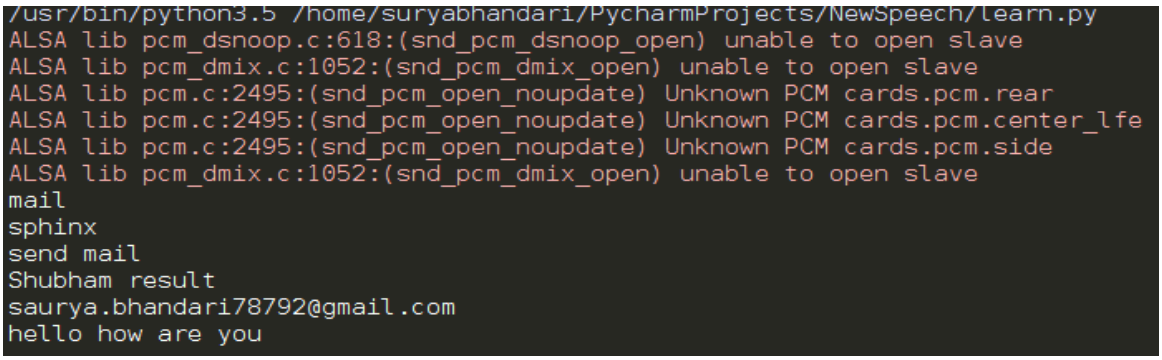

Fig 11: Results of mail handling

VI. CONCLUSION AND FUTURE ENHANCEMENT

6.1 Conclusion

During the course of this project we have learnt a great deal of programming in application level. We learned variety of resources and libraries along with the programming in Python. Basically, we built a software package that combines different modules and libraries. Designing a system, calibrating the values, applying the algorithms for manipulation of the images for the detection of faces, recognizing the voice command, to analyze the string and executing the command.

6.2 Future Enhancement

The system we designed doesn't have a good accuracy of face detection and speech recognition which can be enhanced further more. Neural network can be used where speech is analyzed by natural language processing. Machine learning can be done by capturing different type of faces in different environment and local language can be implemented through natural language processing. Furthermore, mobile robots and movements of its arms can be controlled through voice command.

cognizancejournal.com
Grishma Poudel *et al*, Cognizance Journal of Multidisciplinary Studies, Vol.4, Issue.8, August 2024, pg. 67-85
(An Open Accessible, Multidisciplinary, Fully Refereed and Peer Reviewed Journal)
ISSN: 0976-7797
Impact Factor: 4.843
Index Copernicus Value (ICV) = 77.57

References

[1] M. Mukund, "Offline Speech to Text conversion," pp. 5-9, 13 may 2016.

[2] S. Khairi, Artist, *Haar Algorithm in Face Detection.* [Art]. 2007.

[3] Md. Abdur Rahim, Md. Najmul Hossain, Tanzillah Wahid, "Face Recognition using Local Binary Patterns (LBP)," *Global Journal of Computer Science and Technology Graphics & Vision* , vol. 13, no. 4, pp. 2-7, 2013.

[4] "PyAudio Documentation," [Online]. Available: https://people.csail.mit.edu/hubert/pyaudio/docs/. [Accessed 2006].

[5] Tutorials Point, "Tutorialspoint," 2017. [Online]. Available: https://www.tutorialspoint.com/about/contact_us.htm.

[6] pip, "pip documentation," 2016. [Online]. Available: https://pip.pypa.io/en/stable/.

[7] eWEEK, "JetBrains Strikes Python Developers with PyCharm 1.0 IDE," October 2010. [Online]. Available: http://www.eweek.com/development/jetbrains-strikes-python-developers-with-pycharm-1.0-ide.

[8] JetBrains, "PyCharm 3.0 Community Edition source code now available," October 2013. [Online]. Available: https://blog.jetbrains.com/pycharm/2013/10/pycharm-3-0-community-edition-source-code-now-available/.

[9] python, "google 1.9.3," 2017. [Online]. Available: https://pypi.python.org/pypi/google.

[10] WebHarvy, "What is Web Scraping ?," [Online]. Available: https://www.webharvy.com/articles/what-is-web-scraping.html.

[11] L. Richardson, 2015. [Online]. Available: https://www.crummy.com/software/BeautifulSoup/bs4/doc/.

[12] Python, "SMTP protocol client," [Online]. Available: https://docs.python.org/2/library/smtplib.html.

[13] M. J. Foley, "Microsoft's 'Cortana' alternative to Siri makes a video debut," march 2014. [Online]. Available: http://www.zdnet.com/article/microsofts-cortana-alternative-to-siri-makes-a-video-debut/.

[14] N. Kaur, "Review of Face Recognition System Using MATLAB," *International Journal of Computer Science Trends and Technology (IJCST)* , vol. 4, pp. 30-33, 2016.

[15] Nil Goksel-Canbek Mehmet Emin Mutlu, "On the track of Artificial Intelligence: Learning with Intelligent Personal Assistants," *International Journal of Human Sciences* , vol. 13, no. 1, pp. 552-600, 2016.

[16] L. Gil, "How to Use Siri on Apple Watch," 12 may 2015. [Online]. Available: https://www.macrumors.com/how-to/siri-apple-watch/.

[17] V. Loans, "Google Assistant could come to non-Pixel Android smart phones soon," 13 February 2017. [Online]. Available: http://vakloans.com/blog/2017/02/13/google-assistant-come-non-pixel-android-smart-phones-soon/.

[18] Python Software Foundation, "python," 201. [Online]. Available: https://www.python.org/doc/essays/blurb/.

[19] OpenCV team, "opencv," may 2004. [Online]. Available: http://opencv.org/.

[20] L. Richardson, "Beautiful Soup Documentation," january 2004. [Online]. Available: https://www.crummy.com/software/BeautifulSoup/bs4/doc/.

[21] J. Couto, "Monkeylearn," *The Definitive Guide to Natural Language Processing*, pp. 7-14, 20 february 2007.

[22] m. r. v. jonsd, "eSpeak: speech synthesis," 10 february 2006. [Online]. Available: https://sourceforge.net/projects/espeak/.

