



# INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

## INTERACTIVE ASSISTANT USING FACE RECOGNITION

<sup>1</sup>Kamuju Naganandini, <sup>2</sup>Dr Prakash Bethapudi, <sup>3</sup>Nemani Sai Pallavi, <sup>4</sup>Addepalli Geetha Sri, <sup>5</sup>Teki Sai Srujana

<sup>1</sup>B.Tech II, <sup>2</sup>Professor, <sup>3</sup>B.Tech II <sup>4</sup>B.Tech II, <sup>5</sup>B.Tech II

<sup>1,2,3,4,5</sup>Information Technology,

<sup>1,2,3,4,5</sup>Vignan's Institute of Engineering for Women, Visakhapatnam, India

**Abstract:** This Project follows an interactive assistant which is accessible by authorized personnel using face recognition and performs the user requests, using this assistant we launched a poster with the help of IOT in which the system used register server module to connect with the node mcu to run the aurdino code which is connected with a DC motor to run the poster down. When this project is executed the assistant invokes and asks the name of the user, after that it asks the user for specific tasks, if the user gives any task it recognises with the help of GTTS package and performs given tasks. The assistant can perform two types of tasks. They are basic tasks and authorized tasks. if the user requests basic tasks it normally executes without any authorization. If the user requests authorized tasks then the assistant checks whether the user is authorized person or not, if yes it executes the request else it denies the access for the task.

**Index Terms – Face Recognition, Authorized, Interactive, Tasks, Assistant.**

### INTRODUCTION

This Project is developed in python2.7.X for pc, the code is written in python using multiple packages for recognition of faces, listening of user's voice and replying, performing the given tasks, invoking the aurdino board if needed. Whenever the project is executed output screen appears in which conversation between the user and system can be seen. The assistant invokes and asks the user's name and greets the user. The assistant can perform different tasks like searching the web browser, opening an app, simple numerical calculations. The tasks are divided into basic tasks and special tasks. Basic tasks doesn't need any confirmation to execute but when any special task is requested by the user it asks the user to confirm the task initiation with face recognition now, a new window appears which uses camera to record the face data, if the face data is matched with the database of the system the assistant the requested task executes if not the assistant replies with a message of access denied due to authorisation error. This project the special task is launching a poster from a height which is accessible only by the authorised person. To run the poster automatically a DC motor must be fixed to one end of the poster and the motor have to connect with aurdino board to start working the aurdino circuit and the system in which the assistant uses are connected with register server module from internet. A wifi module and IC08 is used to receive signal from the system. After running all these tasks the assistant gives a confirmation message "task finished" and comes to the normal position. The assistant asks the user for any other task to do if the answer is exit or close the assistant the assistant bids goodbye and closes the process.

### EXISTING SYSTEM

The module is focussed on the facial recognition. the project was developed on python 3.X using OpenCV often known as computer vision. The opencv module has the wide range of applications on the face recognition. Face recognition module is developed in the virtual environment, Microsoft power shell is used in this project for setting up the virtual environment. A Virtual Environment, put simply, is an isolated working copy of Python which allows you to work on a specific project without worry of affecting other projects It enables multiple side-by-side installations of Python. the opencv is installed in virtual environment to avoid confusion, to create virtual environment in python we need to install virtual environment of pip by running Pip install virtualenv pipenv command in Microsoft powershell and virtual environment can be installed after changing the execution policy to all users. Set up of execution policy for installing opencv: Our initial step is to give permissions to all users to execute the scripts in the drives. Giving permissions is done by administrator so we run the power shell as administrator. In the power shell we give the command to give permissions. The command is: Set-Executionpolicy RemoteSigned Given Screenshot of the system will give a visual of giving permissions After giving command as 'Y' the execution policy is changed to all users The system is ready to install and execute the opencv and its scripts



Fig 1. Setting Up Execution Policy

```

PS C:\Users\NANDU> pip install virtualenv pipenv
Collecting virtualenv
  Downloading https://files.pythonhosted.org/packages/62/77/6a86ef945ad39aae34aed4cc1ae4a2f941b9870917a974ed7c5b6f
/virtualenv-16.7.8-py2.py3-none-any.whl (3.4MB)
  |-----| 3.4MB 1.3MB/s
Collecting pipenv
  Downloading https://files.pythonhosted.org/packages/13/b4/3ffa55f77161cff9a5220f162670f7c5eb00df52e00939e203f601
/pipenv-2018.11.26-py3-none-any.whl (5.2MB)
  |-----| 5.2MB 1.6MB/s
Requirement already satisfied: pip>=9.0.1 in c:\users\nandu\appdata\local\programs\python\python36\lib\site-packag
om pipenv) (19.3.1)
Requirement already satisfied: setuptools>=36.2.1 in c:\users\nandu\appdata\local\programs\python\python36\lib\sit
ages (from pipenv) (40.6.2)
Requirement already satisfied: certifi in c:\users\nandu\appdata\local\programs\python\python36\lib\site-packages
pipenv) (2019.11.28)
Collecting virtualenv-clone>=0.2.5
  Downloading https://files.pythonhosted.org/packages/ba/f8/50c2b7dbc99e05f5ce5e5b9d9a31f37c988c99acd4e8dedd720b7b8
/virtualenv_clone-0.5.3-py2.py3-none-any.whl
Installing collected packages: virtualenv, virtualenv-clone, pipenv
Successfully installed pipenv-2018.11.26 virtualenv-16.7.8 virtualenv-clone-0.5.3
PS C:\Users\NANDU> virtualenv testopencv
Using base prefix 'c:\users\nandu\appdata\local\programs\python\python36'
New python executable in C:\Users\NANDU\testopencv\Scripts\python.exe
Installing setuptools, pip, wheel...
done.
PS C:\Users\NANDU>

```

Fig 2: Installation of Virtual and Pipe Environment

The above picture shows how to install the virtual environment the command is 'virtualenv filename'. The filename which was given in the above command will contain the virtual environment. Now open the file that contains the virtual environment and activate it with the command '.\Scripts\activate' when the file activates install the opencv with the command 'pip install opencv-contrib-python --upgrade' this command will download the opencv and numpy packages. The opencv is used for the face recognition and numpy is used for calculations and numerical calculations involved in the opencv. the screenshots below will give a brief visual of the process. The installed packages are numpy and opencv-contrib-python.

```

PS C:\Users\NANDU\testopencv> .\Scripts/activate
(testopencv) PS C:\Users\NANDU\testopencv> pip install opencv-contrib-python --upgrade
Collecting opencv-contrib-python
  Downloading https://files.pythonhosted.org/packages/bf/2a/954507b9308378cc0de3082bf0716dcfd915fe6f9c1bb786f0c6c2b21
/opencv_contrib_python-4.1.2.30-cp36-cp36m-win_amd64.whl (39.4MB)
  |-----| 39.4MB 3.3MB/s
Collecting numpy>=1.11.3
  Using cached https://files.pythonhosted.org/packages/b0/ee/5ff445dd43b9820e5494d21240e689d3b7cb52bc93f4f164eba84206
d/numpy-1.17.4-cp36-cp36m-win_amd64.whl
Installing collected packages: numpy, opencv-contrib-python
Successfully installed numpy-1.17.4 opencv-contrib-python-4.1.2.30
(testopencv) PS C:\Users\NANDU\testopencv>

```

Fig 3: Installation of Opencv

To check the opencv installed without errors just type python at the prompt of powershell if everything is right the python prompt and version is displayed which means the python is installed, now at the prompt import cv2 and print cv2 version

```

(testopencv) PS C:\Users\NANDU\testopencv> python
Python 3.6.8 (tags/v3.6.8:3c6b436a57, Dec 24 2018, 00:16:47) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> print(cv2.__version__)
4.1.2
>>>

```

Fig 4: checking the Opencv and its version

The system is ready to execute the opencv scripts and the face recognition. The code of face recognition module developed in the python language and the script needed in the project is saved at the virtual environment created recently. Whenever this module is needed to run the user must travel to the folder where the opencv resides and execute it from there. This module focusses on the face recognition and the code will be discussed according to the face recognition code. The code contains two files first file 'faces-train.py' is used to train the faces of the user and the second file 'faces.py' is used for face recognizing the faces. virtual environment of opencv helps the faces to train and the images saved in the data folder in the same folder where the code exists. The images that contain the data of the user must have a png extension and for better recognition the images used in the training should have better quality and only face of the user is encouraged. When the faces-train.py file is executed the haar cascades which have to be saved in that python scripts folder from the library folder 'lib' created in the opencv environment folder. The haar cascades are the A Haar Cascade is basically a classifier which is used to detect the object for which it has been trained for, from the source. The Haar Cascade is trained by superimposing the positive image over a set of negative images. The training is generally done on a server and on various stages. In the developed code of this project uses the 'frontalface' cascade, it is an '.xml' file which is used to recognize the frontal face of the user if the user is recognised from the database then the name and a box is printed along the borders of the face of the user and the name of the user is stored continuously in the variable of the user's data and the name predicts and prints only if the confidence of the prediction is within the range.



The confidence range given in this project is between four to eighty five percent. the faces are trained and the system is ready to execute face recognition, this module is further integrates with the all the other module. Given screenshots give the result of the module. For Face recognition module user need to create a dataset to store the photos and that file destination should be given in the program. Now import cv2 and numpy modules. The most common way to detect a face (or any objects), is using the " haarcascade\_frontalface\_default.xml" Cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones in their paper, "Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001. It is a machine learning based approach. It is then used to detect objects in other images. if you want to train your own classifier for any object like car, planes etc. you can use OpenCV to create one. If you do not want to create your own classifier, OpenCV already contains many pre-trained classifiers for face, eyes, smile, etc. Those XML files can be download from haarcascades "directory.faceDetect=cv2.CascadeClassifier ('haarcascade\_frontalface\_default.xml');" This is the line that loads the "classifier" that must be in a directory named "Cascades/",under the project directory. Then, set the camera inside the loop, load the input video in grayscale mode. train the dataset using faces-train.py program which consists of recognizer folder in the same directory this folder should have a " trainerData.yml".Now you need to implement detector program to detect your faces with your name and we have inserted a mp3 audio clip that consists the information of our college to import this audio file we need to use command " from playsound import playsound".So after detection of the faces the audio will pe played .Then the signal is transferred to aurdino which is connected with wifi module and this helps the motor to rotate

```
PS C:\Users\NANDU> cd Dev
PS C:\Users\NANDU\Dev> cd opencvtube
PS C:\Users\NANDU\Dev\opencvtube> ./Scripts/activate
(opencvtube) PS C:\Users\NANDU\Dev\opencvtube> python
Python 3.6.8 (tags/v3.6.8:3c6b436a57, Dec 24 2018, 00:16:47) [MSC v.1916 64 bit (AMD64)] on win32
type "help", "copyright", "credits" or "license" for more information.
>> import cv2
>> print(cv2.__version__)
4.1.2
>> ls
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'ls' is not defined
>> ^Z

(opencvtube) PS C:\Users\NANDU\Dev\opencvtube> cd src
(opencvtube) PS C:\Users\NANDU\Dev\opencvtube\src> ls
```

Fig 5: navigate to the source file

```
(opencvtube) PS C:\Users\NANDU\Dev\opencvtube\src> python faces-train.py
(opencvtube) PS C:\Users\NANDU\Dev\opencvtube\src> python faces.py
```

Fig 6: Train and test the data

## SPEECH RECOGNITION

This module will allow the user to [3]interact with the assistant and system. The module explains how the voice assistant works based on the user's requirements and how it is done. This module is developed in [4]python 3 and this module need additional packages to be installed via pip in the systems environment. The additional packages are 'google [6]text to speech' for text converting, [5]'speech\_recognition' for analyzing the user's voice, 'selenium' for web browsing, 'playsound' for playing the saved files of the assistant and playing it. The code developed for this module is integrated with the face recognition module. this module works using the internet because the code involves external packages like gtts, selenium which works on the internet.

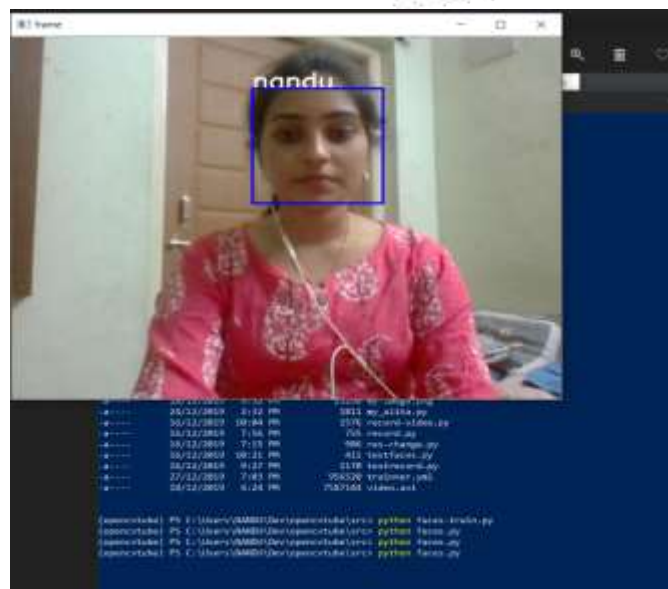


Fig 7: Run the program

## CIRCUIT FOR THE MOTOR

A DC motor is the most common type of motor.[7]DC motors normally have just two leads, one positive and one negative. To rotate the motor used components are jumper wires, ARDUINO UNO, BREADBOARD, BATTERY 9V, IC, NODEMCU. 'Jumper wires' are simply 'wires' that have connector pins at each end, allowing them to be used to connect two points to each other without soldering. Few steps been followed

- Connect 5V and the ground of the IC to 5V and the ground of Arduino, respectively.
- Connect the motor to pins 2 and 3 of the IC.
- Connect IN1 of the IC to pin 8 of Arduino.
- Connect IN2 of the IC to pin 9 of Arduino.
- Connect EN1 of IC to pin 2 of Arduino.
- Connect SENS A pin of IC to the ground.
- Connect Arduino using Arduino USB cable and upload the program to Arduino using Arduino IDE software.
- Provide power to Arduino board using power supply, battery, or USB cable.

In this project the power is connected and passed through the 9v cells and connecting them in series will make a battery and the speed is controlled by number of cells. Use of higher voltage battery will burn out the arduino and other integrated circuits. After completion of facial recognition it invokes the Arduino then the connected motor starts it work and let the poster down.

## HARDWARE REQUIREMENTS

### ARDUINO UNO

Arduino UNO is a open-source micro controller developed by Arduino.cc. The board is equipped with 20 digital and analog input, output pins that can be interfaced to various expansion boards and circuits.



Fig 8: arduino uno

### BREAD BOARD

A bread board is a construction base for prototyping of electronics. The arduino uno, L293d, nodeMCU and all the jumper wires are connected to the bread board where the required circuit can be developed.



Fig 9: bread board

### NODE MCU

NodeMCU is an open source LUA based firmware developed for ESP8266 wifi chip. By exploring functionality with ESP8266 chip, NodeMCU firmware comes with ESP8266 Development board i.e. NodeMCU Development board. Since NodeMCU is open source platform, their hardware design is open for edit/build. NodeMCU Dev Kit/board consist of ESP8266 wifi enabled chip. The ESP8266 is a low-cost Wi-Fi chip developed by Espressif Systems with TCP/IP protocol. For more information about ESP8266, you can refer ESP8266 WiFi Module.



Fig 10: Node mcu

### JUMPER WIRES

Jumper wire is simply a wire that have connector pins at each end, allowing them to be used to connect two points to each other without soldering. Jumper wires are typically used with breadboards and other prototyping tools in order to make it easy to change a circuit as needed. Fairly simple. In fact, it doesn't get much more basic than jumper wires.



Fig 12: Jumper wires (female – female)

**L293D MOTOR DRIVER**

The L293D is a popular 16-Pin Motor Driver IC. As the name suggests it is mainly used to drive motors. A single L293D IC is capable of running two [8]DC motors at the same time, the direction of these two motors can be controlled independently. So if you have motors which has operating voltage less than 36V and operating current less than 600mA, which are to be controlled by digital circuits like Op-Amp, 555 timers, digital gates or even Micron rollers like Arduino, PIC, ARM etc. this IC will be the right choice for you.



Fig 11: L293D Motor Driver

**BATTERY**

The nine-volt battery is a common size of battery that was introduced for the early transistor radios. It has a rectangular prism shape with rounded edges and a polarized snap connector at the top.



Fig 13: 9v battery

**ARDUINO UNO WIFI**

This is a module where nodemcu ESP8266 is connected with Arduino IDE. Node mcu ESP 8266 which is an open source platform that can connect objects and allows data transfer using wifi protocol. After completion of face recognition using register server module we send signal to nodemcu ESP8266 which is connected with Arduino. As mentioned above system passes a command to run motor for poster launch as an Information to nodemcu ESP8266 and then it will collaborate with Arduino to run the motor, thus poster launch protocol will be initiated. In this module the Arduino Uno WiFi is highly focused. The Arduino Uno WiFi is an Arduino Uno with an integrated WiFi module. The board is based on the AtMEGA328P with an ESP8266 WiFi Module integrated. The ESP8266 [9]WiFi Module is a self-contained SoC WiFi integrated TCP/IP protocol stack that can give access to your WiFi network (or the device can act as an access point). One useful feature of Uno WiFi is to support for OTA (over-the-air) programming either for transfer of Arduino sketches or WiFi firmware. The Arduino Uno [10]WiFi is programmed using the Arduino Software (IDE), Integrated Development Environment common to all the boards of the project and running both online and offline. Nodemcu will serve a web page on your mobile, desktop, laptop, network web browser. Web page enlists the DC motor controls. User can control the rotation speed of the motor from web page and change the DC motor rotation direction through web page controls. This system controlled forward rotation with the required speed and the length of the time. The speed of the motor is decided by the number of the cells connected to its battery. One DC motor is used in this project with the power consumption of 12 volts at 300mA of current and uses a battery made up of 9V cells.

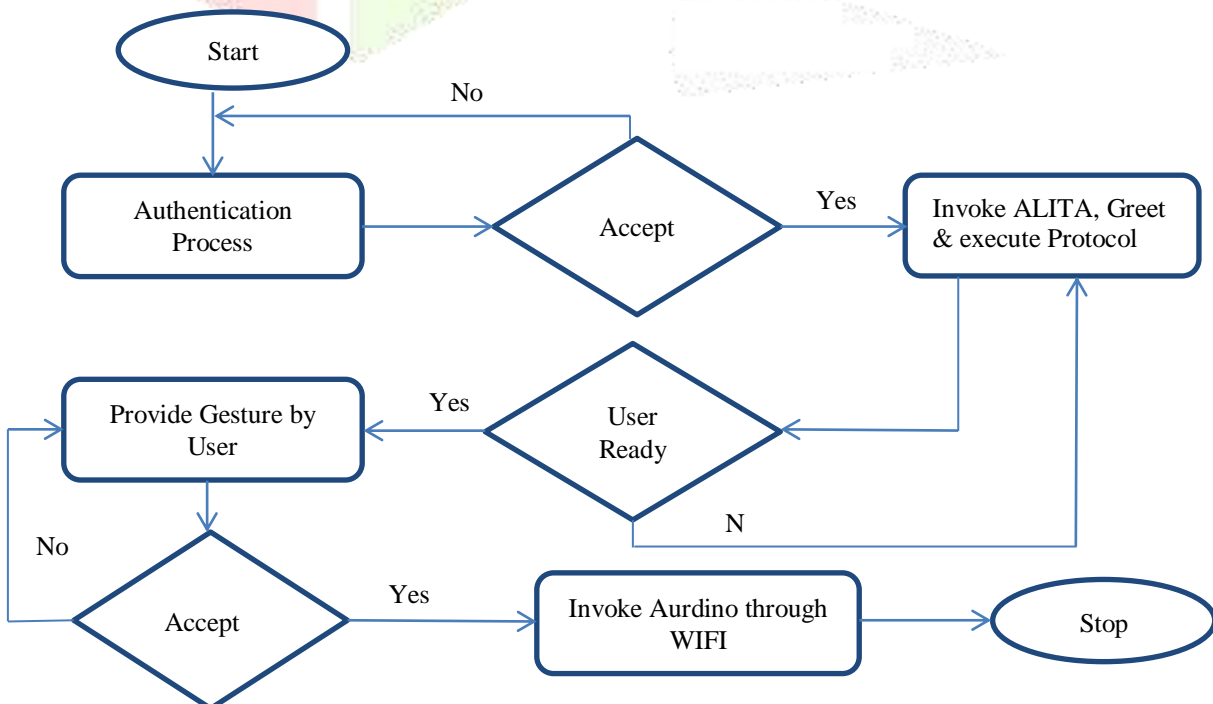


Fig 14: Flow chart



## CONCLUSION

In this paper, face recognition is used. The system is capable of recognizing faces and overcome following challenges,

- Objects partially covering face (e.g.: Sunglasses, Long Hair)
- Low resolution images
- Facial expressions
- Microphone and webcam failure in the system

when the program starts the webcam is opened when it recognizes the face of either principal sir or HOD sir (in this project the data base contains the principal and head of the department of the college) and then it invokes voice assistant .the assistant greets the people and introduce about it and then it invites the user to stand in front of the camera it starts recognising the faces and then Arduino initiates the poster launch protocol.



Fig 9: Prototype of the project

Above picture represents the project's integral part which is the laptop. A paper robot is made by cardboard and speakers and paint for the look of the project. The laptop is placed at the centre void of the paper robot and the arduino circuit and poster can be placed on the desired area only it has to connect with the same wifi connection and no obstacles must be there in the middle. further development of this project can be packed into a package like an end-user application and improved facial recognition is observed which may be expected in future versions.

## SCOPE FOR FURTHER DEVELOPMENT

Algorithm used for the face recognition could be enhanced and results in faster recognition. This project used speech recognition if we could upgrade the algorithm using voice recognition the authorised person can access using voice commands alongside face recognition.

## REFERENCES

1. FACE RECOGNITION - <http://joincfe.com/github/>,
2. Face Recognition Python Coding <http://kirr.co/712sv4/>
3. Automatic speech recognition –A Review, Shipraj Arora et al, International Journal of Computer Applications, Vol 60, No-9, Dec 2012
4. SPEECH RECOGNITION - <http://kirr.co/712sv4/>
5. C Breazeal et al, Recognition of effective communicative intent in robot directed speech autonomous robots2(2002) 83-104
6. M A Anusuya et al, 2009, Speech Recognition by Machine : Review, International Journal of Computer Science and Information Study, Vol 6, No 3
7. Design and Implementation od driver dircuits for DC motor using 8051, Ashmi, et al, International Journal of Engineering Research and Technology, Vol 2, Issue 6, Jun 2013
8. CIRCUIT FOR THE MOTOR - <https://www.engineersgarage.com/esp8266/nodemcu-and-l293d-motor-driver-controlling-dc-motor/>
9. Design and implementation of WI-FI based Automation System, Ahmed Elishafra et al, World Academy of Survey, Engineering and Technology 68, 2012, 2177 – 2183
10. ARDUINO UNO WIFI - <https://www.arduino.cc/en/Guide/ArduinoUnoWiFi>