Smart Aid For Speech And Hearing Impaired Community Using CNN

PROJECT REPORT

Submitted by

Akash (15BCE1369**)**

Manisha Chaudhary (15BCE1358**)**

Lavanya Shivani (15BCE1076**)**

in partial fulfillment for the award of the degree of

Bachelor of Technology

in

Computer Science and Engineering

School of Computing Science and Engineering

Vandalur - Kelambakkam Road, Chennai - 600 127

April - 2019

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Computing Science and Engineering

DECLARATION

We hereby declare that the project entitled "Smart Aid For Speech And Hearing Impaired Community Using CNN" submitted by us to the School of Computing Science and Engineering, VIT Chennai, 600127 in partial fulfilment of the requirements for the award of the degree of B.Tech CSE is a bonafide record of the work carried out by us under the supervision of Dr. B V A N S S Prabhakar Rao. We further declare that the work reported in this project, has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma of this institute or of any other institute or University.

Place : Chennai

Signature of Candidate (s)

Date :

Akash (15BCE1369)

Manisha Chaudhary(15BCE1358)

Lavanya Shivani (15BCE1076)

School of Computing Science and Engineering

CERTIFICATE

This is to certify that the report entitled " Smart Aid For Speech And Hearing Impaired Community Using CNN" is prepared and submitted by Akash (Reg. No. 15BCE1369), Manisha Chaudhary (Reg. No. 15BCE1358), & Lavanya Shivani (Reg. No. 15BCE1076), to VIT Chennai, in partial fulfilment of the requirement for the award of the degree of B. Tech CSE programme is a bonafide record carried out under my guidance. The project fulfils the requirements as per the regulations of this University and in my opinion meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma and the same is certified.

Signature of the Guide:

Name: Dr. B V A N S S Prabhakar Rao Date:

Signature of the Internal Examiner	Signature of the External Examiner
Name:	Name:
Date:	Date:

Approved by the **B. Tech CSE Program Chair**

Name: Dr. B Rajesh Khanna

Date:

(Seal of SCSE)

Acknowledgement

We wish to express our sincere thanks to a number of people without whom we could not have completed the thesis successfully.

We would like to place on record our deep sense of gratitude and thanks to our guide Dr. B V A N S S Prabhakar Rao, School of Computer Science and Engineering (SCSE), VIT, Chennai campus whose esteemed support and immense guidance encouraged us to complete the project successfully.

We would like to thank our Program Chair Dr. B Rajesh Kanna & Co-Chair, Dr. C Sweetlin Hemalatha, B. Tech. Computer Science and Engineering and Project Coordinator Dr. B V A N S S Prabhakar Rao, VIT Chennai campus, for their valuable support and encouragement to take up and complete the thesis.

Special mention to our Dean, Dr. Vaidehi Vijayakumar, & Associate Deans Dr. Vijayakumar V and Dr. Subbulakshmi T, School of Computing Science and Engineering (SCSE), VIT Chennai campus, for spending their valuable time and efforts in sharing their knowledge and for helping us in every minute aspect of software engineering.

We thank our management of VIT Chennai campus for permitting us to use the library resources. We also thank all the faculty members for giving us the courage and the strength that we needed to complete our goal. This acknowledgment would be incomplete without expressing the whole hearted thanks to our family and friends who motivated us during the course of our work.

Akash (Reg. No. 15BCE1369)

Manisha Chaudhary (Reg. No. 15BCE1358)

Lavanya Shivani (Reg. No. 15BCE1076)

Table Of Contents

List of Figures

Executive Summary

In this era of 21st century, technologies play an important role in making our lives sustainable. Life is very easy for the ones who are god gifted, but this doesn't hold good for the ones who suffer from certain disabilities like hearing con or speech disorder. Others can interact among themselves without any kind of hindrance, but this isn't affirmative for the differently abled. So our project throws light on this societal issue, thus proposing a framework using CNN which would further help the people anthropoid to deal with all such complications with the differently abled group, deaf and dumb. The proposed scheme is cost effective and can be afforded by all group humans. For the talking and hearing disabled minority, there is a communication gap. Visual guides, or a translator, are utilized for speaking with them. However, these techniques are somewhat cumbersome and costly, and can't be utilized in a crisis. So our proposed schema requires less time and is quite reliable and easily accessible too. Input is given in the form of the image which would further get converted into text and then further into voice accordingly. The text can be read by deaf and dumb and the voice can be heard by the dumb thus helping them to reciprocate to the world be it physically challenged or the common people. Therefore we can implement this system which would serve as a helping hand to the deaf and dumb and which would also help them to get a bit closer to the world.

Abstract

Deaf and dumb people use gesture based communication to communicate with the world using hand gestures. Sign language is the only single way of communication for deaf and dumb people. But common people face difficulty in understanding the gesture language therefore often these physically challenged people has too keep the translator with them to communicate with the world. The progression in implanted framework can give a space to plan and build up an interpreter framework to change over the communication via gestures into discourse. These days implanted framework has turned into a critical pattern in all applications. Our prototype can be used for the welfare of the society . Firstly for the travelling purpose , dumb people face problems while expressing . So they can use our translator which gets converted into text thus helping the pedestrians or the common people. This can be used the either way too that is conversation amongst different groups of deaf and dumb. Different group refers to deaf and dumb people knowing different sign languages as (ASL,ISL,BSL etc.). Secondly , this can be used for the shopping purpose when the mute people can't reply to the owner or the deaf people can't hear to the directions shown through audio. Also , for the meeting purposes where one of the co-worker wants to share his/her views with (HR) but can't speak so here our prototype comes into the picture . Now let us take the situation where the HR who is dumb wants to convey some regards to the one who is deaf, to which he can use our prototype which could be helpful enough to make a successful meeting . So we would be using Convolution Neural Network (CNN) which would implement our prototype .Thus reducing the pre-processing and enhancing the overall recognition process. The work exhibited fundamentally lessens the correspondence gap amongst imbecilic and standard individuals and intends to encourage dumb individuals way of life.

1.Introduction

Motion of anyone part like face, hand is a type of Gesture. Here for gesture acknowledgment we are utilizing CNN. Gesture acknowledgment empowers PC to comprehend human activities and furthermore goes about as an translator among PC and human. This could give potential to human to connect normally with the PCs with no physical contact of the mechanical gadgets. Gestures are performed by deaf and dumb to perform gesture based communication. This people group utilized communication via gestures for their correspondence when broadcasting sound is outlandish, or composing and composing is troublesome, yet there is the vision probability. Around then communication through signing is the way to trade data between individuals. Ordinarily communication through signing is utilized by everybody when they would prefer not to talk, however this is the main method for correspondence for hard of hearing and dumb. Communication through signing is likewise serving a similar importance as spoken language does. This is utilized by hard of hearing and dumb community everywhere throughout the world yet in their local structure like ISL, ASL. Sign language can be performed by utilizing Hand signal either by one hand or two hands. It is of two kind Isolated gesture based communication and constant communication via gestures. Separated communication through signing comprises of single gesture having single word while nonstop ISL or Continuous Sign language is a grouping of gestures that produce a significant sentence. In this report we performed ASL gestures recognition system.

1.1 Objective

To create an effective communication tool for the people who are not able to speak or hear anything. A User Friendly, Cost effective system which reduces communication gap between deaf and dumb with ordinary person. This framework offers voice to voiceless i.e. voice is given to the individual who can't talk. Imbecilic/quiet individuals utilize gesture based communication for correspondence reason. Communication through signing utilizes signals rather than sound to pass on data. This dialect incorporates consolidating hand shapes, hand developments, outward appearances to express person's considerations. Some of them are the variety of the hand motion appearance, scaled, pivoted adaptation of picture and the picture handling speed as it includes numerous scientific estimations. Generally, there are diverse methodologies presented to perceive hand signal.

1.2 Background

Gesture recognition and sign language recognition has been a well researched topic for American Sign Language(ASL), however few research works have been distributed with respect to Sign Language(ISL). One of the methodologies included key point discovery of Image utilizing SIFT and afterward coordinating the key point of another picture with the key points of standard pictures per letters in order in a database to arrange the new picture with the mark of one with the nearest coordinate . Another determined the Eigen vectors of covariance framework determined from the vector portrayal of picture and utilized Euclidean separation of new picture Eigen vector with those in preparing informational collection to characterize new picture . However , rather than utilizing top of the line innovation like gloves or then again Kinect, we plan to take care of this issue utilizing CNN.

Figure 1 : Real Life Problem Scenario

1.3 Commonly Used Sign Languages

On moving across the globe we came to know that there is no universal sign language . usage of sign language varies from community to community so we cannot stick to one particular language . Let us deal few of them one by one. Namely ASL(American Sign Language), ISL(Indian Sign Language) , BSL (British Sign language) etc.

ASL   -   Other than North America, lingos of ASL and ASL-based creoles are utilized in numerous nations around the globe, including quite a bit of West Africa and parts of Southeast Asia. ASL is likewise generally learned as a second language.

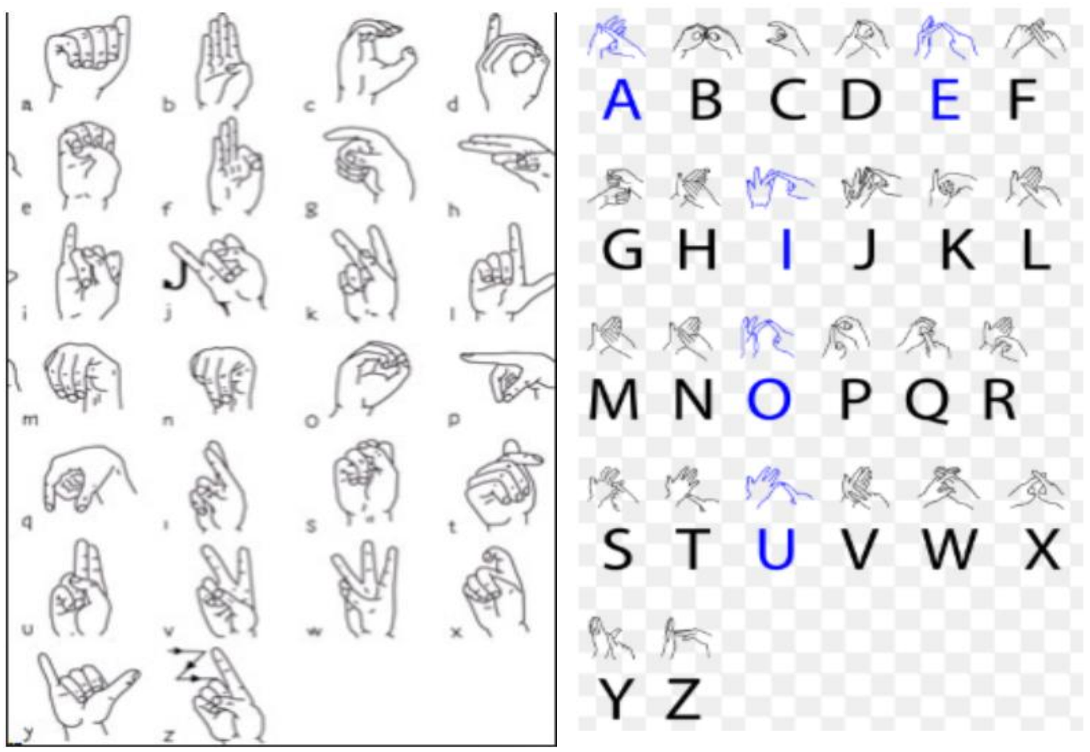


Figure 2 : Alphabets Of ASL And ISL

BSL - BSL uses two handed alphabet. Mainly used in Scotland, England, European Union and Wales.

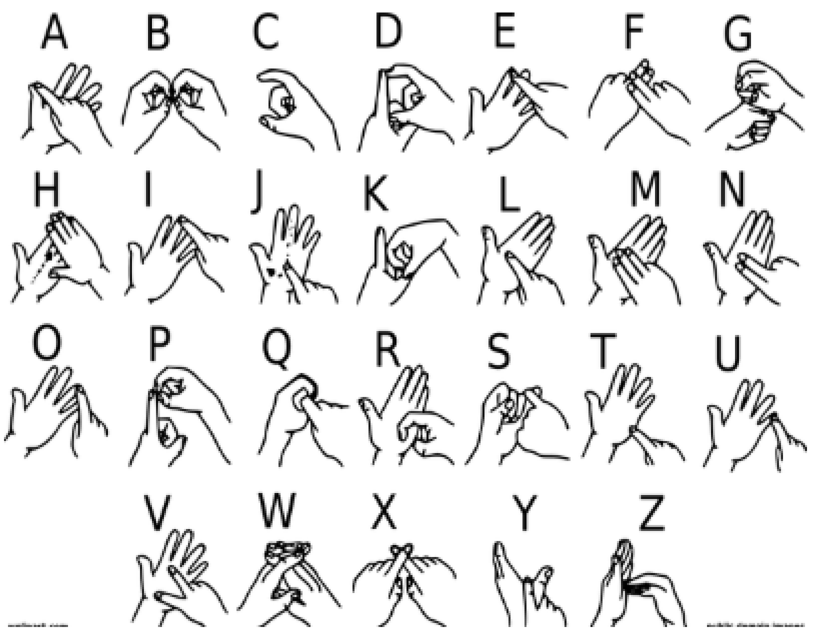


Figure 3 : Alphabets Of BSL

1.4 Motivation

Correspondence is one of the essential prerequisite for survival in the public arena. Deaf and dump people impart among themselves utilizing communication via gestures however typical individuals think that its troublesome to comprehend their language. Broad work has been done on American gesture based communication acknowledgment however Indian gesture based communication contrasts altogether from American sign language.ISL utilizes two hands for communicating while ASL utilizes single hand for conveying. Utilizing two hands regularly prompts lack of definition of highlights because of covering of hands. What's more, absence of datasets alongside fluctuation in communication via gestures with region has brought about limited endeavors in ISL motion discovery. Our undertaking goes for taking the essential advance in crossing over the correspondence gap between ordinary individuals and not too sharp individuals utilizing Indian gesture based communication. Compelling augmentation of this venture to words and typical statements may not just influence the almost totally senseless individuals to impart quicker and simpler with external world, yet additionally give a lift in creating self-ruling frameworks for understanding and helping them.

2. Project Description and Goals

2.1 Project Description

Inability to talk is viewed as evident incapacity. Individuals with this inability utilize distinctive modes to speak with others, there are number of strategies accessible for their correspondence one such regular strategy for correspondence is communication through signing. Creating communication through signing application for hard of hearing individuals can be essential, as they'll have the capacity to discuss effectively with even the individuals who don't get it communication through signing. Our project aims for taking the essential steps in connecting the communication gap between ordinary individuals, hearing and speech impaired individuals utilizing sign language. The principle focal point of this work is to make a vision based framework to recognize sign language gestures from the video groupings. The explanation behind picking a framework is relates to the fact that it gives an easier and progressively natural method for correspondence between a human and a computer. In this project, 46 diverse gestures have been considered. To prepare the model on the spatial features of the video sequences we have utilized Inception model which is a profound CNN (Convolution Neural Network). CNN was trained on the frames obtained from the video sequences of train data.

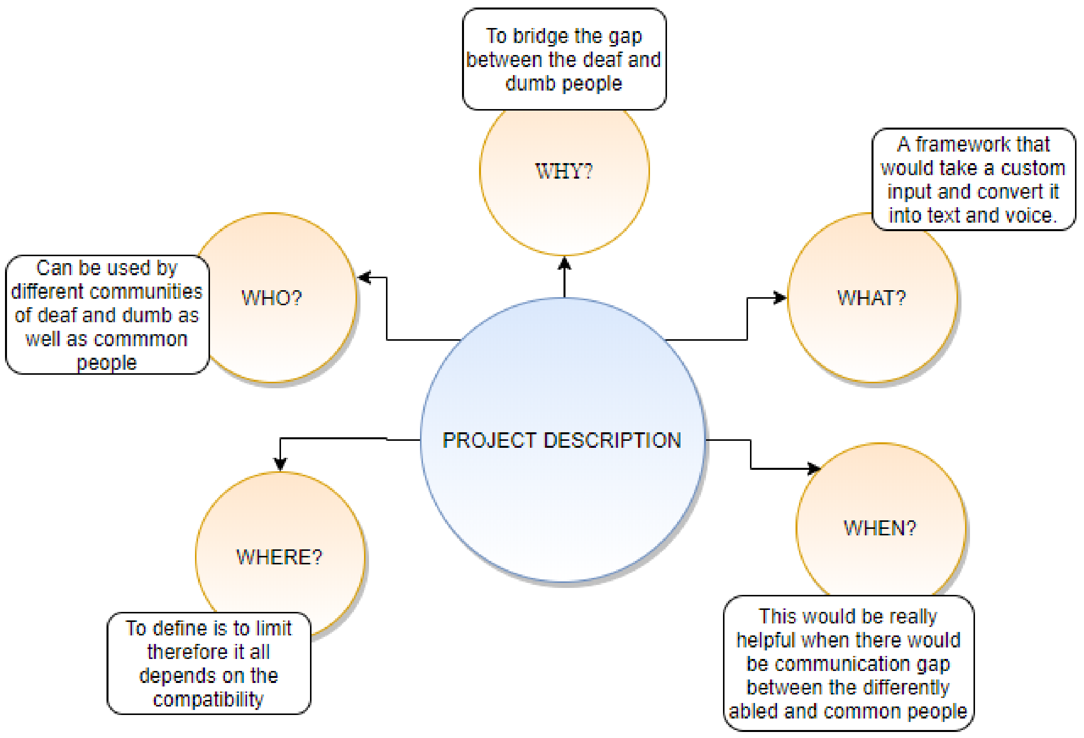

To bridge the gap between the deaf and dumb people
WHY?
A framework that would take a custom input and convert it into text and voice.
WHO?
Can be used by different communities of deaf and dumb as well as commmon people
WHAT?
PROJECT DESCRIPTION
WHERE?
WHEN?
To define is to limit therefore it all depends on the compatibility
This would be really helpful when there would be communication gap between the differently abled and common people

Figure 4 : Project Description

2.2 Goals

The major idea of our project is to create an effective communication tool for the people who are not able to speak or hear anything. A User Friendly, Cost effective system which reduces communication gap between deaf and dumb with ordinary person.

The proposed system can be implemented successfully to a major extent with high accuracy and is supposed to receive a good feedback from the people. The proposed system will be cost effective in nature and could support different languages.

Since the dataset is enormous so training and classification would be quite time consuming as well as extra storage would be required so we would further proceed working in the cloud environment as it provides these benefits:

- Easy Deployment
- Fast developments : Cloud app's can be upgraded, maintained and deployed faster than the local machine deployment over various different places. As per the user need the modifications can be done in a faster manner.

3. Literature Survey

In the ongoing years, there has been gigantic research on the hand sign language gesture recognition. The technology for gesture acknowledgment is given beneath.

3.1 Sensor Based

The sensor based system consists a combination of hardware and software modules. Hardware segment will incorporate flex sensors on every finger, a microcontroller, power supply and android application connecting through Bluetooth module. Software segment will incorporate programming for android application. Flex sensors takes input as different gestures through gloves, microcontroller is used to convert input analog signal to digital data and further processing can be done, power supply i used to provide voltages to specific units, and at last Bluetooth module is used to send the data from controller to android mobile.

ALGORITHM

Below is the algorithm of the proposed system

Step 1 Start

Step 2 Gather value from all the sensors

Step 3 Are values forming any meaning? If yes then sends conversion of sign language to text data to Bluetooth. Android app gathers data from Bluetooth and displays it.

Step 4 Android app gathers data from Bluetooth and displays it.

Step 5 Stop.

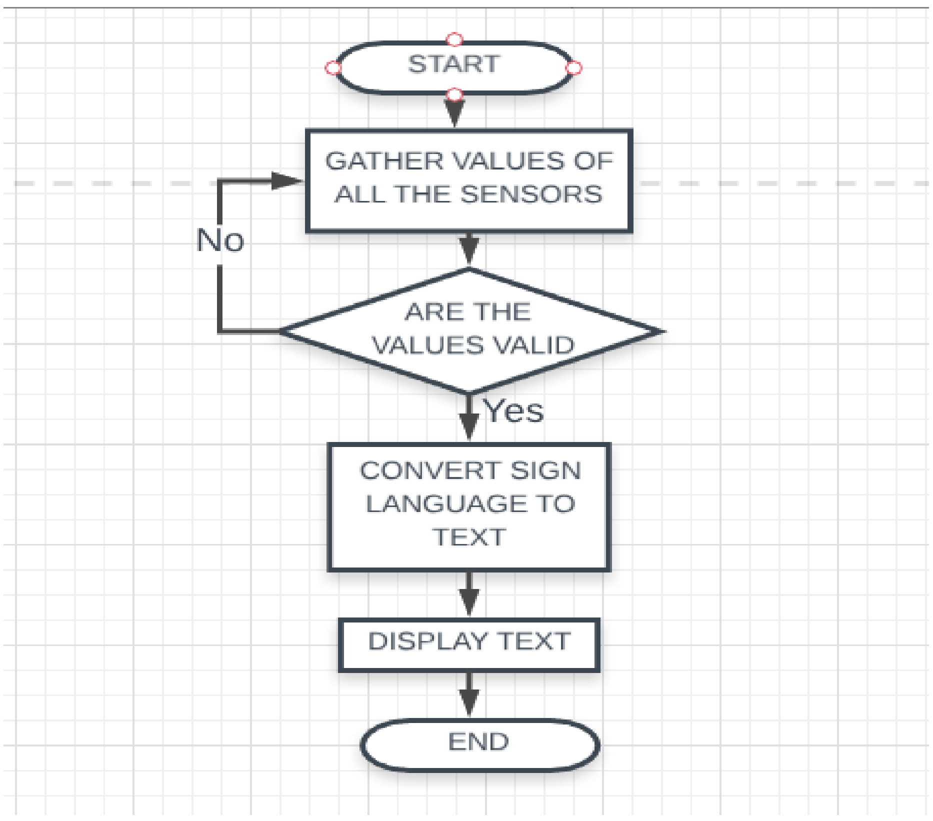

START
GATHER VALUES OF ALL THE SENSORS
No
ARE THE VALUES VALID
Yes
CONVERT SIGN LANGUAGE TO TEXT
DISPLAY TEXT
END

Figure 5 : Flow Model (Sensor Based)

3.2 Using Image Processing

ALGORITHM -

Step 1 : Reading image from camera and applying pre-processing techniques like gamma correction, blurring.

Step 2: Hand Segmentation using background subtraction algorithm

Step 3: Hand detection using thresholding and dilation

Step 4: Finding contours of hand for getting shape of hand

Step 5: Finding contour area, convex hull, hull area, solidity

Step 6: Also find the angle between two fingers and aspect ratio of hand

Step 7: Finding the defects of hand using convex hull

Step 8: Finally classifying using solidity, aspect ratio, convex defects and angle

Step 9: if image (sign) ==database image, then ON the led and speak the meaning of that sign
Repeat Step 9

END.

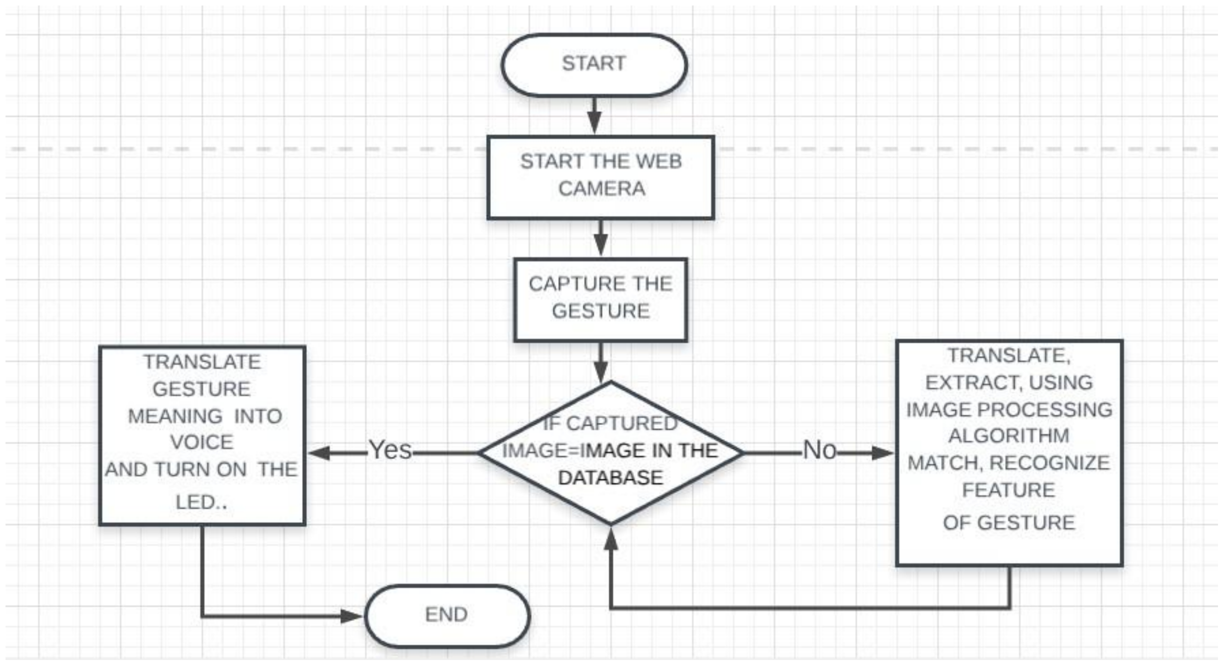

START
START THE WEB CAMERA
CAPTURE THE GESTURE
IF CAPTURED IMAGE=IMAGE IN THE DATABASE
Yes
No
TRANSLATE GESTURE MEANING INTO VOICE AND TURN ON THE LED..
TRANSLATE, EXTRACT, USING IMAGE PROCESSING ALGORITHM MATCH, RECOGNIZE FEATURE OF GESTURE
END

Figure 6 : Flow Model (Using Image Processing)

3.3 Vision Based

In vision based techniques PC camera is the info gadget for watching the data of hands or fingers. The Vision Based techniques require just a camera, along these lines understanding a characteristic cooperation among people and PCs without the utilization of any additional gadgets. These frameworks will in general supplement biological vision by depicting artificial vision frameworks that are executed in programming as well as equipment. This represents a difficult issue as these frameworks should be foundation invariant, lighting insensitive, individual and camera autonomous to accomplish constant execution. Also, such frameworks must be improved to meet the prerequisites, including accuracy and robustness. Vision based examination, depends on the way human beings perceive information about their surroundings , yet it is most difficult to implement in a way that would be quite satisfying. A few unique methodologies have been tried up until now.

1. One is to build a three dimensional model of the human hand. The model is coordinated to the image of the hand by at least one cameras, and parameters relating to palm orientation and joint angles are evaluated. The parameters generated are used to successfully perform gesture classification.

2. Second methodology is used to capture image input using a camera and then extract features from the image and those are used as input in a classification algorithm.

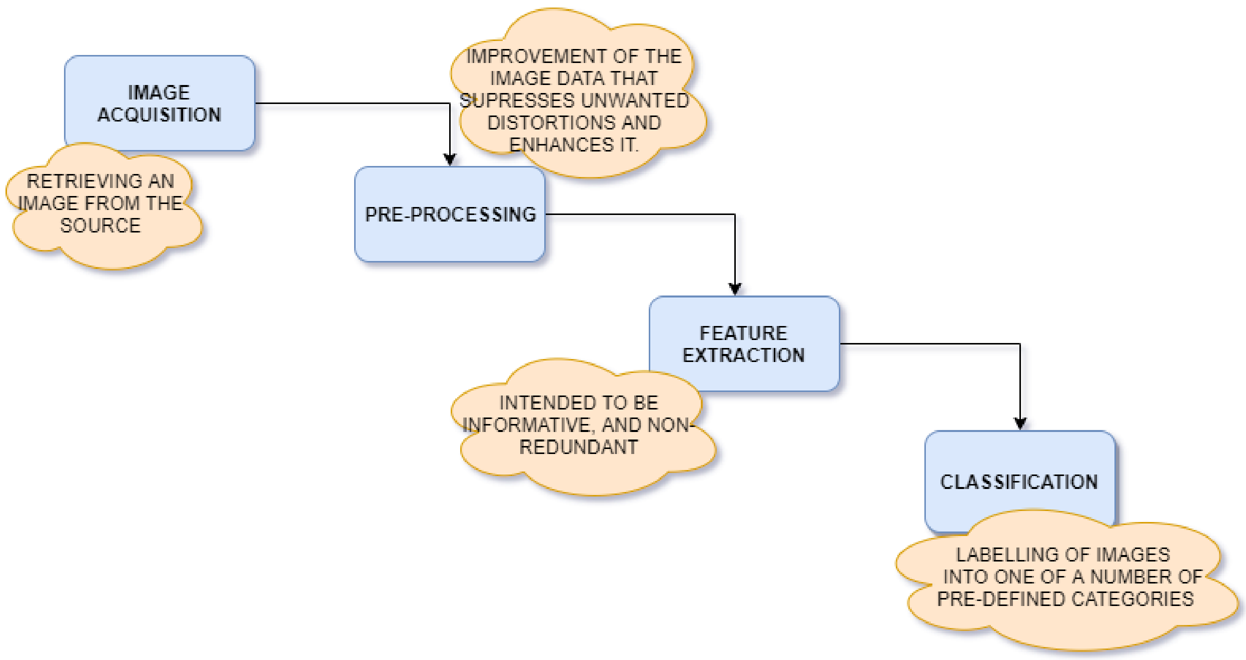


IMAGE ACQUISITION
RETRIEVING AN IMAGE FROM THE SOURCE
IMPROVEMENT OF THE IMAGE DATA THAT SUPRESSES UNWANTED DISTORTIONS AND ENHANCES IT.
PRE-PROCESSING
FEATURE EXTRACTION
INTENDED TO BE INFORMATIVE, AND NON-REDUNDANT
CLASSIFICATION
LABELLING OF IMAGES INTO ONE OF A NUMBER OF PRE-DEFINED CATEGORIES

Figure 7 : Vision Based

3.4 Using CNN

Convolution neural networks mean to utilize spatial data between the pixels of a picture. Along these lines, they depend on discrete convolution. CNNs is the most prominent neural network model is utilized for image classification purpose. The ultimate decision behind CNNs is that a local understanding of a image is sufficient. The functional advantage is that having less parameters extraordinarily improves the time it takes to learn just as diminishes the measure of data required to train the model. Rather than a fully connected network of weights from every pixel, a model has just suitable weights to observe a small spot of the image It resembles perusing a book by utilizing an amplifying glass; in the long run, you read the entire page, however you focus at just a little patch of the page at any particular time.

On analyzing the above approaches we further come to a conclusion that in Sensor-based approach normally underwriter requires to wear gloves in hand, additionally a few sensors ,etc that are utilized as sign signal to demonstrate the hand stance or motion, while vision-based methodology endorser does not require to wear anything regular hand utilized as sign flag to display hand stance or signal. We are keen on vision-based methods to perceive hand pose, which are increasingly normal method for correspondence with incapacitated individuals and robots. further on proceeding with vision based we came into a dilemma of two approaches firstly image classification algorithms secondly CNN. Since image classification algorithms requires preprocessing which is the reason why handling the data is more time taking and have several constraints . Due to above loop holes we come to a conclusion that CNN would give the more accurate results.

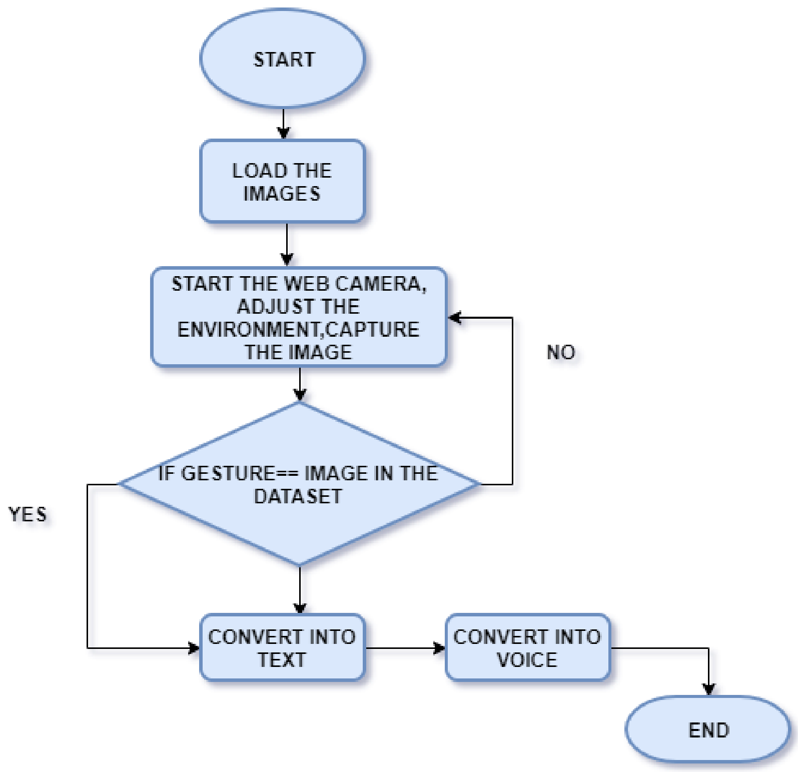

START
LOAD THE IMAGES
START THE WEB CAMERA, ADJUST THE ENVIRONMENT,CAPTURE THE IMAGE
NO
IF GESTURE== IMAGE IN THE DATASET
YES
CONVERT INTO TEXT
CONVERT INTO VOICE
END

Figure 8 : Flow Model (CNN Based)

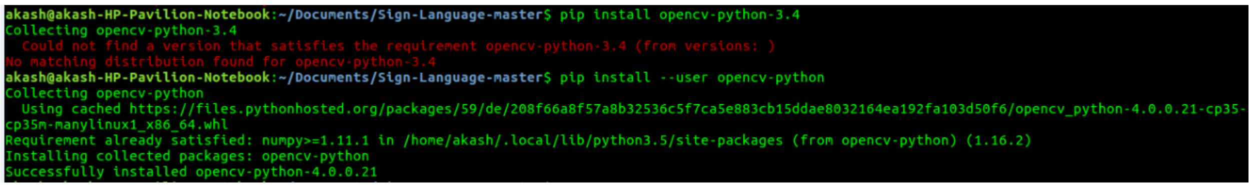
4. Technical Specification

In order to ease the life style of dumb and deaf people the proposed system is developed. CNNs is the main technique used and implemented. Processing involves data training , data testing in the cloud environment. For training and testing data we have used python libraries , python pre-packages which would provide the best environment .

4.1 Prerequisites

Python 3 : Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. The Python interpreter and the huge standard library is freely available in open source or binary format for all major platforms to use. It is an easy to learn, powerful programming language which helped us in working with the machine learning concept .

OpenCV : OpenCV (Open source computer vision) is a library of programming capacities primarily went for constant computer vision. OpenCV-Python is a library of Python ties intended to take care of computer vision issues. One of the use of Open CV is Gesture acknowledgment. The innovation goes for accomplishing the objective of deciphering human signals by means of numerical calculations. Signals can begin from any substantial movement or state yet usually start from the face or hand. Current concentrations in the field incorporate feeling acknowledgment from face and hand signal acknowledgment. Clients can utilize basic motions to control or associate with gadgets without physically contacting them. Numerous methodologies have been made utilizing cameras and PC vision calculations to decipher gesture based communication. Notwithstanding, the recognizable proof and acknowledgment of stance, stride, proxemics, and human practices is additionally the subject of motion acknowledgment methods. Motion acknowledgment can be viewed as a route for PCs to start to comprehend human non-verbal communication, along these lines assembling a more extravagant scaffold among machines and people than crude content UIs or even GUIs (graphical UIs), which still cut off most of contribution to console and mouse and collaborate normally with no mechanical gadgets. Utilizing the idea of signal acknowledgment, it is conceivable to point a finger now will move as needs be. This could make customary contribution on gadgets such and even repetitive.



```
akash@akash-HP-Pavilion-Notebook:~/Documents/Sign-Language-master$ pip install opencv-python-3.4
Collecting opencv-python-3.4
  Could not find a version that satisfies the requirement opencv-python-3.4 (from versions: )
No matching distribution found for opencv-python-3.4
akash@akash-HP-Pavilion-Notebook:~/Documents/Sign-Language-master$ pip install --user opencv-python
Collecting opencv-python
  Using cached https://files.pythonhosted.org/packages/59/de/208f66a8f57a8b32536c5f7ca5e883cb15ddae8032164ea192fa103d50f6/opencv_python-4.0.0.21-cp35-
cp35m-manylinux1_x86_64.whl
Requirement already satisfied: numpy>=1.11.1 in /home/akash/.local/lib/python3.5/site-packages (from opencv-python) (1.16.2)
Installing collected packages: opencv-python
Successfully installed opencv-python-4.0.0.21
```

Figure 9 : Installation of OpenCV Using PIP

NumPy : NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.
It is a significant package for scientific computing with Python. It contains various features few

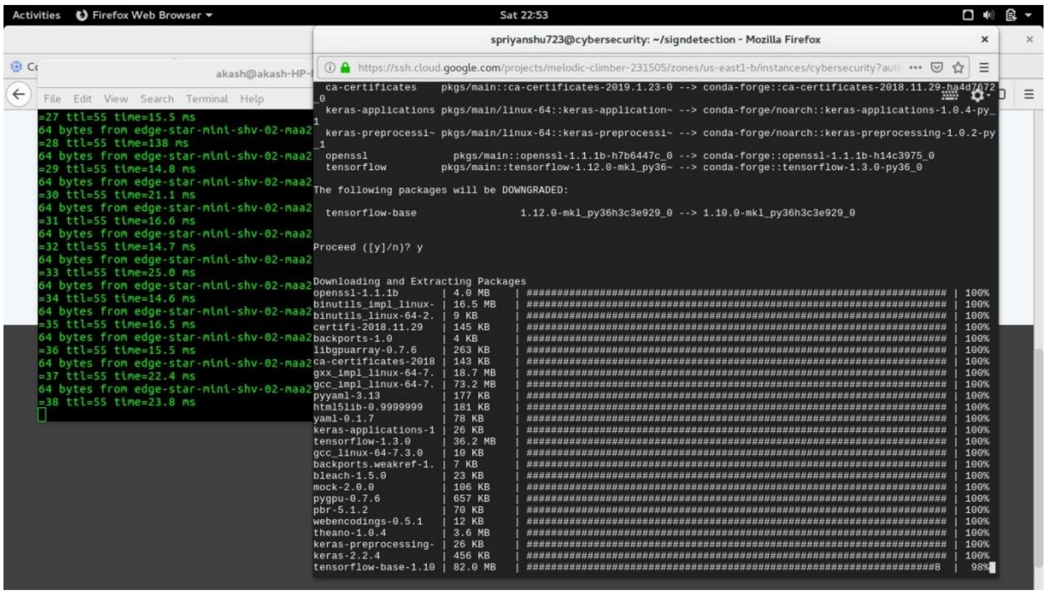
of which are:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows NumPy to seamlessly and speedily integrate with a wide variety of databases. The dataset is store in a .npy file, Numpy array format. There are two file, namely "X.npy" which has the feature vector, and "Y.npy" which has the classification vector.

Tensorflow : TensorFlow is an open source working environment for superior numerical calculation. Its adaptable design permits simple sending of calculation over an number of stages (CPUs, GPUs, TPUs), and from work areas to clusters of servers to portable and edge gadgets. Initially created by specialists and designers from the Google Brain group inside Google's AI association, it accompanies solid help for AI and deep learning and the adaptable numerical calculation center is across other numerous scientific domains. Building TensorFlow utilizing conda packages offers various advantages, including a total package provides the wider framework, more extensive stage support, an increasingly streamlined GPU experience, and better CPU execution. These packages are accessible by means of the Anaconda Repository, and building them is as simple as running "conda build tensorflow" or "conda build tensorflow-gpu" from an command line interface.

Keras : Keras being a high-level neural networks API, written in Python which is effective enough for running over Tensorflow, CNTK . It was created with an emphasis on empowering quick experimentation. Having the capacity to go from thought to result with least conceivable deferral is critical to doing great research. Keras takes into account simple and quick prototyping (through ease of use, measured quality, and extensibility) and it aids both convolution networks and recurrent networks, just as combination of the two and runs consistently on CPU and GPU.

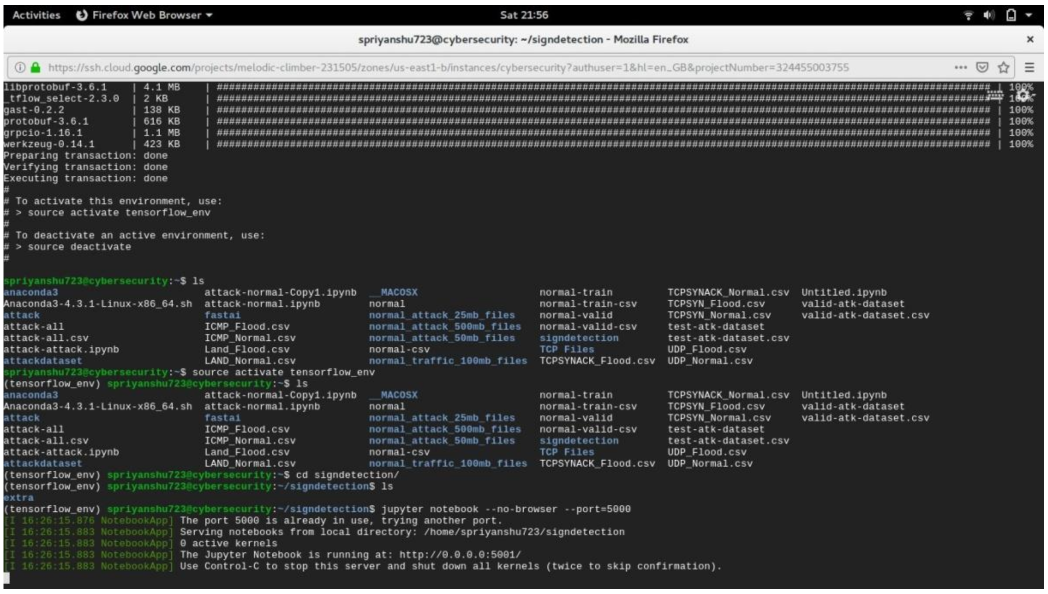

Activities
Firefox Web Browser
Sat 22:53
spriyanshu723@cybersecurity: ~/signdetection - Mozilla Firefox
https://ssh.cloud.google.com/projects/melodic-climber-231505/zones/us-east1-b/instances/cybersecurity?aut
akash@akash-HP-
File Edit View Search Terminal Help

```
=27 ttl=55 time=15.5 ms
64 bytes from edge-star-mini-shv-02-maa2
=28 ttl=55 time=138 ms
64 bytes from edge-star-mini-shv-02-maa2
=29 ttl=55 time=14.8 ms
64 bytes from edge-star-mini-shv-02-maa2
=30 ttl=55 time=21.1 ms
64 bytes from edge-star-mini-shv-02-maa2
=31 ttl=55 time=16.6 ms
64 bytes from edge-star-mini-shv-02-maa2
=32 ttl=55 time=14.7 ms
64 bytes from edge-star-mini-shv-02-maa2
=33 ttl=55 time=25.0 ms
64 bytes from edge-star-mini-shv-02-maa2
=34 ttl=55 time=14.6 ms
64 bytes from edge-star-mini-shv-02-maa2
=35 ttl=55 time=16.5 ms
64 bytes from edge-star-mini-shv-02-maa2
=36 ttl=55 time=15.5 ms
64 bytes from edge-star-mini-shv-02-maa2
=37 ttl=55 time=22.4 ms
64 bytes from edge-star-mini-shv-02-maa2
=38 ttl=55 time=23.8 ms
```

```
  ca-certificates    pkgs/main::ca-certificates-2019.1.23-0 --> conda-forge::ca-certificates-2018.11.29-ha4d7672
_0
  keras-applications pkgs/main/linux-64::keras-application~ --> conda-forge/noarch::keras-applications-1.0.4-py_
1
  keras-preprocessi~ pkgs/main/linux-64::keras-preprocessi~ --> conda-forge/noarch::keras-preprocessing-1.0.2-py
_1
  openssl             pkgs/main::openssl-1.1.1b-h7b6447c_0 --> conda-forge::openssl-1.1.1b-h14c3975_0
  tensorflow         pkgs/main::tensorflow-1.12.0-mkl_py36~ --> conda-forge::tensorflow-1.3.0-py36_0

The following packages will be DOWNGRADED:

  tensorflow-base                  1.12.0-mkl_py36h3c3e929_0 --> 1.10.0-mkl_py36h3c3e929_0


Proceed ([y]/n)? y


Downloading and Extracting Packages
openssl-1.1.1b       | 4.0 MB    | ###################################################################### | 100%
binutils_impl_linux- | 16.5 MB   | ###################################################################### | 100%
binutils_linux-64-2. | 9 KB      | ###################################################################### | 100%
certifi-2018.11.29   | 145 KB    | ###################################################################### | 100%
backports-1.0        | 4 KB      | ###################################################################### | 100%
libgpuarray-0.7.6    | 263 KB    | ###################################################################### | 100%
ca-certificates-2018 | 143 KB    | ###################################################################### | 100%
gxx_impl_linux-64-7. | 18.7 MB   | ###################################################################### | 100%
gcc_impl_linux-64-7. | 73.2 MB   | ###################################################################### | 100%
pyyaml-3.13          | 177 KB    | ###################################################################### | 100%
html5lib-0.9999999   | 181 KB    | ###################################################################### | 100%
yaml-0.1.7           | 78 KB     | ###################################################################### | 100%
keras-applications-1 | 26 KB     | ###################################################################### | 100%
tensorflow-1.3.0     | 36.2 MB   | ###################################################################### | 100%
gcc_linux-64-7.3.0   | 10 KB     | ###################################################################### | 100%
backports.weakref-1. | 7 KB      | ###################################################################### | 100%
bleach-1.5.0         | 23 KB     | ###################################################################### | 100%
mock-2.0.0           | 106 KB    | ###################################################################### | 100%
pygpu-0.7.6          | 657 KB    | ###################################################################### | 100%
pbr-5.1.2            | 70 KB     | ###################################################################### | 100%
webencodings-0.5.1   | 12 KB     | ###################################################################### | 100%
theano-1.0.4         | 3.6 MB    | ###################################################################### | 100%
keras-preprocessing- | 26 KB     | ###################################################################### | 100%
keras-2.2.4          | 456 KB    | ###################################################################### | 100%
tensorflow-base-1.10 | 82.0 MB   | #####################################################################8  |  98%
```

Figure 10 : Keras Set Up

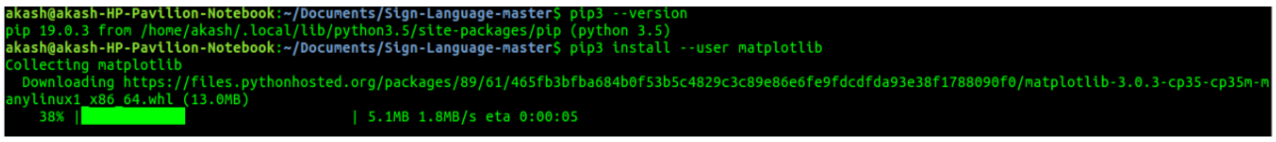
Anaconda and Jupyter : Anaconda is package manager. Jupyter is a presentation layer. The Jupyter Notebook being an open-source web based technology that helps us to build and exchange archives that contain running codes, equations, visualizations and account content. It incorporates data cleaning and change, numerical simulation, factual demonstrating, data representation, machine learning, and significantly more. Anaconda constrictor attempts to unravel the dependency hell in python—where distinctive ventures have diverse reliance versions—to not influence diverse task conditions to require distinctive versions, which may interfere with one another.


Activities
Firefox Web Browser
Sat 21:56
spriyanshu723@cybersecurity: ~/signdetection - Mozilla Firefox
https://ssh.cloud.google.com/projects/melodic-climber-231505/zones/us-east1-b/instances/cybersecurity?authuser=1&hl=en_GB&projectNumber=324455003755

```
libprotobuf-3.6.1    | 4.1 MB    | ########################################################################################## | 100%
_tflow_select-2.3.0  | 2 KB      | ########################################################################################## | 100%
gast-0.2.2           | 138 KB    | ########################################################################################## | 100%
protobuf-3.6.1       | 616 KB    | ########################################################################################## | 100%
grpcio-1.16.1        | 1.1 MB    | ########################################################################################## | 100%
werkzeug-0.14.1      | 423 KB    | ########################################################################################## | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use:
# > source activate tensorflow_env
#
# To deactivate an active environment, use:
# > source deactivate
#

spriyanshu723@cybersecurity:~$ ls
anaconda3                     attack-normal-Copy1.ipynb  __MACOSX                     normal-train          TCPSYNACK_Normal.csv  Untitled.ipynb
Anaconda3-4.3.1-Linux-x86_64.sh  attack-normal.ipynb     normal                       normal-train-csv      TCPSYN_Flood.csv      valid-atk-dataset
attack                        fastai                     normal_attack_25mb_files     normal-valid          TCPSYN_Normal.csv     valid-atk-dataset.csv
attack-all                    ICMP_Flood.csv             normal_attack_500mb_files    normal-valid-csv      test-atk-dataset
attack-all.csv                ICMP_Normal.csv            normal_attack_50mb_files     signdetection         test-atk-dataset.csv
attack-attack.ipynb           Land_Flood.csv             normal-csv                   TCP Files             UDP_Flood.csv
attackdataset                 LAND_Normal.csv            normal_traffic_100mb_files   TCPSYNACK_Flood.csv   UDP_Normal.csv
spriyanshu723@cybersecurity:~$ source activate tensorflow_env
(tensorflow_env) spriyanshu723@cybersecurity:~$ ls
anaconda3                     attack-normal-Copy1.ipynb  __MACOSX                     normal-train          TCPSYNACK_Normal.csv  Untitled.ipynb
Anaconda3-4.3.1-Linux-x86_64.sh  attack-normal.ipynb     normal                       normal-train-csv      TCPSYN_Flood.csv      valid-atk-dataset
attack                        fastai                     normal_attack_25mb_files     normal-valid          TCPSYN_Normal.csv     valid-atk-dataset.csv
attack-all                    ICMP_Flood.csv             normal_attack_500mb_files    normal-valid-csv      test-atk-dataset
attack-all.csv                ICMP_Normal.csv            normal_attack_50mb_files     signdetection         test-atk-dataset.csv
attack-attack.ipynb           Land_Flood.csv             normal-csv                   TCP Files             UDP_Flood.csv
attackdataset                 LAND_Normal.csv            normal_traffic_100mb_files   TCPSYNACK_Flood.csv   UDP_Normal.csv
(tensorflow_env) spriyanshu723@cybersecurity:~$ cd signdetection/
(tensorflow_env) spriyanshu723@cybersecurity:~/signdetection$ ls
extra
(tensorflow_env) spriyanshu723@cybersecurity:~/signdetection$ jupyter notebook --no-browser --port=5000
[I 16:26:15.876 NotebookApp] The port 5000 is already in use, trying another port.
[I 16:26:15.883 NotebookApp] Serving notebooks from local directory: /home/spriyanshu723/signdetection
[I 16:26:15.883 NotebookApp] 0 active kernels
[I 16:26:15.883 NotebookApp] The Jupyter Notebook is running at: http://0.0.0.0:5001/
[I 16:26:15.883 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
```

Figure 11 : Environment Set Up (Anaconda and Jupyter)

PIP - It is a package management system used to install and manage software packages/libraries written in Python. In particular pip has an feature to oversee full lists of packages and comparing form numbers, conceivable through a "requirement" file. This allows the effective re-formation of a whole groups of packages in a different environment (for example another PC) or virtual environment.



```
akash@akash-HP-Pavilion-Notebook:~/Documents/Sign-Language-master$ pip3 --version
pip 19.0.3 from /home/akash/.local/lib/python3.5/site-packages/pip (python 3.5)
akash@akash-HP-Pavilion-Notebook:~/Documents/Sign-Language-master$ pip3 install --user matplotlib
Collecting matplotlib
  Downloading https://files.pythonhosted.org/packages/89/61/465fb3bfba684b0f53b5c4829c3c89e86e6fe9fdcdfda93e38f1788090f0/matplotlib-3.0.3-cp35-cp35m-m
anylinux1_x86_64.whl (13.0MB)
    38% |████████████                    | 5.1MB 1.8MB/s eta 0:00:05
```

Figure 12 : Installation Of Matplotlib Using PIP

h5py - The h5py package is a Pythonic interface to the HDF5 binary data format. A HDF5 file is a holder for two sorts of objects: datasets, which are array like accumulations of data, and groups, which are folder like compartments that hold datasets and different groups. The most principal thing to recollect when utilizing h5py is: Groups work like dictionaries , and datasets work like NumPy arrays. The best feature of HDF5 is that one can store metadata directly alongside the data it describe. It lets you store tremendous measures of numerical data, and effectively control that data from NumPy. For instance, you can cut into multi-terabyte datasets put away on disk, as though they were real NumPy arrays. A great many datasets can be put away in a single file, ordered and labelled anyway you need.H5py utilizes direct NumPy and Python metaphors, similar to dictionary and NumPy array syntax. Best of all, the files you make are in a widely utilized standard binary format , which you can trade with other individuals, including the individuals who use programs like IDL and MATLAB.

Scikit-learn - Simple and efficient tools for data mining and data analysis. Scikit-learn gives a scope of supervised and unsupervised learning algorithms through a consistent interface in Python. The library is based upon the SciPy (Scientific Python) that must be introduced before you can utilize scikit-learn. This stack includes :

NumPy: Base n-dimensional array package

SciPy: Fundamental library for scientific computing

Matplotlib: Comprehensive 2D/3D plotting

IPython: Enhanced interactive console

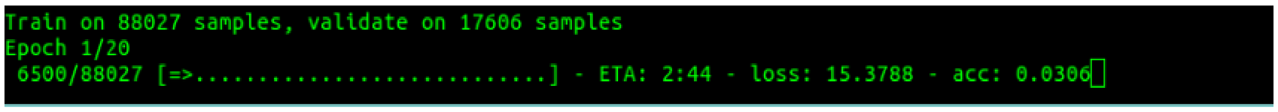
Sympy: Symbolic mathematics

Pandas: Data structures and analysis

Extensions or modules for SciPy care conventionally named SciKits. As such, the module provides learning algorithms and is named scikit-learn.Open to everyone, and reusable in different contexts . It is built on NumPy, SciPy, and matplotlib, Open source, commercially usable - BSD license.

pyttsx - This gives you a chance to convert text in to audio you can hear. This package works in Windows, Mac, and Linux. It utilizes local speech drivers when accessible and works totally disconnected. pyttsx is completely offline and works seemlesly and has multiple tts-engine support.

Epoch : An epoch is a single step in traing a neural network; as it were the point at which a neural network is trained on each training tests just in one pass we can state that epoch is completed. So training procedure may comprise multiple epochs.



```
Train on 88027 samples, validate on 17606 samples
Epoch 1/20
 6500/88027 [=>...........................] - ETA: 2:44 - loss: 15.3788 - acc: 0.0306
```

Figure 13 :Epoch

Softmax : The softmax function is used in the final and last layer of any neural network-based classifier. Used for the multi-classification task.

Adam Optimizer : Adam is an optimization method that can preferred over other ancient stochastic gradient descent method to update network weights iterative situated in training data. Moderately low memory prerequisites (however higher than gradient descent and gradient descent with momentum). Appropriate for issues that are large as far as data and additionally parameters. Computationally effective .

Threshold - Threshold value is used to classify the pixel values. OpenCV provides different styles of thresholding and it is decided by the fourth parameter of the function. Here we have used Otsu's thresholding after gaussain filter.

pickle - Python pickle module is utilized for serializing and de-serializing a Python object structure. What pickle does is that it "serializes" the item first before composing it to document. Pickling is an approach to change over a python object (list, dict, and so on.) into a character stream. The thought is that this character stream contains all the data important to recreate the article in another python content.

Pickle monitors the items it has just serialized, so later references to a similar article won't be serialized once more.

User defined classes and their occurrences

Object sharing (references to a similar item in better places)

4.2 CNN (Convolution Neural Network)

Neural networks, as its name proposes, is an machine learning method which is designed according to the brain structure. It involves of a network of learning units called neurons. These neurons figure out how to change over input signals (for example picture of a hand) into relating yield signals (for example the mark "hand"), shaping the basis of automated recognition.

A convolution neural network (CNN, or ConvNet) is a sort of feed-forward artificial neural network in which the availability design between its neurons is inspired by the association of the animal visual cortex. CNNs have repetitive blocks of neurons that are connected crosswise over space (for pictures) or time (for sound signs and so on). For pictures, these blocks of neurons can be translated as 2D convolution parts, more than once connected over each fix of the picture. For speech, they can be viewed as the 1D convolution kernels connected crosswise over time - window. At preparing time, the weights for these repeated blocks are 'shared', for example the weight gradients learned over different picture patches are averaged.

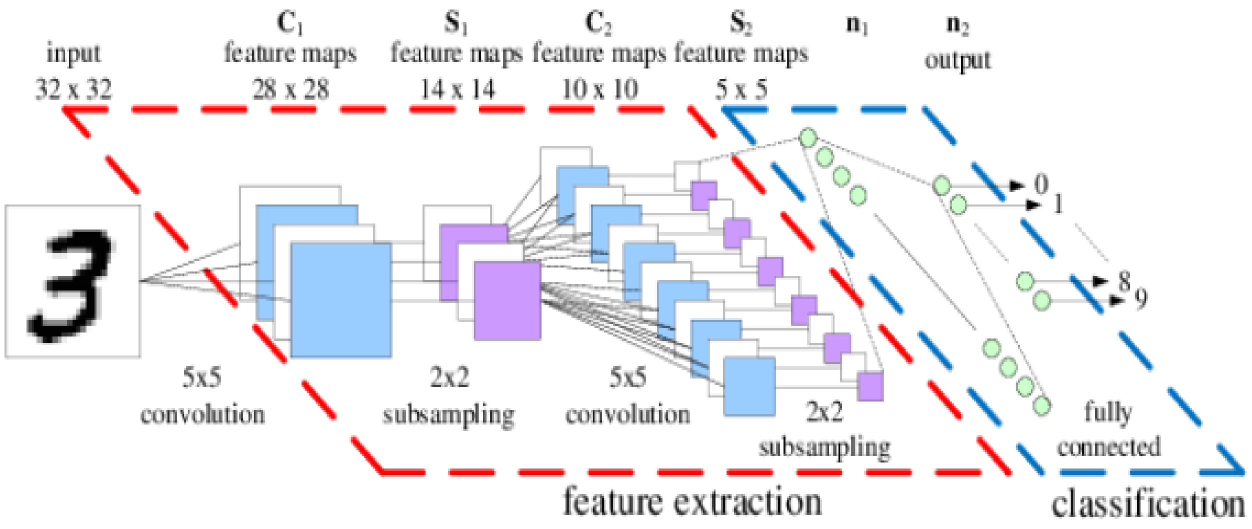


Figure 14 : Convolution Neural Network

Four major Steps Involved in CNN

There are four fundamental steps in CNN: Convolution, Subsampling , Activation and full Connectedness.

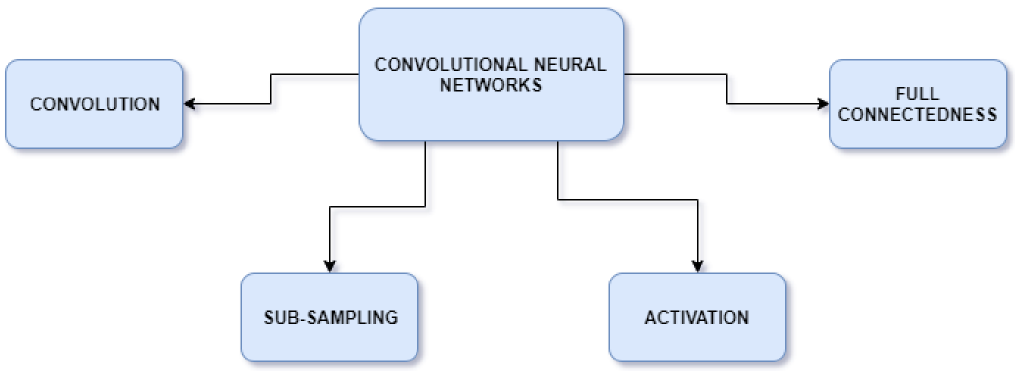


Figure 15 : Major Steps in CNN

4.2.1 Convolution

The first layers that get an input signal are called convolution filters. Convolution is where the system attempts to label the input signal by referring to what it has realized before. On the off chance that the input signal resembles past hand pictures it has seen previously, the "hand" reference signal will be mixed into, or convolved with, the input signal. The output signal at that point passed on to the following layer.

Convolution has the pleasant property of being translational invariant . Intuitively , this implies every convolution filter represents a feature of interest (e.g finger), and the CNN algorithm realizes which features contain the subsequent reference (for example hand).

The output signal quality isn't reliant on where the features are found, yet essentially whether the features are available. Subsequently, a hand could be in various positions, and the CNN algorithm would in any case have the capacity to remember it. For e.g assume we convolve a 32x32x3 (32x32 picture with 3 channels R,G ,B) with a 5x5x3 filter. We take the 5*5*3 filter and slide it over the whole picture and en route take the dot product between the filter and pieces of the input picture. The convolution layer is the primary structure block of a convolution neural network. The convolution layer includes a lot of independent filters. Each filter is independently convolved with the picture. The CNN may comprises of a few Convolution layers every one of which can have comparative or distinctive number of independent filters.

Each of these filters are introduced randomly and become our parameters which will be learned by the network subsequently.

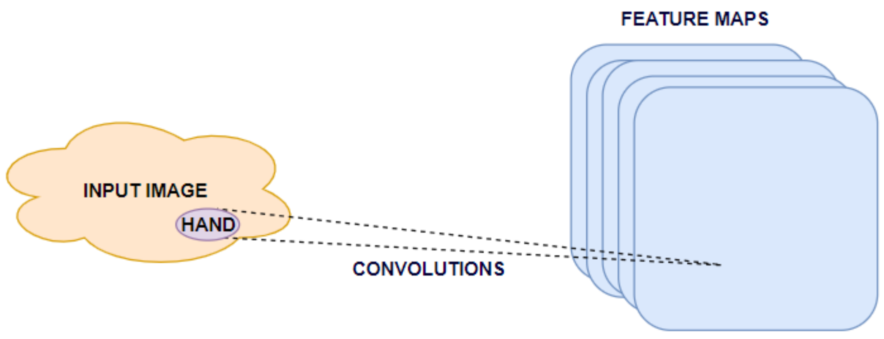


Figure 16 : Convolution Layer

4.2.2 Subsampling or Pooling

A pooling or subsampling layer frequently quickly pursues a convolution layer in CNN. Its job is to downsample the yield of a convolution layer along both the spatial components of height and width. The pooling and subsampling hidden units. Pooling fundamentally takes a lot of hidden units in a feature map and somehow aggregates the activations to obtain a single number.

Pooling layers segment would reduce the quantity of parameters when the images are excessively substantial. Spatial pooling likewise called subsampling or downsampling which lessens the dimensionality of each guide however holds the critical data. Spatial pooling can be of various sorts:

1. Max pooling take the greatest element from the rectified feature map. Taking the biggest component could likewise take the average pooling. Aggregate of all components in the element map call as sum pooling.

Spatial Pooling (likewise called subsampling or downsampling) lessens the dimensionality of each element map however holds the most imperative data. Spatial Pooling can be of various sorts: Max, Average, Sum and so forth.

If there arises an occurrence of Max Pooling, we characterize a spatial neighbourhood (for instance, a 2×2 window) and take the biggest component from the amended element map inside that window. Rather than taking the biggest component we could likewise take the normal (Average Pooling) or sum of all components in that window. Overall , Max Pooling works better.

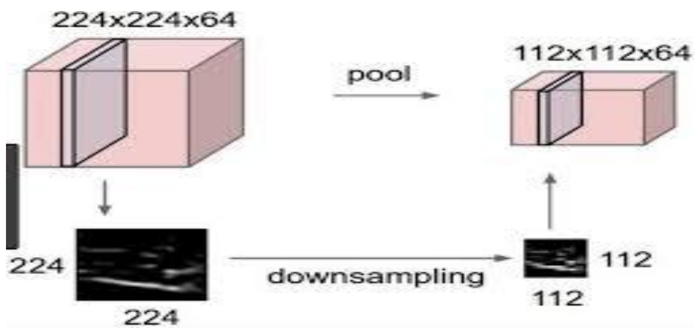


Figure 17 : Pooling Layer

4.2.3. Activation

Activation layer is only the output of the function. You will feed contribution to one of the activation capacity and it will give one output, and this activity we call as layer. Yes, just like we have distinctive functions in mathematics, you feed some matrix or values to a function and it will give output.

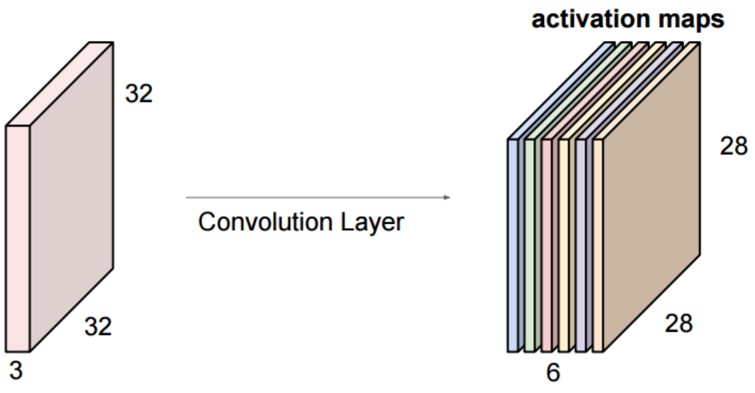


Figure 18 : Activation Maps

4.2.4 Fully Connectedness

Convolution layer help in separating highlights. The output layer in a CNN as referenced earlier is a complete connected layer, where the input taken from other layers are flattened out and sent so as to transform the output into a number of classes as required by the network.

The Fully Connected layer is a conventional Multi Layer Perceptron that utilizes a softmax activation work in the output layer (different classifiers like SVM can likewise be utilized, yet will stick to softmax in this post). The expression "Completely Connected" suggests that each neuron in the past layer is associated with each neuron on the following layer.

The output from the convolution and pooling layers represent high-level features of the input image. The motivation behind the Fully Connected layer is to utilize these highlights for arranging the input image into different classes dependent on the training dataset.

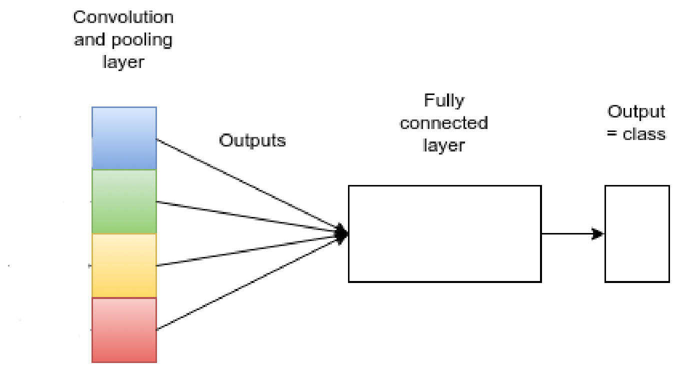
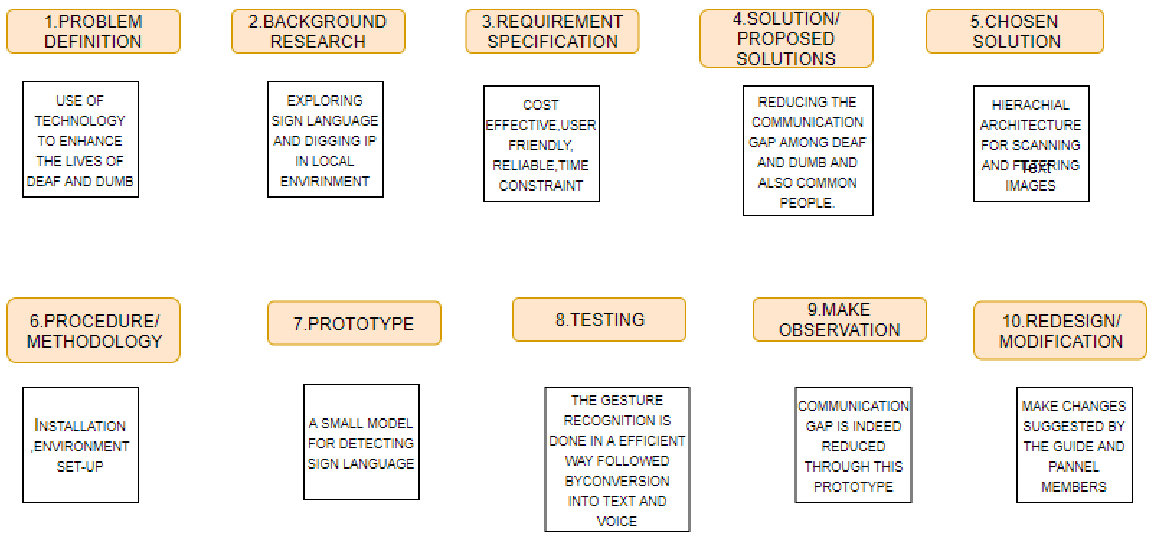


Figure 19 : Fully Connectedness Layer

5. DESIGN APPROACH AND DETAILS

5.1 Engineering Design



1.PROBLEM DEFINITION
USE OF TECHNOLOGY TO ENHANCE THE LIVES OF DEAF AND DUMB
2.BACKGROUND RESEARCH
EXPLORING SIGN LANGUAGE AND DIGGING IP IN LOCAL ENVIRINMENT
3.REQUIREMENT SPECIFICATION
COST EFFECTIVE,USER FRIENDLY, RELIABLE,TIME CONSTRAINT
4.SOLUTION/ PROPOSED SOLUTIONS
REDUCING THE COMMUNICATION GAP AMONG DEAF AND DUMB AND ALSO COMMON PEOPLE.
5.CHOSEN SOLUTION
HIERACHIAL ARCHITECTURE FOR SCANNING AND FILTERING IMAGES
6.PROCEDURE/ METHODOLOGY
INSTALLATION ,ENVIRONMENT SET-UP
7.PROTOTYPE
A SMALL MODEL FOR DETECTING SIGN LANGUAGE
8.TESTING
THE GESTURE RECOGNITION IS DONE IN A EFFICIENT WAY FOLLOWED BYCONVERSION INTO TEXT AND VOICE
9.MAKE OBSERVATION
COMMUNICATION GAP IS INDEED REDUCED THROUGH THIS PROTOTYPE
10.REDESIGN/ MODIFICATION
MAKE CHANGES SUGGESTED BY THE GUIDE AND PANNEL MEMBERS

Figure 20 : Engineering Design

The fundamental steps involved in the engineering design are as follows:

1. Problem Definition : The goal of the project was to use of best technology present in a efficient way to enhance the lives of the deaf and dumb .

2. Background Research : Exploration of various sign languages , digging up in the local environment . Exploring various algorithms which would be feasible to achieve the goal.

3.Requirement Specification : A cost effective , user friendly , reliable which involves the latest trends and new technology which the need of our.

4.Proposed Solution : The proposed solution is to reduce the communication gap among deaf and dumb and also the common people .

5.Selected Solution : The selected solution is a hierarchical architecture for scanning and filtering images as well as gesture recognition termed as CNN.

6.Methodology : Installation of various packages , Environment set up according to feasibility then training , testing and recognition .

7.Prototype : Building a small model for sign language detection deployed on cloud environment through which gesture recognition takes place which include numeric and alphabets.

8.Testing : The gesture recognition is done in a efficient way which is further followed by text conversion and then into voice.

9.Make Observations : Communication gap is indeed reduce through this prototype.

10.Modification : Make changes suggested by the guide and panel members.

5.2 Design Approach

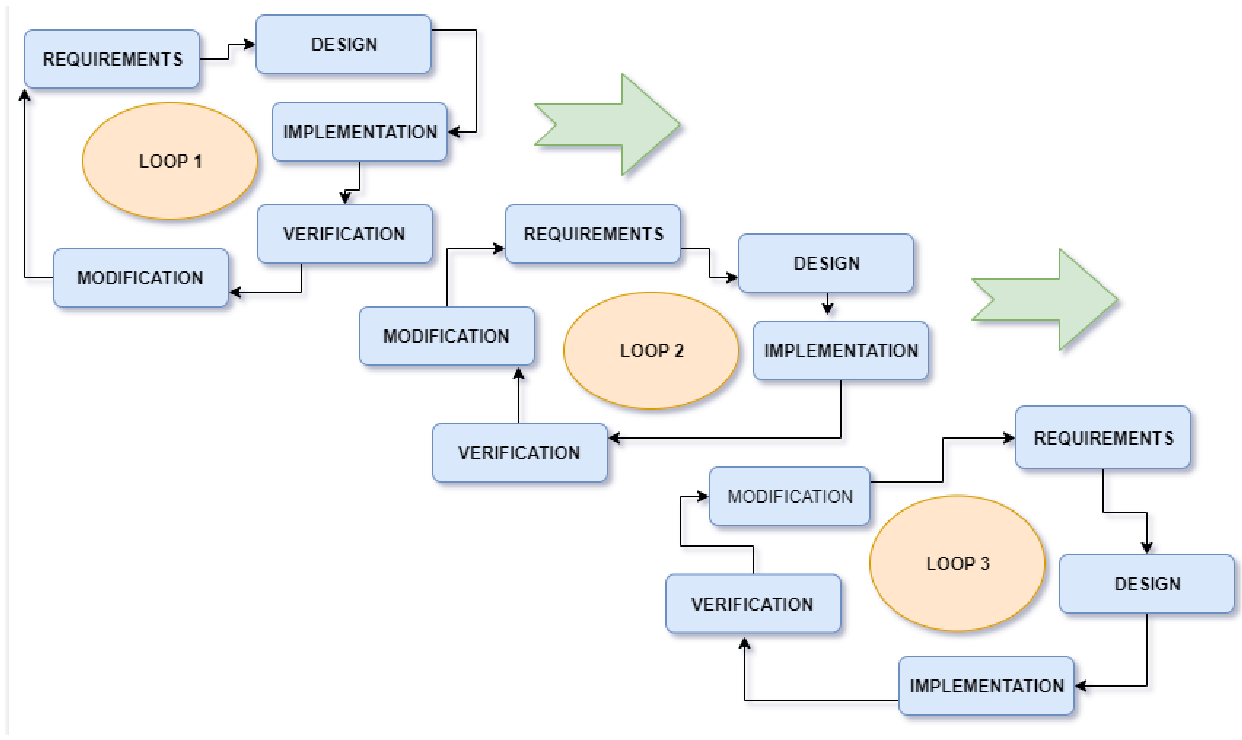


Figure 21 : Design Approach

In order to achieve a particular goal there are several steps which needs to be taken care in the project.

Firstly we need to focus on the requirement then its design then implement the prototype, verify it using its constraints as well then modify it and we keep following the above steps until the desired goal is achieved.

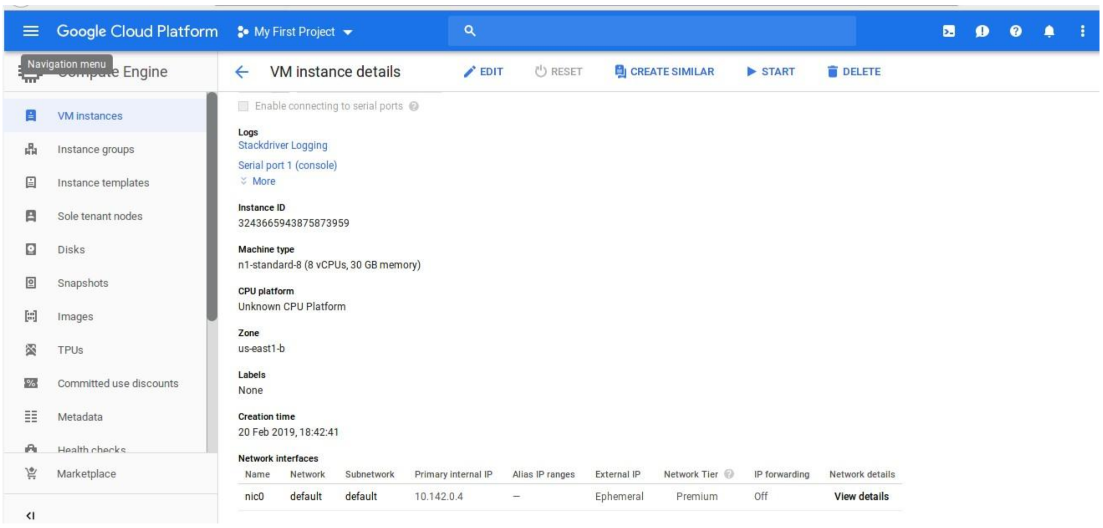
We have to keep analysing each modules/task undertaken whether it meets the goal or is one step closer to the goal. So the above picture is the diagrammatic view of our prototype.

6. Codes and Results

On working in local environment, training and testing of the huge data set requires enormous resources and requires ample of hours so usage of certain technologies in the cloud environment enhances the overall performance, accuracy and provide user friendly environment .

We are using Google Cloud Platform (GCP) for implementing our project in cloud environment.

- On fulfilling all the credentials we further proceed in making a VM instance.



Enable connecting to serial ports

Logs
Stackdriver Logging
Serial port 1 (console)
More

Instance ID
3243665943875873959

Machine type
n1-standard-8 (8 vCPUs, 30 GB memory)

CPU platform
Unknown CPU Platform

Zone
us-east1-b

Labels
None

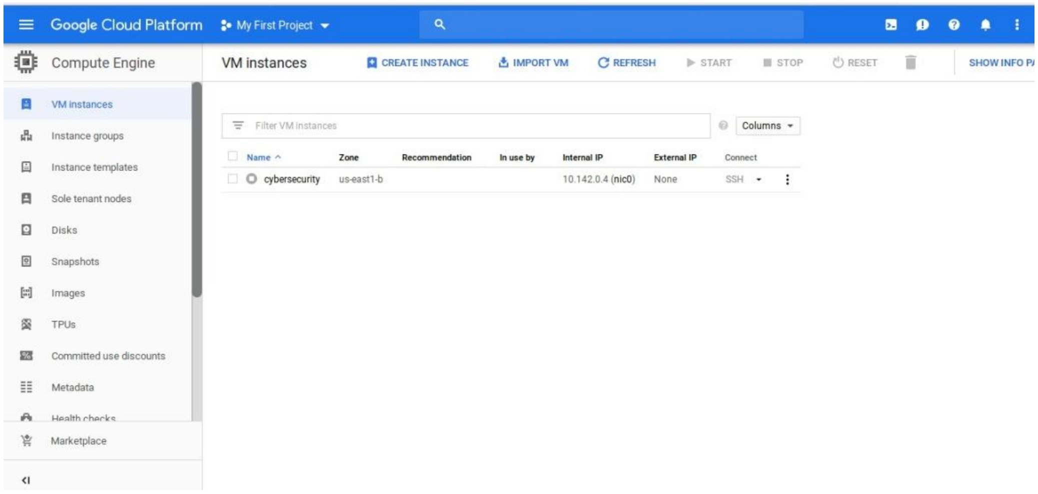
Creation time
20 Feb 2019, 18:42:41

Network interfaces

Name	Network	Subnetwork	Primary internal IP	Alias IP ranges	External IP	Network Tier	IP forwarding	Network details
nic0	default	default	10.142.0.4	–	Ephemeral	Premium	Off	View details

Figure 22 : VM Instance

We pushed the entire data on github repository and deployed it back on the VM instance in the cloud environment.



Name	Zone	Recommendation	In use by	Internal IP	External IP	Connect
cybersecurity	us-east1-b			10.142.0.4 (nic0)	None	SSH

Figure 23 : Refreshing IP Address

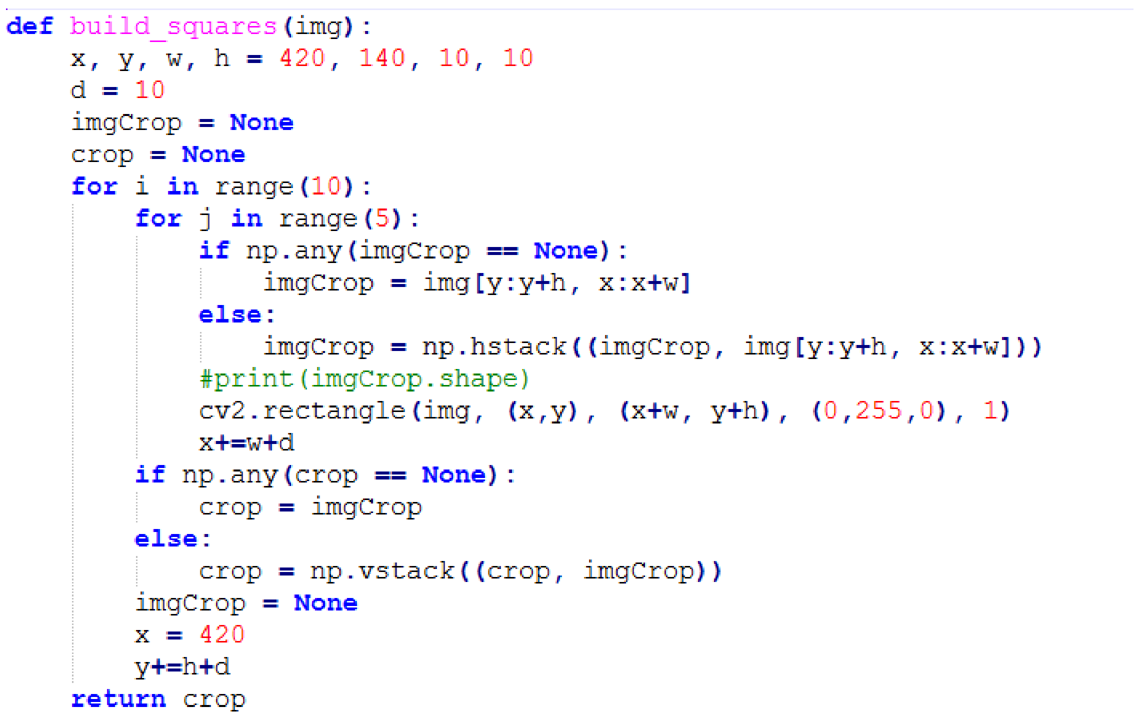
Set Hand Histogram

A histogram is a graph or a plot that speaks to the dispersion of the pixel intensities in an image. Figuring the histogram of a picture is exceptionally valuable as it gives an instinct with respect to certain properties of the picture, for example, the tonal range, the contrast and the brightness. Set histogram peruses input pictures and convert those pictures to HSV shading space (Hue, Saturation, Value).

Smoothing, additionally called blurring, is a basic and every now and again utilized picture processing task. To carry on smoothing process we will apply Gaussian filter to our image.

In image processing, a Gaussian blur (otherwise called Gaussian smoothing) is the consequence of obscuring a picture by a Gaussian capacity . It is a broadly utilized impact in designs programming, ordinarily to decrease picture clamour and lessen detail.

This is exceptionally successful in removing salt-and-pepper noise. The Median filter is a typical procedure for smoothing. Median smoothening is generally used in edge identification calculations on the grounds that under specific conditions, it preserves edges while removing noise.



```python
def build_squares(img):
    x, y, w, h = 420, 140, 10, 10
    d = 10
    imgCrop = None
    crop = None
    for i in range(10):
        for j in range(5):
            if np.any(imgCrop == None):
                imgCrop = img[y:y+h, x:x+w]
            else:
                imgCrop = np.hstack((imgCrop, img[y:y+h, x:x+w]))
            #print(imgCrop.shape)
            cv2.rectangle(img, (x,y), (x+w, y+h), (0,255,0), 1)
            x+=w+d
        if np.any(crop == None):
            crop = imgCrop
        else:
            crop = np.vstack((crop, imgCrop))
        imgCrop = None
        x = 420
        y+=h+d
    return crop
```

```
def get_hand_hist():
    cam = cv2.VideoCapture(1)
    if cam.read()[0]==False:
        cam = cv2.VideoCapture(0)
    x, y, w, h = 300, 100, 300, 300
    flagPressedC, flagPressedS = False, False
    imgCrop = None
    while True:
        img = cam.read()[1]
        img = cv2.flip(img, 1)
        hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)

        keypress = cv2.waitKey(1)
        if keypress == ord('c'):
            hsvCrop = cv2.cvtColor(imgCrop, cv2.COLOR_BGR2HSV)
            flagPressedC = True
            hist = cv2.calcHist([hsvCrop], [0, 1], None, [180, 256], [0, 180, 0, 256])
            cv2.normalize(hist, hist, 0, 255, cv2.NORM_MINMAX)
        elif keypress == ord('s'):
            flagPressedS = True
            break
        if flagPressedC:
            dst = cv2.calcBackProject([hsv], [0, 1], hist, [0, 180, 0, 256], 1)
            dst1 = dst.copy()
            disc = cv2.getStructuringElement(cv2.MORPH_ELLIPSE,(10,10))
            cv2.filter2D(dst,-1,disc,dst)
            blur = cv2.GaussianBlur(dst, (11,11), 0)
            blur = cv2.medianBlur(blur, 15)
            ret,thresh = cv2.threshold(blur,0,255,cv2.THRESH_BINARY+cv2.THRESH_OTSU)
            thresh = cv2.merge((thresh,thresh,thresh))
            #cv2.imshow("res", res)
            cv2.imshow("Thresh", thresh)
```

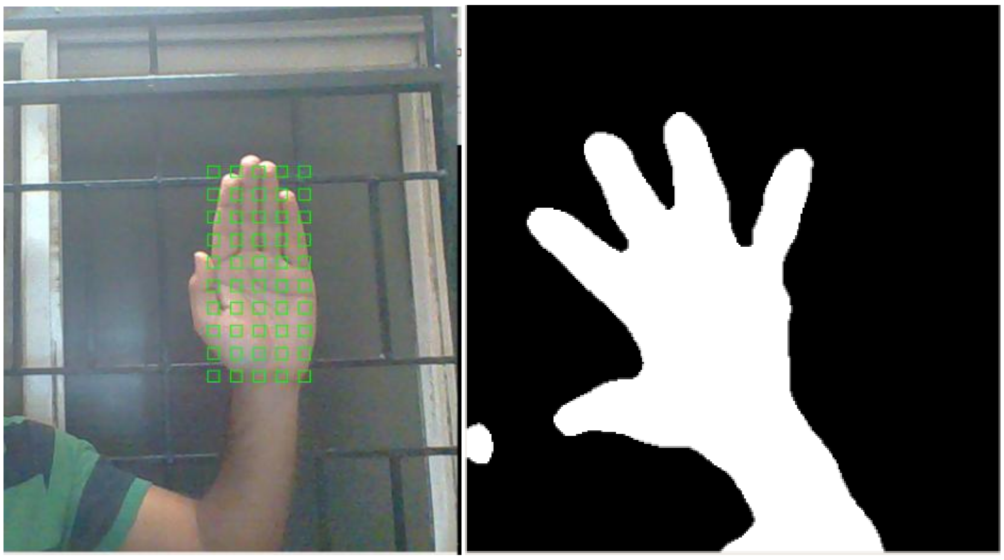
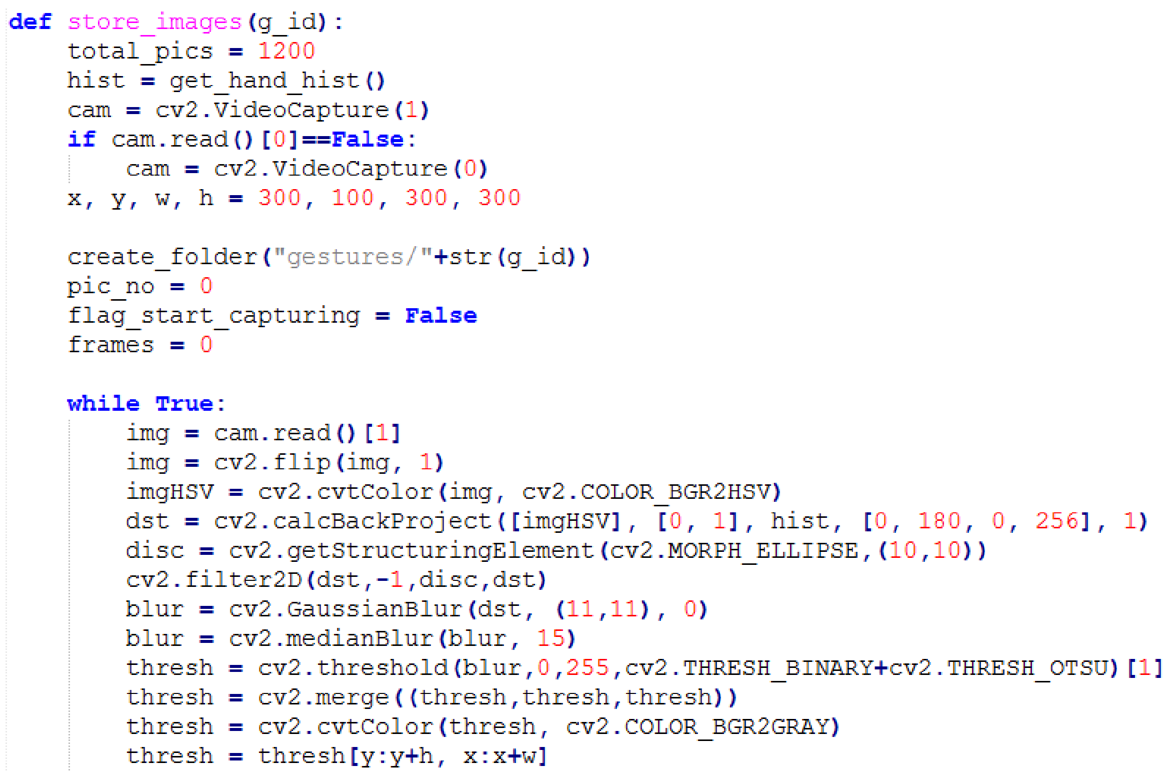


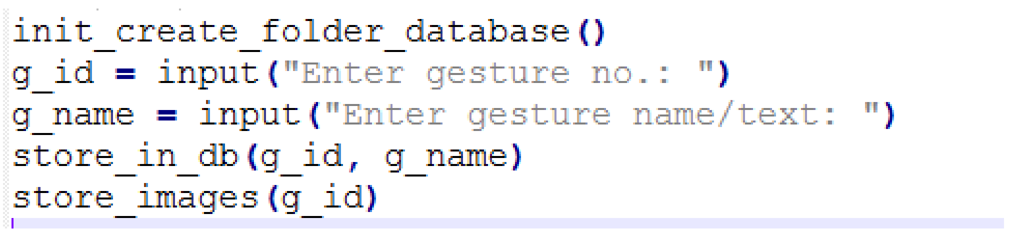
Figure 24 : Histogram Set Up

Create Gestures

Create gestures can be used to create new gestures. The create gesture is used to create the gesture database and save the converted images in a folder. The hsv image database will be used to train the convolution neural network model.



```python
def store_images(g_id):
    total_pics = 1200
    hist = get_hand_hist()
    cam = cv2.VideoCapture(1)
    if cam.read()[0]==False:
        cam = cv2.VideoCapture(0)
    x, y, w, h = 300, 100, 300, 300

    create_folder("gestures/"+str(g_id))
    pic_no = 0
    flag_start_capturing = False
    frames = 0

    while True:
        img = cam.read()[1]
        img = cv2.flip(img, 1)
        imgHSV = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
        dst = cv2.calcBackProject([imgHSV], [0, 1], hist, [0, 180, 0, 256], 1)
        disc = cv2.getStructuringElement(cv2.MORPH_ELLIPSE,(10,10))
        cv2.filter2D(dst,-1,disc,dst)
        blur = cv2.GaussianBlur(dst, (11,11), 0)
        blur = cv2.medianBlur(blur, 15)
        thresh = cv2.threshold(blur,0,255,cv2.THRESH_BINARY+cv2.THRESH_OTSU)[1]
        thresh = cv2.merge((thresh,thresh,thresh))
        thresh = cv2.cvtColor(thresh, cv2.COLOR_BGR2GRAY)
        thresh = thresh[y:y+h, x:x+w]
```

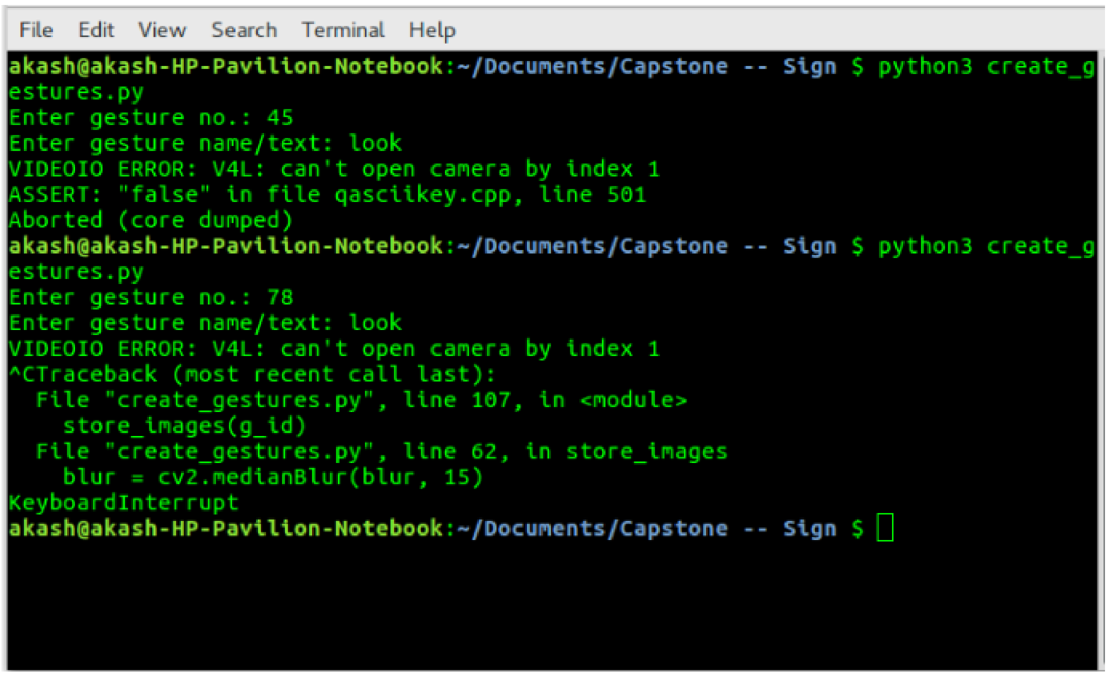


```python
init_create_folder_database()
g_id = input("Enter gesture no.: ")
g_name = input("Enter gesture name/text: ")
store_in_db(g_id, g_name)
store_images(g_id)
```

Flip Images

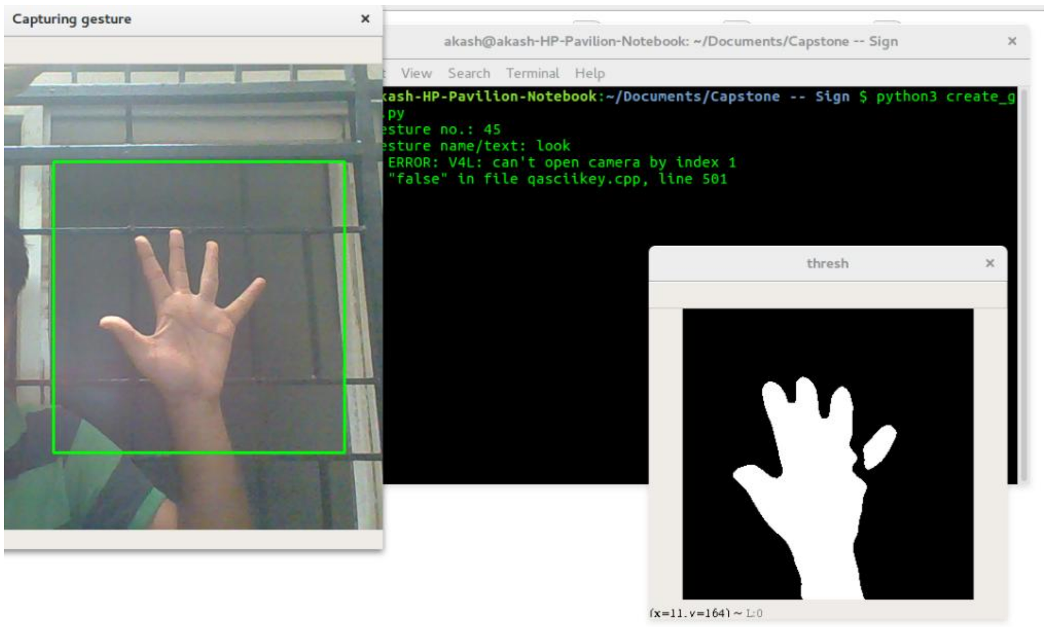
flip - Flip the image vertically to avoid the confusion between different hand gestures. It is utilized for serializing and de-serializing a Python object structure. What pickle does is that it "serializes" the article first before composing it to document. Pickling is an approach to change over a python object (list, dict. , and so forth.)

```
import cv2, os

def flip_images():
    gest_folder = "gestures"
    images_labels = []
    images = []
    labels = []
    for g_id in os.listdir(gest_folder):
        for i in range(1200):
            path = gest_folder+"/"+g_id+"/"+str(i+1)+".jpg"
            new_path = gest_folder+"/"+g_id+"/"+str(i+1+1200)+".jpg"
            print(path)
            img = cv2.imread(path, 0)
            img = cv2.flip(img, 1)
            cv2.imwrite(new_path, img)

flip_images()
```



```
akash@akash-HP-Pavilion-Notebook:~/Documents/Capstone -- Sign $ python3 create_g
estures.py
Enter gesture no.: 45
Enter gesture name/text: look
VIDEOIO ERROR: V4L: can't open camera by index 1
ASSERT: "false" in file qasciikey.cpp, line 501
Aborted (core dumped)
akash@akash-HP-Pavilion-Notebook:~/Documents/Capstone -- Sign $ python3 create_g
estures.py
Enter gesture no.: 78
Enter gesture name/text: look
VIDEOIO ERROR: V4L: can't open camera by index 1
^CTraceback (most recent call last):
  File "create_gestures.py", line 107, in <module>
    store_images(g_id)
  File "create_gestures.py", line 62, in store_images
    blur = cv2.medianBlur(blur, 15)
KeyboardInterrupt
akash@akash-HP-Pavilion-Notebook:~/Documents/Capstone -- Sign $ 
```

Figure 25 : Executing File To Create Gestures


Capturing gesture
akash@akash-HP-Pavilion-Notebook: ~/Documents/Capstone -- Sign
View Search Terminal Help
thresh
(x=11, y=164) ~ L:0

```
kash-HP-Pavilion-Notebook:~/Documents/Capstone -- Sign $ python3 create_g
.py
esture no.: 45
esture name/text: look
 ERROR: V4L: can't open camera by index 1
 "false" in file qasciikey.cpp, line 501
```

Figure 26 : Creating Gesture I

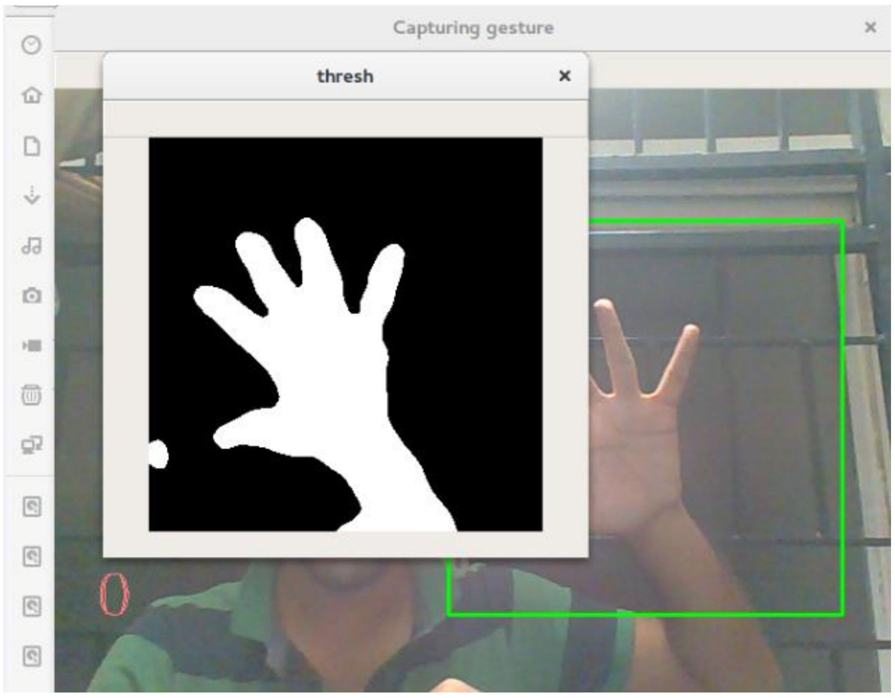
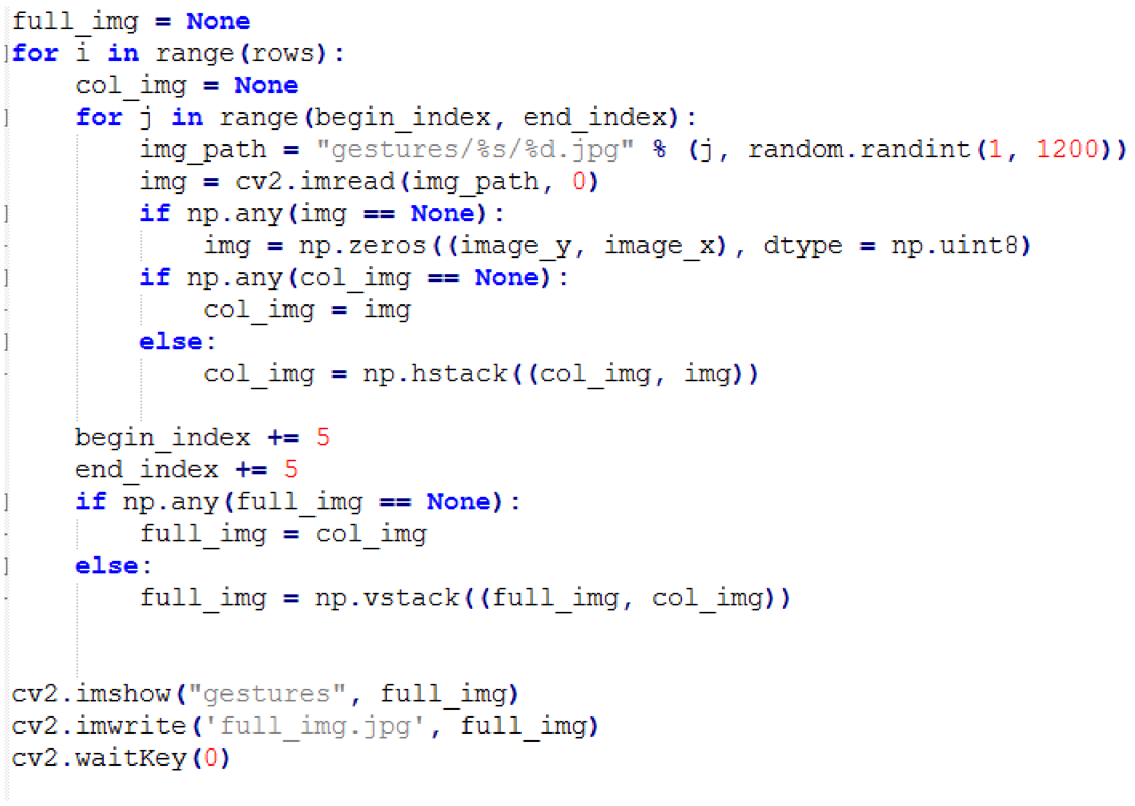
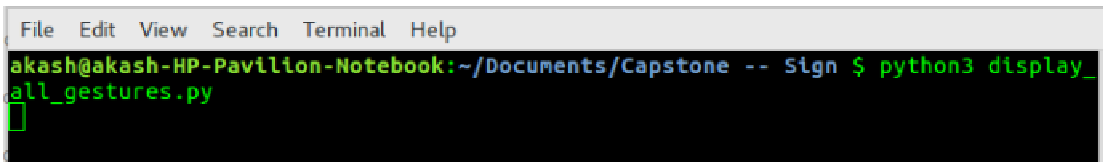


Figure 27 : Creating Gesture II

Display All Gestures



```
full_img = None
for i in range(rows):
    col_img = None
    for j in range(begin_index, end_index):
        img_path = "gestures/%s/%d.jpg" % (j, random.randint(1, 1200))
        img = cv2.imread(img_path, 0)
        if np.any(img == None):
            img = np.zeros((image_y, image_x), dtype = np.uint8)
        if np.any(col_img == None):
            col_img = img
        else:
            col_img = np.hstack((col_img, img))

    begin_index += 5
    end_index += 5
    if np.any(full_img == None):
        full_img = col_img
    else:
        full_img = np.vstack((full_img, col_img))


cv2.imshow("gestures", full_img)
cv2.imwrite('full_img.jpg', full_img)
cv2.waitKey(0)
```



```
akash@akash-HP-Pavilion-Notebook:~/Documents/Capstone -- Sign $ python3 display_
all_gestures.py
```

Fig 28 : Executing File To Display All Gestures

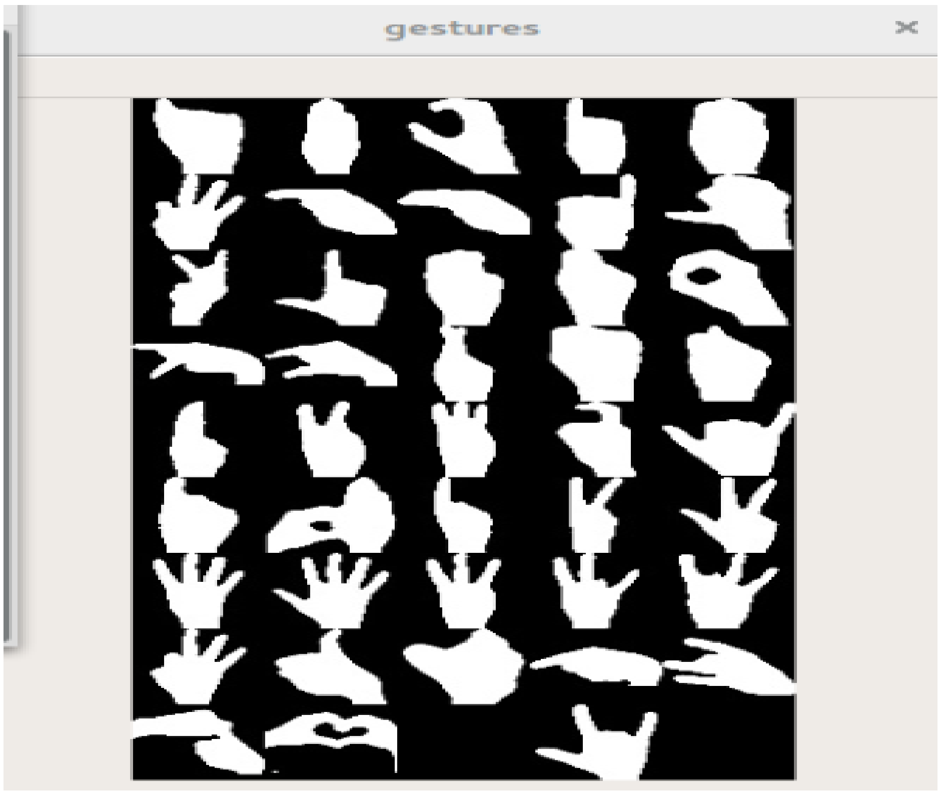
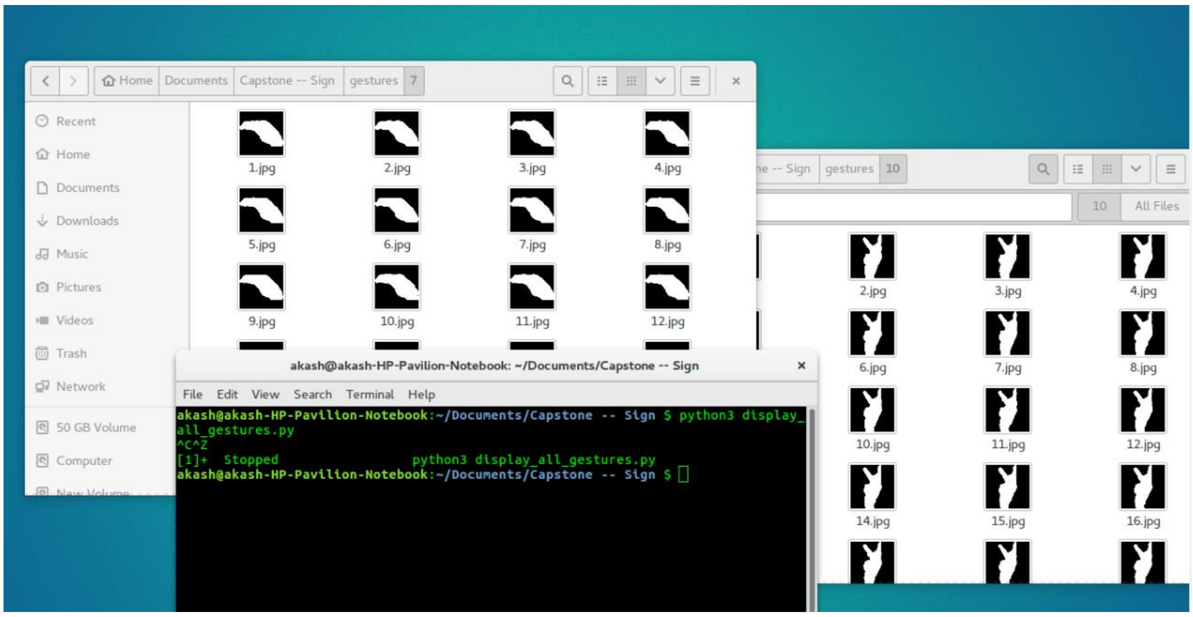


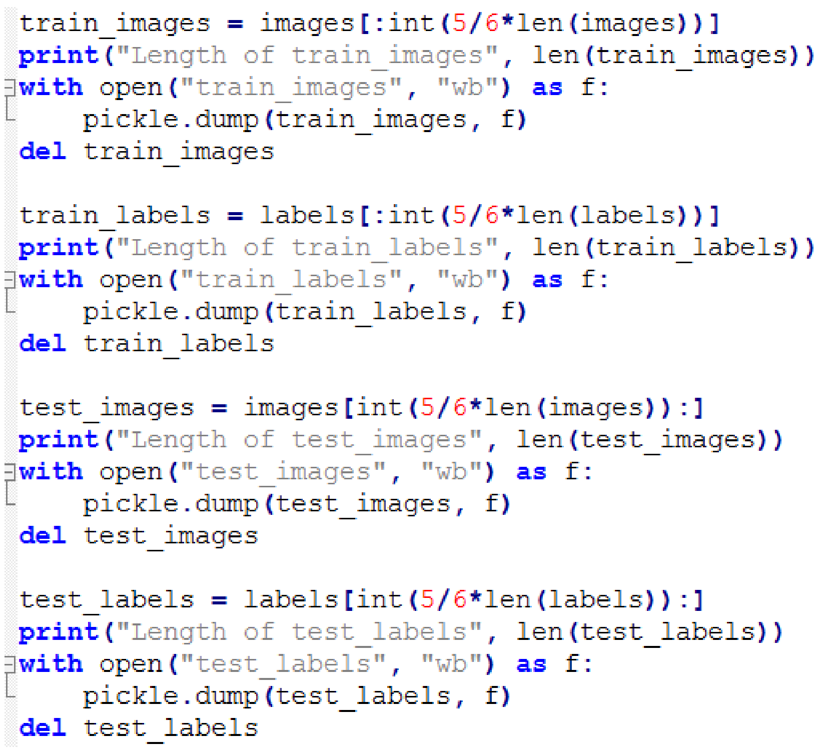
Figure 29 : All Gestures



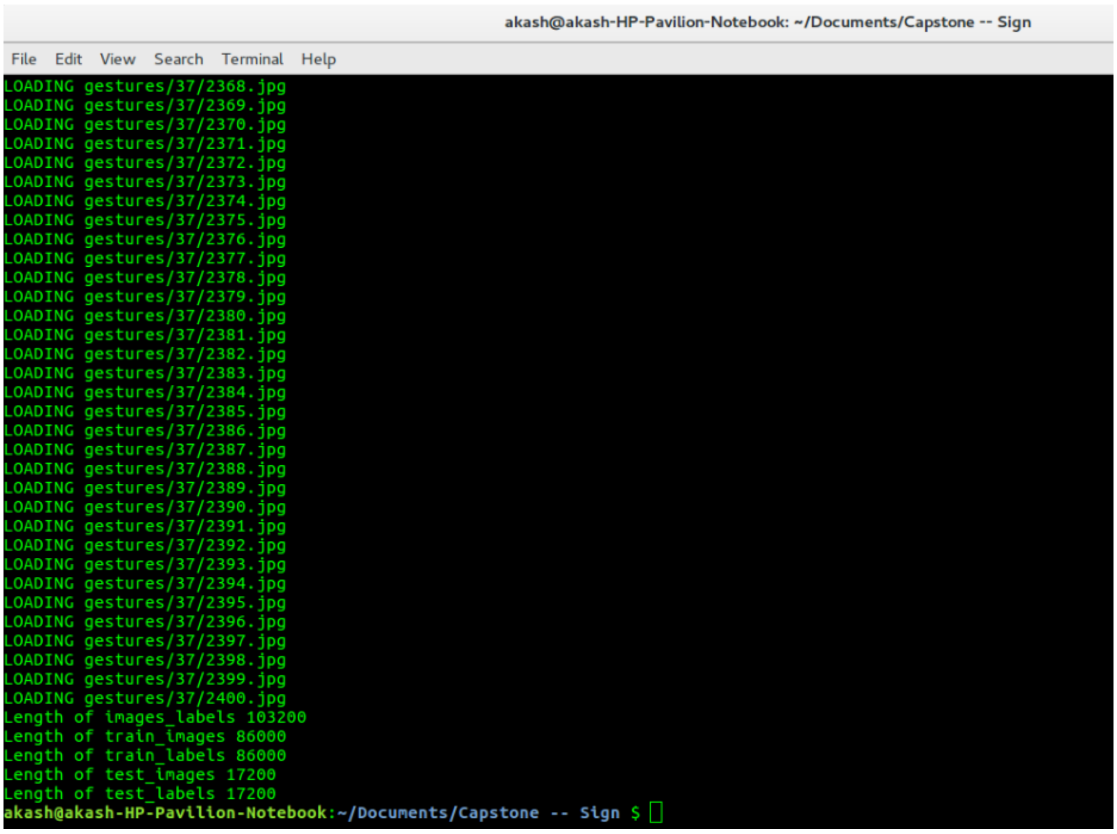
```
akash@akash-HP-Pavilion-Notebook:~/Documents/Capstone -- Sign $ python3 display_
all_gestures.py
^C^Z
[1]+  Stopped                 python3 display_all_gestures.py
akash@akash-HP-Pavilion-Notebook:~/Documents/Capstone -- Sign $ 
```

Figure 30 : Dataset Creation

Load Images



```
train_images = images[:int(5/6*len(images))]
print("Length of train_images", len(train_images))
with open("train_images", "wb") as f:
    pickle.dump(train_images, f)
del train_images

train_labels = labels[:int(5/6*len(labels))]
print("Length of train_labels", len(train_labels))
with open("train_labels", "wb") as f:
    pickle.dump(train_labels, f)
del train_labels

test_images = images[int(5/6*len(images)):]
print("Length of test_images", len(test_images))
with open("test_images", "wb") as f:
    pickle.dump(test_images, f)
del test_images

test_labels = labels[int(5/6*len(labels)):]
print("Length of test_labels", len(test_labels))
with open("test_labels", "wb") as f:
    pickle.dump(test_labels, f)
del test_labels
```

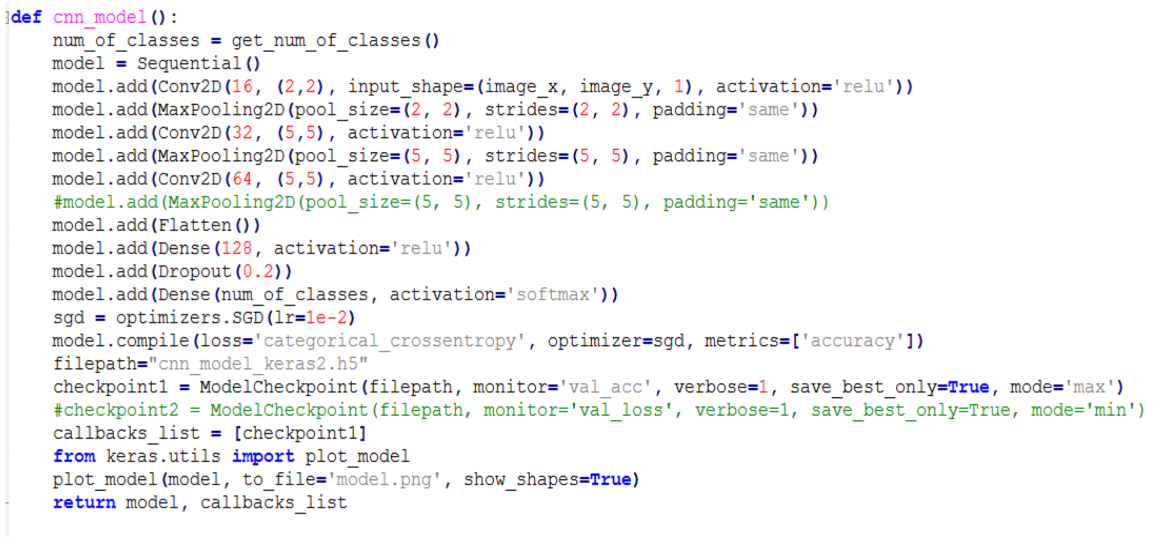


akash@akash-HP-Pavilion-Notebook: ~/Documents/Capstone -- Sign
File Edit View Search Terminal Help

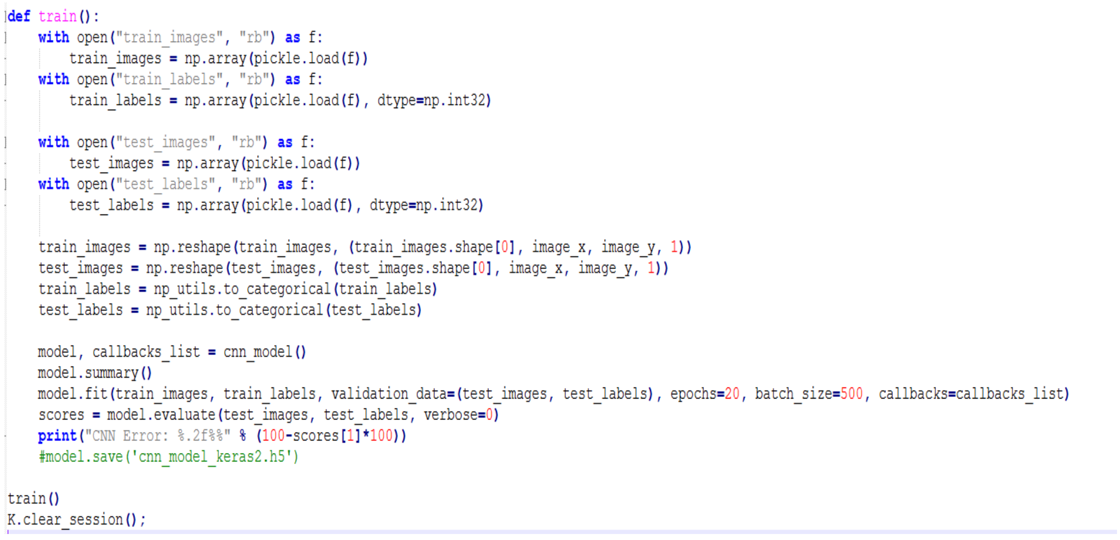
```
LOADING gestures/37/2368.jpg
LOADING gestures/37/2369.jpg
LOADING gestures/37/2370.jpg
LOADING gestures/37/2371.jpg
LOADING gestures/37/2372.jpg
LOADING gestures/37/2373.jpg
LOADING gestures/37/2374.jpg
LOADING gestures/37/2375.jpg
LOADING gestures/37/2376.jpg
LOADING gestures/37/2377.jpg
LOADING gestures/37/2378.jpg
LOADING gestures/37/2379.jpg
LOADING gestures/37/2380.jpg
LOADING gestures/37/2381.jpg
LOADING gestures/37/2382.jpg
LOADING gestures/37/2383.jpg
LOADING gestures/37/2384.jpg
LOADING gestures/37/2385.jpg
LOADING gestures/37/2386.jpg
LOADING gestures/37/2387.jpg
LOADING gestures/37/2388.jpg
LOADING gestures/37/2389.jpg
LOADING gestures/37/2390.jpg
LOADING gestures/37/2391.jpg
LOADING gestures/37/2392.jpg
LOADING gestures/37/2393.jpg
LOADING gestures/37/2394.jpg
LOADING gestures/37/2395.jpg
LOADING gestures/37/2396.jpg
LOADING gestures/37/2397.jpg
LOADING gestures/37/2398.jpg
LOADING gestures/37/2399.jpg
LOADING gestures/37/2400.jpg
Length of images_labels 103200
Length of train_images 86000
Length of train_labels 86000
Length of test_images 17200
Length of test_labels 17200
akash@akash-HP-Pavilion-Notebook:~/Documents/Capstone -- Sign $ 
```

Figure 31 : Loading Images

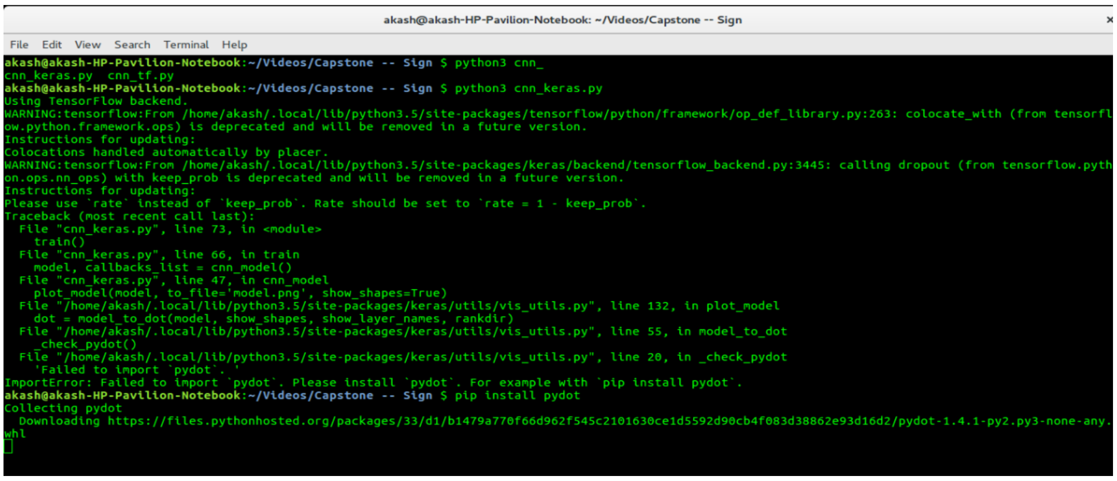
CNN



```python
def cnn_model():
    num_of_classes = get_num_of_classes()
    model = Sequential()
    model.add(Conv2D(16, (2,2), input_shape=(image_x, image_y, 1), activation='relu'))
    model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2), padding='same'))
    model.add(Conv2D(32, (5,5), activation='relu'))
    model.add(MaxPooling2D(pool_size=(5, 5), strides=(5, 5), padding='same'))
    model.add(Conv2D(64, (5,5), activation='relu'))
    #model.add(MaxPooling2D(pool_size=(5, 5), strides=(5, 5), padding='same'))
    model.add(Flatten())
    model.add(Dense(128, activation='relu'))
    model.add(Dropout(0.2))
    model.add(Dense(num_of_classes, activation='softmax'))
    sgd = optimizers.SGD(lr=1e-2)
    model.compile(loss='categorical_crossentropy', optimizer=sgd, metrics=['accuracy'])
    filepath="cnn_model_keras2.h5"
    checkpoint1 = ModelCheckpoint(filepath, monitor='val_acc', verbose=1, save_best_only=True, mode='max')
    #checkpoint2 = ModelCheckpoint(filepath, monitor='val_loss', verbose=1, save_best_only=True, mode='min')
    callbacks_list = [checkpoint1]
    from keras.utils import plot_model
    plot_model(model, to_file='model.png', show_shapes=True)
    return model, callbacks_list
```



```python
def train():
    with open("train_images", "rb") as f:
        train_images = np.array(pickle.load(f))
    with open("train_labels", "rb") as f:
        train_labels = np.array(pickle.load(f), dtype=np.int32)

    with open("test_images", "rb") as f:
        test_images = np.array(pickle.load(f))
    with open("test_labels", "rb") as f:
        test_labels = np.array(pickle.load(f), dtype=np.int32)

    train_images = np.reshape(train_images, (train_images.shape[0], image_x, image_y, 1))
    test_images = np.reshape(test_images, (test_images.shape[0], image_x, image_y, 1))
    train_labels = np_utils.to_categorical(train_labels)
    test_labels = np_utils.to_categorical(test_labels)

    model, callbacks_list = cnn_model()
    model.summary()
    model.fit(train_images, train_labels, validation_data=(test_images, test_labels), epochs=20, batch_size=500, callbacks=callbacks_list)
    scores = model.evaluate(test_images, test_labels, verbose=0)
    print("CNN Error: %.2f%%" % (100-scores[1]*100))
    #model.save('cnn_model_keras2.h5')

train()
K.clear_session();
```

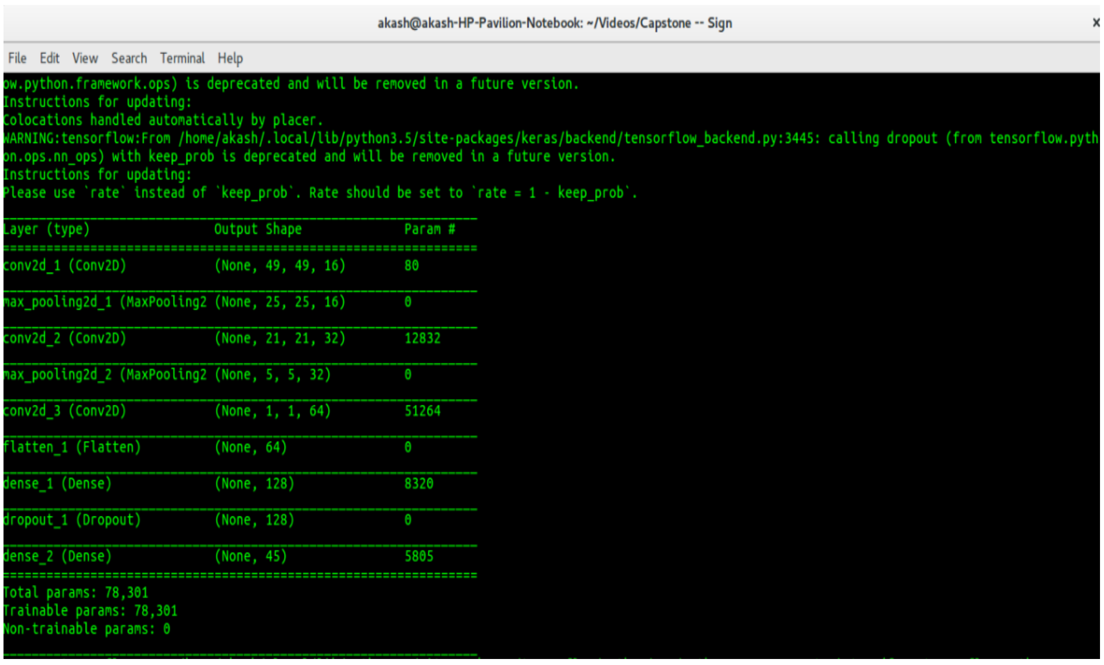


akash@akash-HP-Pavilion-Notebook: ~/Videos/Capstone -- Sign
File Edit View Search Terminal Help

```
akash@akash-HP-Pavilion-Notebook:~/Videos/Capstone -- Sign $ python3 cnn_
cnn_keras.py  cnn_tf.py
akash@akash-HP-Pavilion-Notebook:~/Videos/Capstone -- Sign $ python3 cnn_keras.py
Using TensorFlow backend.
WARNING:tensorflow:From /home/akash/.local/lib/python3.5/site-packages/tensorflow/python/framework/op_def_library.py:263: colocate_with (from tensorfl
ow.python.framework.ops) is deprecated and will be removed in a future version.
Instructions for updating:
Colocations handled automatically by placer.
WARNING:tensorflow:From /home/akash/.local/lib/python3.5/site-packages/keras/backend/tensorflow_backend.py:3445: calling dropout (from tensorflow.pyth
on.ops.nn_ops) with keep_prob is deprecated and will be removed in a future version.
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate = 1 - keep_prob`.
Traceback (most recent call last):
  File "cnn_keras.py", line 73, in <module>
    train()
  File "cnn_keras.py", line 66, in train
    model, callbacks_list = cnn_model()
  File "cnn_keras.py", line 47, in cnn_model
    plot_model(model, to_file='model.png', show_shapes=True)
  File "/home/akash/.local/lib/python3.5/site-packages/keras/utils/vis_utils.py", line 132, in plot_model
    dot = model_to_dot(model, show_shapes, show_layer_names, rankdir)
  File "/home/akash/.local/lib/python3.5/site-packages/keras/utils/vis_utils.py", line 55, in model_to_dot
    _check_pydot()
  File "/home/akash/.local/lib/python3.5/site-packages/keras/utils/vis_utils.py", line 20, in _check_pydot
    'Failed to import `pydot`. '
ImportError: Failed to import `pydot`. Please install `pydot`. For example with `pip install pydot`.
akash@akash-HP-Pavilion-Notebook:~/Videos/Capstone -- Sign $ pip install pydot
Collecting pydot
  Downloading https://files.pythonhosted.org/packages/33/d1/b1479a770f66d962f545c2101630ce1d5592d90cb4f083d38862e93d16d2/pydot-1.4.1-py2.py3-none-any.
whl
```

Figure 32 : Executing File For CNN Model

Model Summary



akash@akash-HP-Pavilion-Notebook: ~/Videos/Capstone -- Sign

File Edit View Search Terminal Help

```
ow.python.framework.ops) is deprecated and will be removed in a future version.
Instructions for updating:
Colocations handled automatically by placer.
WARNING:tensorflow:From /home/akash/.local/lib/python3.5/site-packages/keras/backend/tensorflow_backend.py:3445: calling dropout (from tensorflow.pyth
on.ops.nn_ops) with keep_prob is deprecated and will be removed in a future version.
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate = 1 - keep_prob`.
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 49, 49, 16)        80
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 25, 25, 16)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 21, 21, 32)        12832
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 5, 5, 32)          0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 1, 1, 64)          51264
_________________________________________________________________
flatten_1 (Flatten)          (None, 64)                0
_________________________________________________________________
dense_1 (Dense)              (None, 128)               8320
_________________________________________________________________
dropout_1 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_2 (Dense)              (None, 45)                5805
=================================================================
Total params: 78,301
Trainable params: 78,301
Non-trainable params: 0
_________________________________________________________________
```

Figure 33 : CNN Model Gist

Model Reports

		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

Precision = True Positive / True Positive + False Positive

Recall = True Positive / True Positive + False Negative

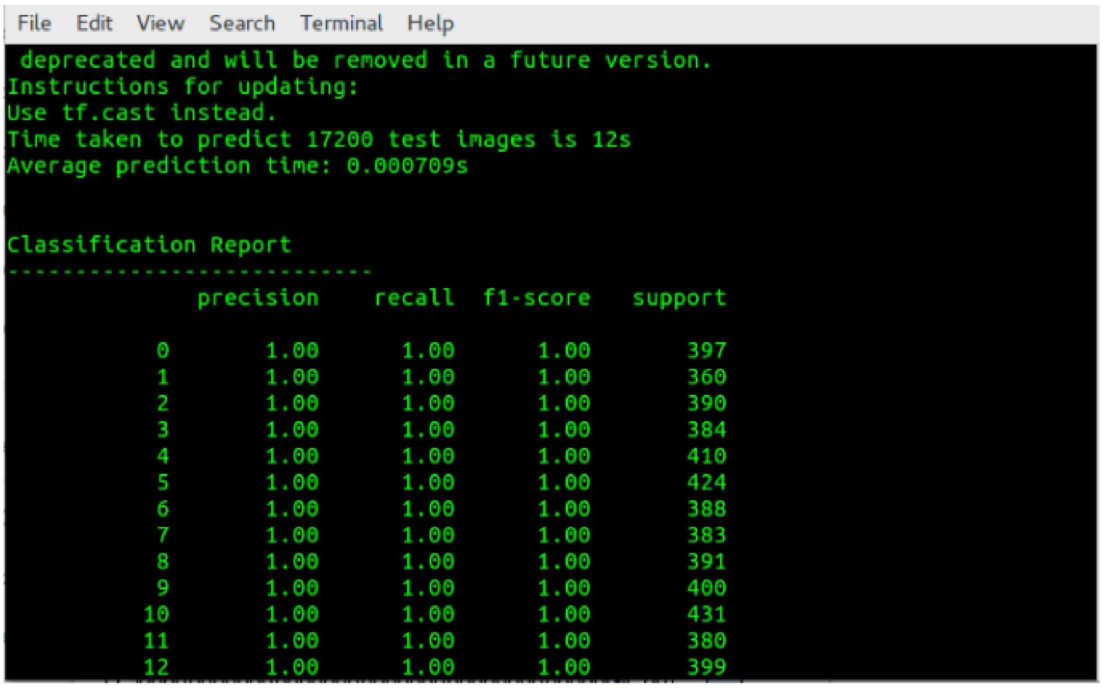
F1 Score = 2*(Recall * Precision) / (Recall + Precision)

Accuracy = True Positive + True Negative / True Positive + True Negative + False Positive + False Negative

precision/recall/f1-score

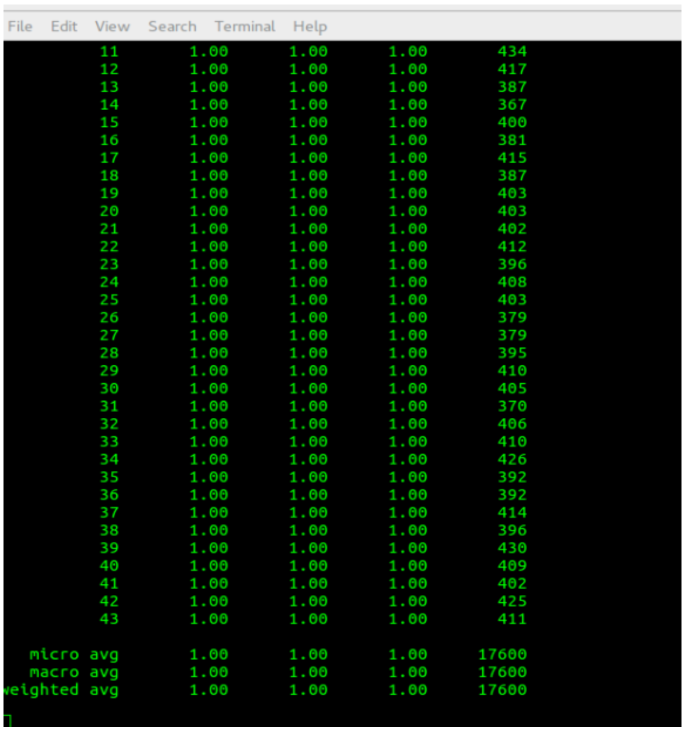
```
import itertools

accuracy = np.trace(cm) / float(np.sum(cm))
misclass = 1 - accuracy

if cmap is None:
    cmap = plt.get_cmap('Blues')

plt.figure(figsize=(5, 5))
plt.imshow(cm, interpolation='nearest', cmap=cmap)
plt.title(title)
plt.colorbar()

if target_names is not None:
    tick_marks = np.arange(len(target_names))
    plt.xticks(tick_marks, target_names, rotation=45)
    plt.yticks(tick_marks, target_names)

if normalize:
    cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]


thresh = cm.max() / 1.5 if normalize else cm.max() / 2
for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
    if normalize:
        plt.text(j, i, "{:0.4f}".format(cm[i, j]),
                 horizontalalignment="center",
                 color="white" if cm[i, j] > thresh else "black")
    else:
        plt.text(j, i, "{:,}".format(cm[i, j]),
                 horizontalalignment="center",
                 color="white" if cm[i, j] > thresh else "black")


plt.tight_layout()
plt.ylabel('True label')
plt.xlabel('Predicted label\naccuracy={:0.4f}; misclass={:0.4f}'.format(accuracy, misclass))
plt.show()
```

```python
start_time = time.time()
pred_probabs = model.predict(test_images)
end_time = time.time()
pred_time = end_time-start_time
avg_pred_time = pred_time/test_images.shape[0]
print("Time taken to predict %d test images is %ds" %(test_images.shape[0], pred_time))
print('Average prediction time: %fs' % (avg_pred_time))

for pred_probab in pred_probabs:
    pred_labels.append(list(pred_probab).index(max(pred_probab)))

cm = confusion_matrix(test_labels, np.array(pred_labels))
classification_report = classification_report(test_labels, np.array(pred_labels))
print('\n\nClassification Report')
print('---------------------------')
print(classification_report)
plot_confusion_matrix(cm, range(44), normalize=False)
```



```
 deprecated and will be removed in a future version.
Instructions for updating:
Use tf.cast instead.
Time taken to predict 17200 test images is 12s
Average prediction time: 0.000709s


Classification Report
---------------------------
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       397
           1       1.00      1.00      1.00       360
           2       1.00      1.00      1.00       390
           3       1.00      1.00      1.00       384
           4       1.00      1.00      1.00       410
           5       1.00      1.00      1.00       424
           6       1.00      1.00      1.00       388
           7       1.00      1.00      1.00       383
           8       1.00      1.00      1.00       391
           9       1.00      1.00      1.00       400
          10       1.00      1.00      1.00       431
          11       1.00      1.00      1.00       380
          12       1.00      1.00      1.00       399
```

Figure 34 : Model Report (Precision,Recall,F1 Score,Support)



```
          11       1.00      1.00      1.00       434
          12       1.00      1.00      1.00       417
          13       1.00      1.00      1.00       387
          14       1.00      1.00      1.00       367
          15       1.00      1.00      1.00       400
          16       1.00      1.00      1.00       381
          17       1.00      1.00      1.00       415
          18       1.00      1.00      1.00       387
          19       1.00      1.00      1.00       403
          20       1.00      1.00      1.00       403
          21       1.00      1.00      1.00       402
          22       1.00      1.00      1.00       412
          23       1.00      1.00      1.00       396
          24       1.00      1.00      1.00       408
          25       1.00      1.00      1.00       403
          26       1.00      1.00      1.00       379
          27       1.00      1.00      1.00       379
          28       1.00      1.00      1.00       395
          29       1.00      1.00      1.00       410
          30       1.00      1.00      1.00       405
          31       1.00      1.00      1.00       370
          32       1.00      1.00      1.00       406
          33       1.00      1.00      1.00       410
          34       1.00      1.00      1.00       426
          35       1.00      1.00      1.00       392
          36       1.00      1.00      1.00       392
          37       1.00      1.00      1.00       414
          38       1.00      1.00      1.00       396
          39       1.00      1.00      1.00       430
          40       1.00      1.00      1.00       409
          41       1.00      1.00      1.00       402
          42       1.00      1.00      1.00       425
          43       1.00      1.00      1.00       411

   micro avg       1.00      1.00      1.00     17600
   macro avg       1.00      1.00      1.00     17600
eighted avg       1.00      1.00      1.00     17600
```

Figure 35 : Model Report 2

Confusion Matrix

Confusion matrix is also termed as an error matrix which is a particular table layout that creates visualization of the performance of an algorithm

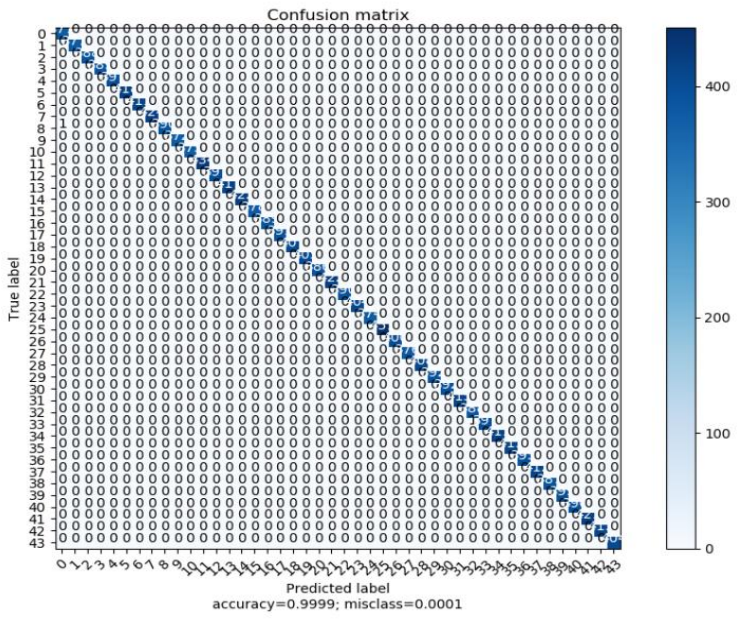


Figure 36 : Confusion Matrix

Validation and Loss Visualization

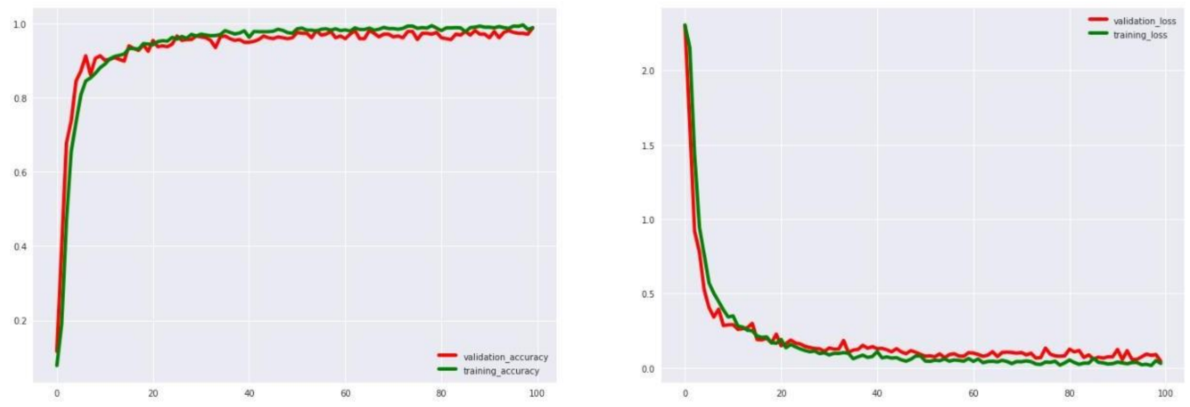
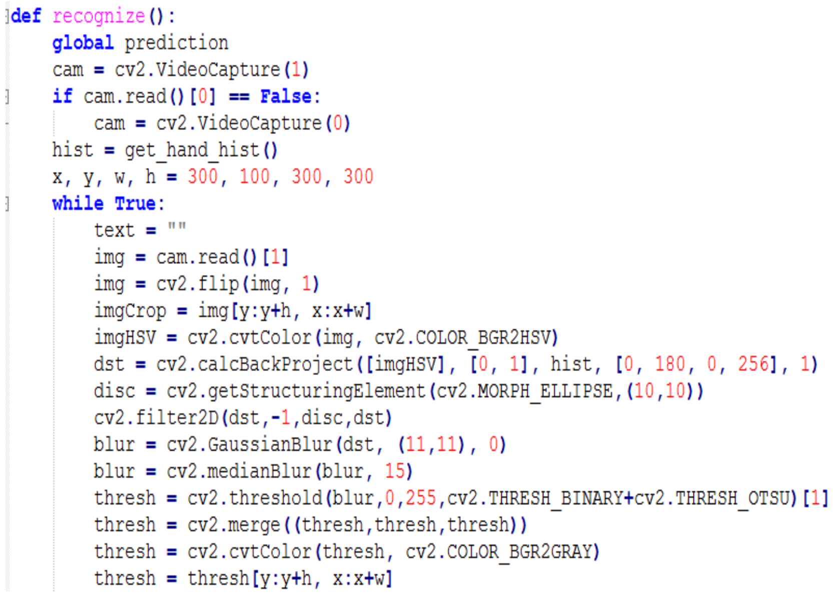
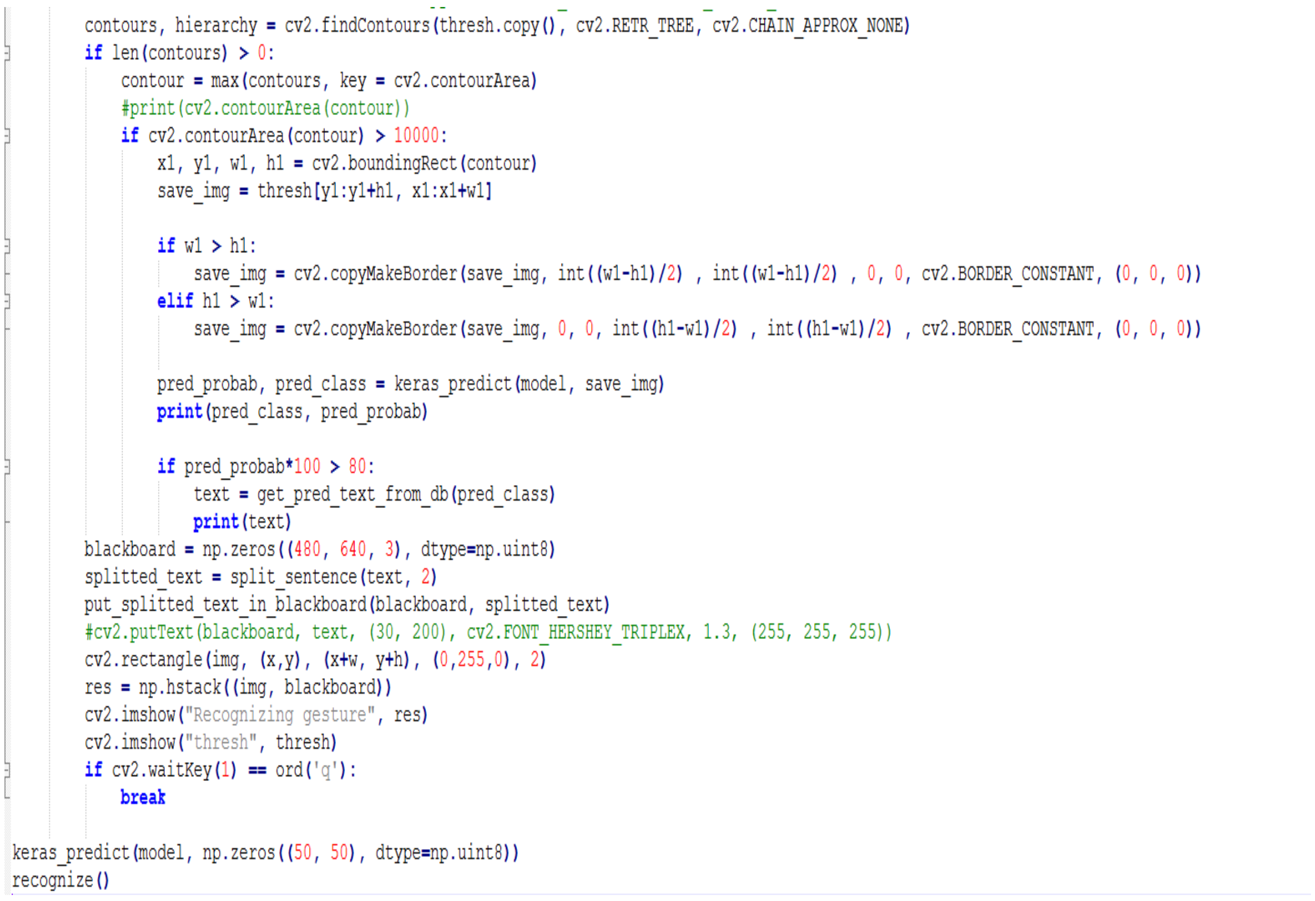


Figure 37 : Accuracy And Loss Graph

Recognize Gesture



```
def recognize():
    global prediction
    cam = cv2.VideoCapture(1)
    if cam.read()[0] == False:
        cam = cv2.VideoCapture(0)
    hist = get_hand_hist()
    x, y, w, h = 300, 100, 300, 300
    while True:
        text = ""
        img = cam.read()[1]
        img = cv2.flip(img, 1)
        imgCrop = img[y:y+h, x:x+w]
        imgHSV = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
        dst = cv2.calcBackProject([imgHSV], [0, 1], hist, [0, 180, 0, 256], 1)
        disc = cv2.getStructuringElement(cv2.MORPH_ELLIPSE,(10,10))
        cv2.filter2D(dst,-1,disc,dst)
        blur = cv2.GaussianBlur(dst, (11,11), 0)
        blur = cv2.medianBlur(blur, 15)
        thresh = cv2.threshold(blur,0,255,cv2.THRESH_BINARY+cv2.THRESH_OTSU)[1]
        thresh = cv2.merge((thresh,thresh,thresh))
        thresh = cv2.cvtColor(thresh, cv2.COLOR_BGR2GRAY)
        thresh = thresh[y:y+h, x:x+w]
```


```
        contours, hierarchy = cv2.findContours(thresh.copy(), cv2.RETR_TREE, cv2.CHAIN_APPROX_NONE)
        if len(contours) > 0:
            contour = max(contours, key = cv2.contourArea)
            #print(cv2.contourArea(contour))
            if cv2.contourArea(contour) > 10000:
                x1, y1, w1, h1 = cv2.boundingRect(contour)
                save_img = thresh[y1:y1+h1, x1:x1+w1]

                if w1 > h1:
                    save_img = cv2.copyMakeBorder(save_img, int((w1-h1)/2) , int((w1-h1)/2) , 0, 0, cv2.BORDER_CONSTANT, (0, 0, 0))
                elif h1 > w1:
                    save_img = cv2.copyMakeBorder(save_img, 0, 0, int((h1-w1)/2) , int((h1-w1)/2) , cv2.BORDER_CONSTANT, (0, 0, 0))

                pred_probab, pred_class = keras_predict(model, save_img)
                print(pred_class, pred_probab)

                if pred_probab*100 > 80:
                    text = get_pred_text_from_db(pred_class)
                    print(text)
        blackboard = np.zeros((480, 640, 3), dtype=np.uint8)
        splitted_text = split_sentence(text, 2)
        put_splitted_text_in_blackboard(blackboard, splitted_text)
        #cv2.putText(blackboard, text, (30, 200), cv2.FONT_HERSHEY_TRIPLEX, 1.3, (255, 255, 255))
        cv2.rectangle(img, (x,y), (x+w, y+h), (0,255,0), 2)
        res = np.hstack((img, blackboard))
        cv2.imshow("Recognizing gesture", res)
        cv2.imshow("thresh", thresh)
        if cv2.waitKey(1) == ord('q'):
            break

keras_predict(model, np.zeros((50, 50), dtype=np.uint8))
recognize()
```

Alphabet

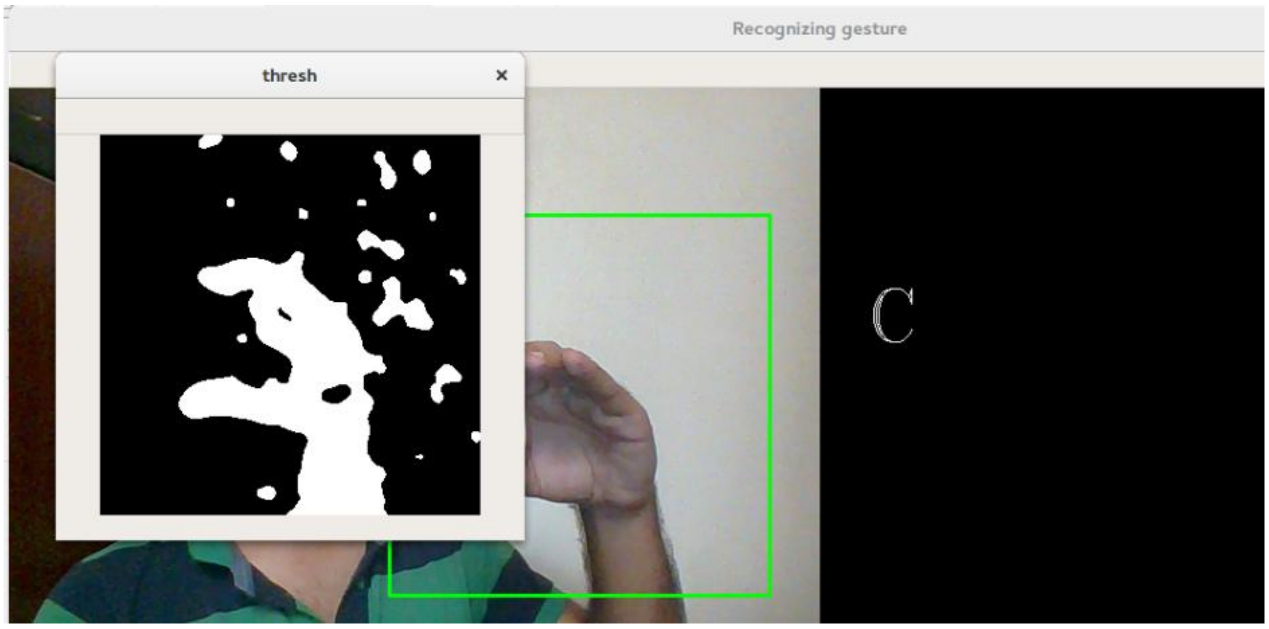


Figure 38 : Recognizing Gesture I

Numerical

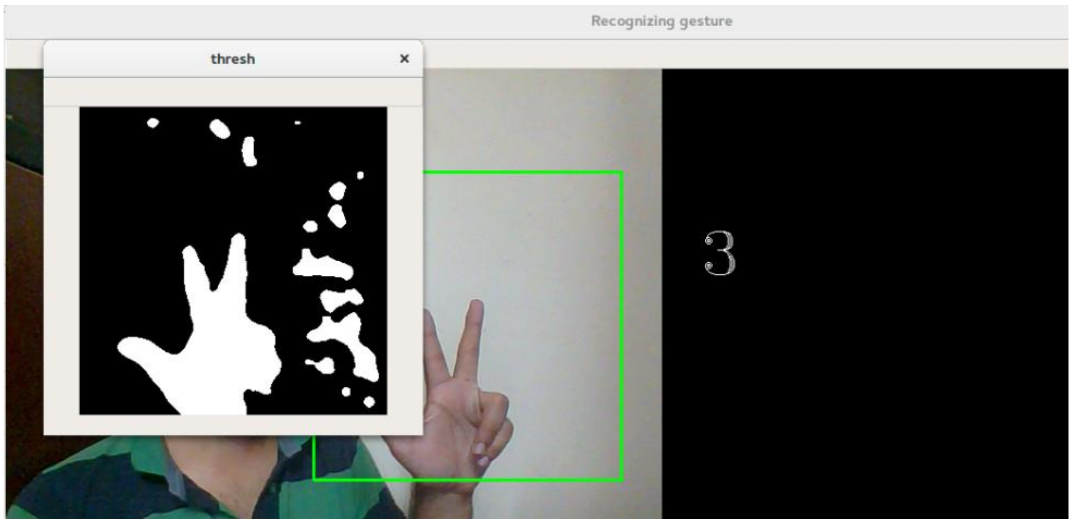


Figure 39 : Recognizing Gesture II

All the Best

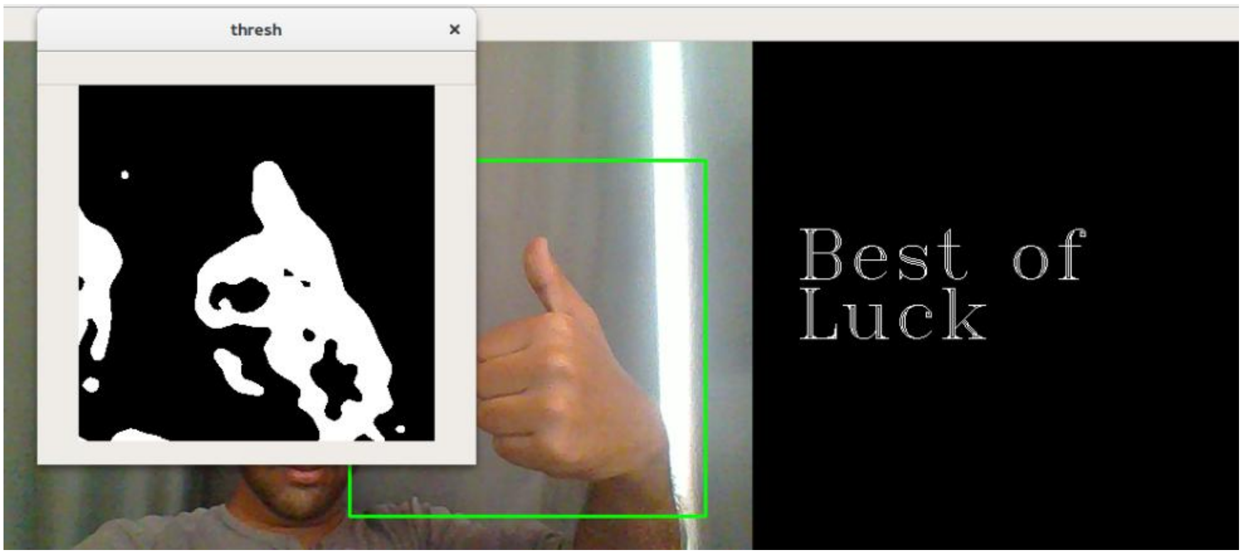
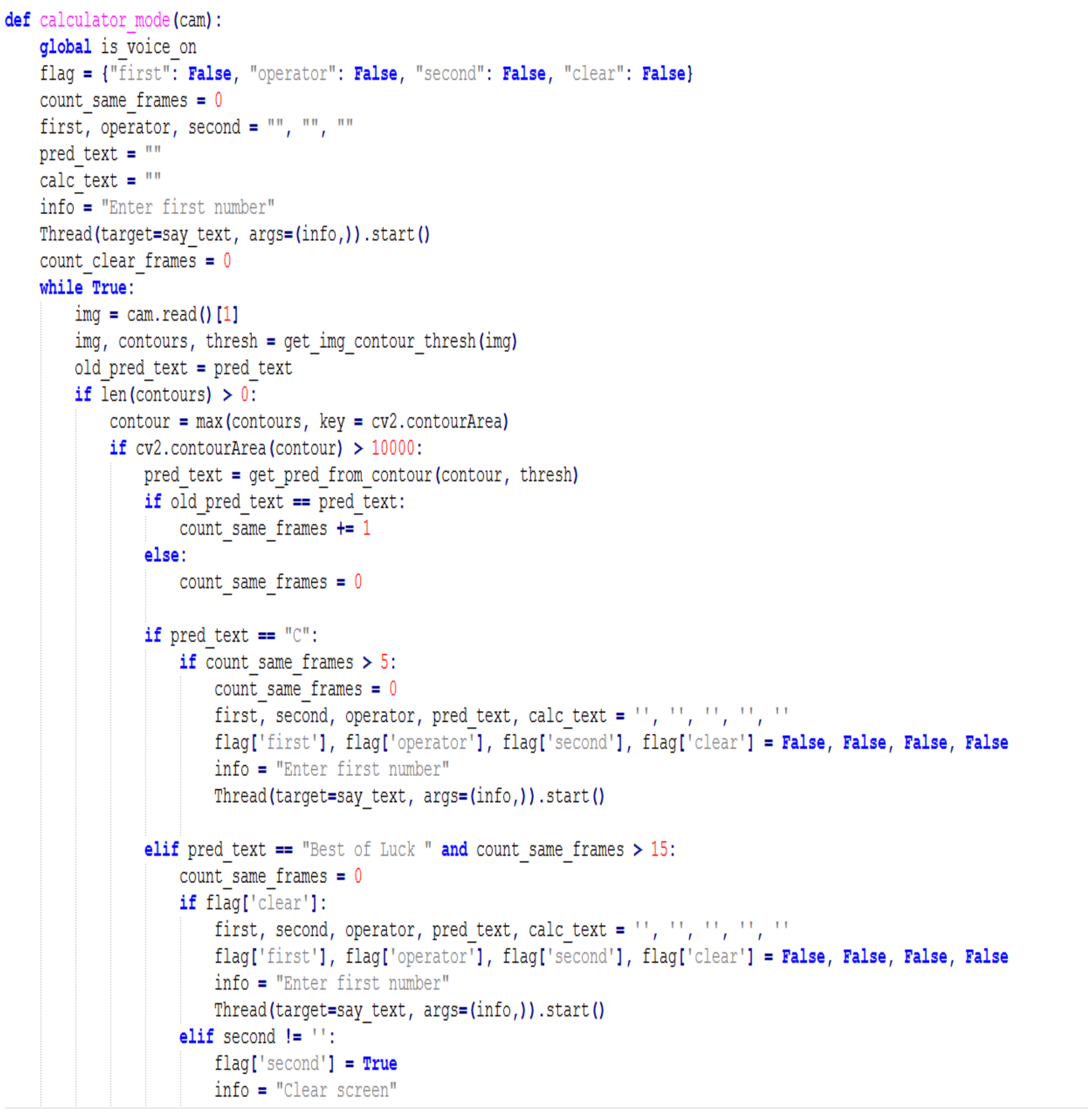


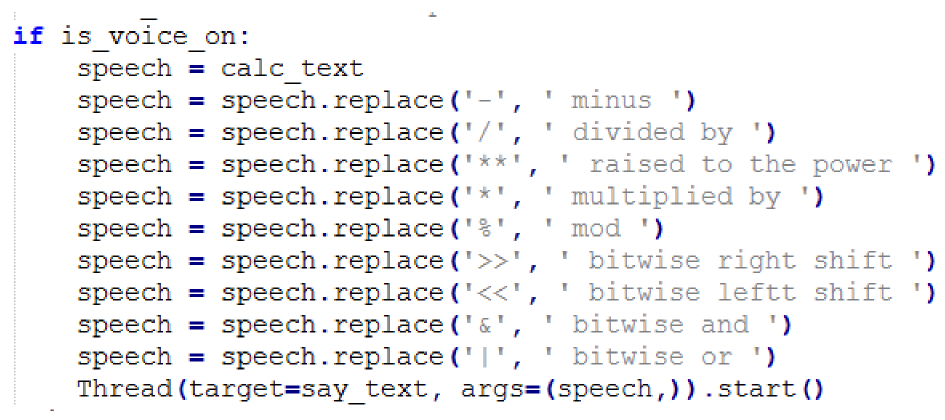
Figure 40 : Recognizing Gesture III

Calculator Using Digit Inputs

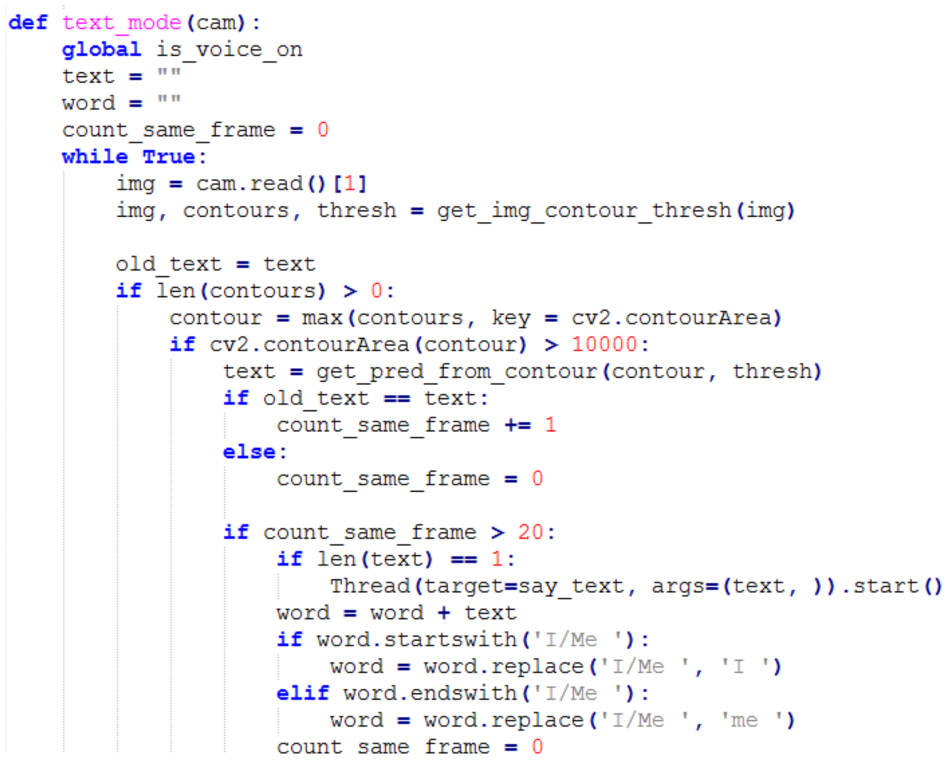
Input - Gestures as Digits



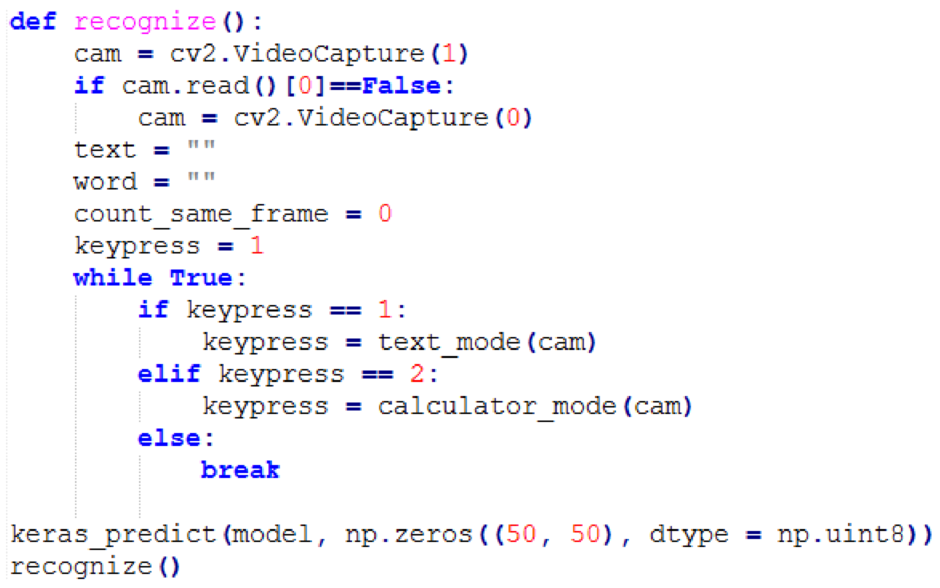
```python
def calculator_mode(cam):
    global is_voice_on
    flag = {"first": False, "operator": False, "second": False, "clear": False}
    count_same_frames = 0
    first, operator, second = "", "", ""
    pred_text = ""
    calc_text = ""
    info = "Enter first number"
    Thread(target=say_text, args=(info,)).start()
    count_clear_frames = 0
    while True:
        img = cam.read()[1]
        img, contours, thresh = get_img_contour_thresh(img)
        old_pred_text = pred_text
        if len(contours) > 0:
            contour = max(contours, key = cv2.contourArea)
            if cv2.contourArea(contour) > 10000:
                pred_text = get_pred_from_contour(contour, thresh)
                if old_pred_text == pred_text:
                    count_same_frames += 1
                else:
                    count_same_frames = 0

                if pred_text == "C":
                    if count_same_frames > 5:
                        count_same_frames = 0
                        first, second, operator, pred_text, calc_text = '', '', '', '', ''
                        flag['first'], flag['operator'], flag['second'], flag['clear'] = False, False, False, False
                        info = "Enter first number"
                        Thread(target=say_text, args=(info,)).start()

                elif pred_text == "Best of Luck " and count_same_frames > 15:
                    count_same_frames = 0
                    if flag['clear']:
                        first, second, operator, pred_text, calc_text = '', '', '', '', ''
                        flag['first'], flag['operator'], flag['second'], flag['clear'] = False, False, False, False
                        info = "Enter first number"
                        Thread(target=say_text, args=(info,)).start()
                    elif second != '':
                        flag['second'] = True
                        info = "Clear screen"
```



```
if is_voice_on:
    speech = calc_text
    speech = speech.replace('-', ' minus ')
    speech = speech.replace('/', ' divided by ')
    speech = speech.replace('**', ' raised to the power ')
    speech = speech.replace('*', ' multiplied by ')
    speech = speech.replace('%', ' mod ')
    speech = speech.replace('>>', ' bitwise right shift ')
    speech = speech.replace('<<', ' bitwise lefft shift ')
    speech = speech.replace('&', ' bitwise and ')
    speech = speech.replace('|', ' bitwise or ')
    Thread(target=say_text, args=(speech,)).start()
```



```
def text_mode(cam):
    global is_voice_on
    text = ""
    word = ""
    count_same_frame = 0
    while True:
        img = cam.read()[1]
        img, contours, thresh = get_img_contour_thresh(img)

        old_text = text
        if len(contours) > 0:
            contour = max(contours, key = cv2.contourArea)
            if cv2.contourArea(contour) > 10000:
                text = get_pred_from_contour(contour, thresh)
                if old_text == text:
                    count_same_frame += 1
                else:
                    count_same_frame = 0

                if count_same_frame > 20:
                    if len(text) == 1:
                        Thread(target=say_text, args=(text, )).start()
                    word = word + text
                    if word.startswith('I/Me '):
                        word = word.replace('I/Me ', 'I ')
                    elif word.endswith('I/Me '):
                        word = word.replace('I/Me ', 'me ')
                    count_same_frame = 0
```



```
def recognize():
    cam = cv2.VideoCapture(1)
    if cam.read()[0]==False:
        cam = cv2.VideoCapture(0)
    text = ""
    word = ""
    count_same_frame = 0
    keypress = 1
    while True:
        if keypress == 1:
            keypress = text_mode(cam)
        elif keypress == 2:
            keypress = calculator_mode(cam)
        else:
            break

keras_predict(model, np.zeros((50, 50), dtype = np.uint8))
recognize()
```

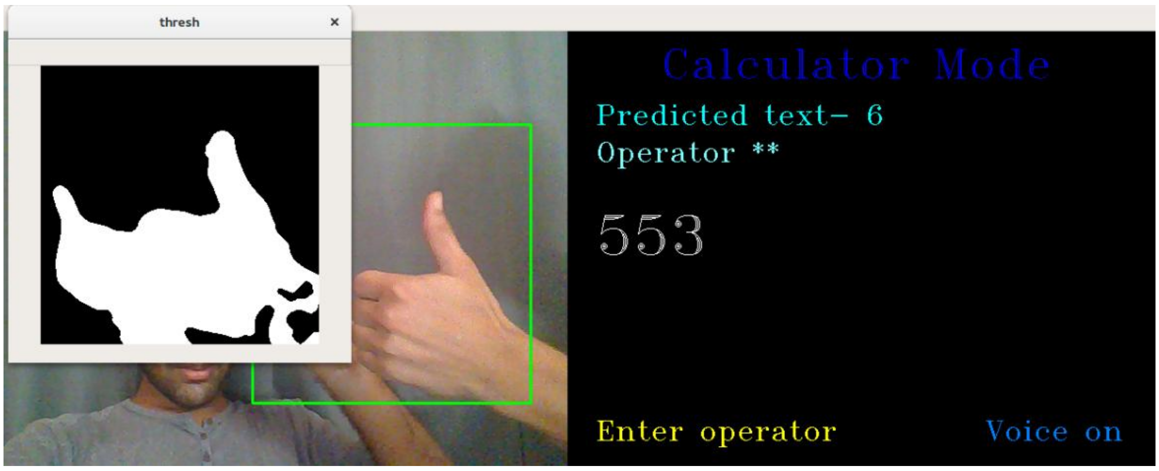


Figure 41 : Taking Input For Calculator

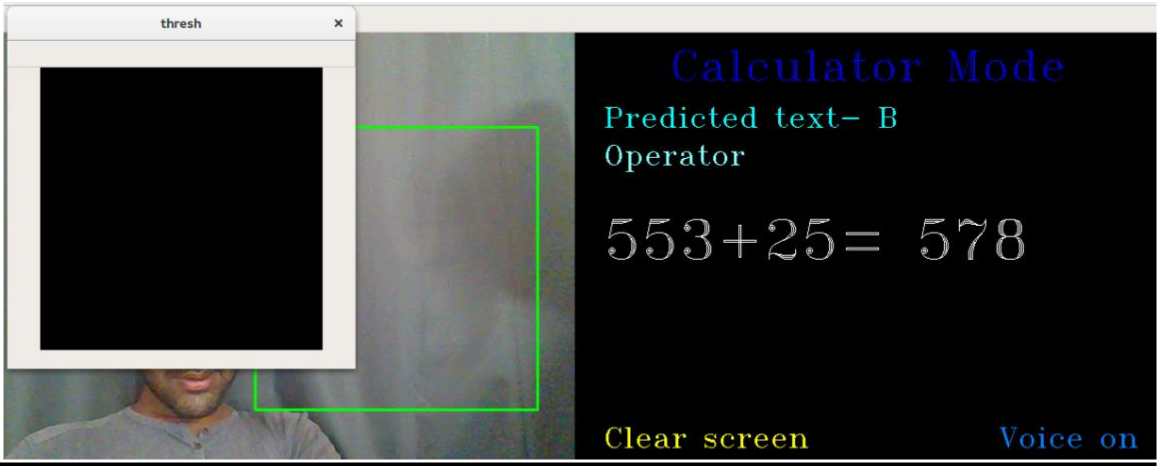


Figure 42 : Performing Mathematical Operations

7. Constraints and Alternatives

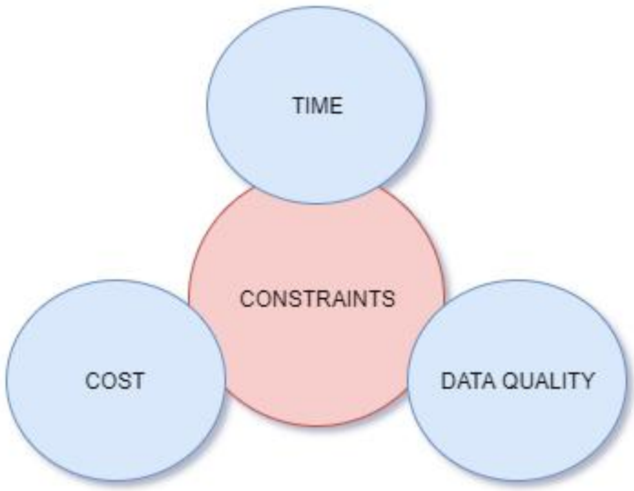


Figure 43 : Major Constraints

7.1 Constraints

The major constraints of our project are as follows:

1. Time :

The time taken for training the data is more in the local machine, so this makes us work on the cloud as it is deployed easily.

2. Cost :

We cannot use sensors as it would be costly, so we end up using CNN model which is cost effective.

3. Data Quality :

Inconsistent duplicate corrupt data needs to be handled which should be done carefully.

Online dataset would be much easier to take in use but it won't be reliable enough as our custom input or our own images dataset.

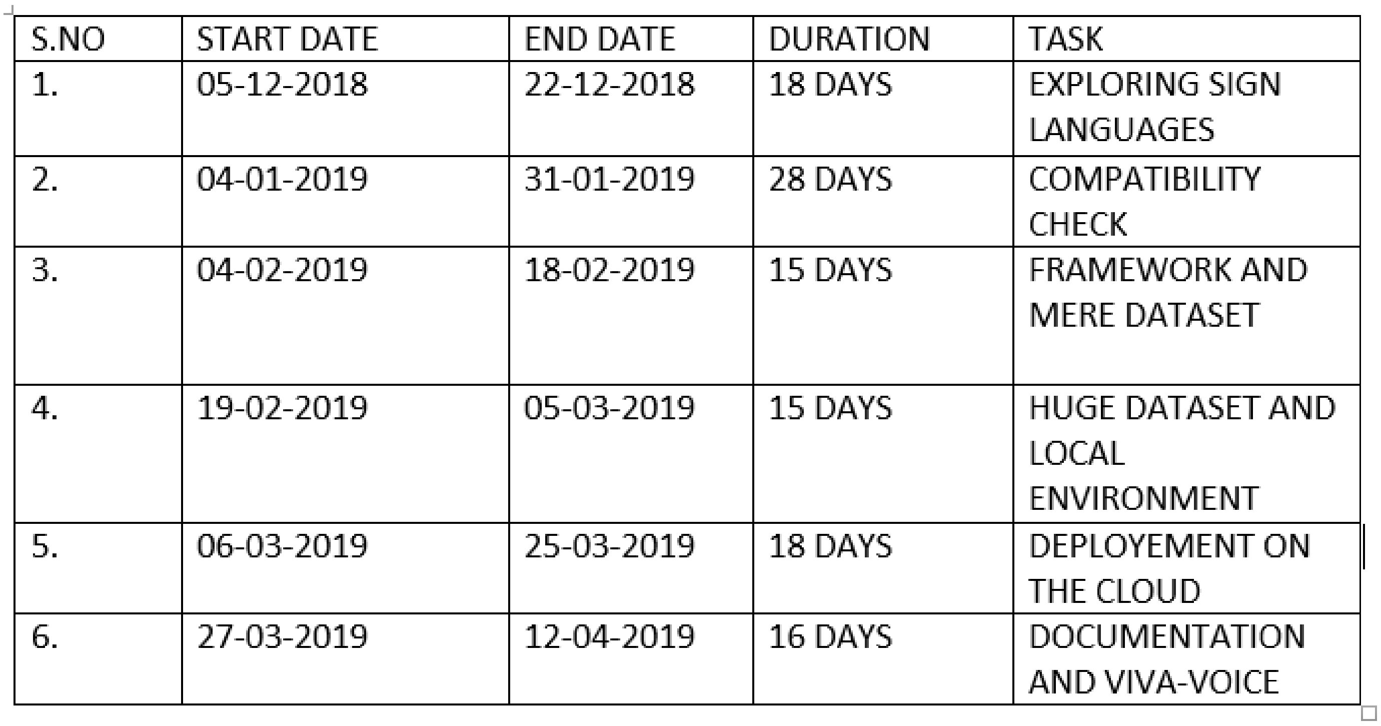
7.2 Alternatives

1. As we had the constraints of time being required more for the local machine so the alternative to that would be cloud environment.

2. Instead of sensors, our framework in which CNN model works and this can be used as an alternative.

3.Since built in images are not reliable so we can use our own image dataset and custom input as an alternative.

8. Schedule, Tasks and Milestones



S.NO	START DATE	END DATE	DURATION	TASK
1.	05-12-2018	22-12-2018	18 DAYS	EXPLORING SIGN LANGUAGES
2.	04-01-2019	31-01-2019	28 DAYS	COMPATIBILITY CHECK
3.	04-02-2019	18-02-2019	15 DAYS	FRAMEWORK AND MERE DATASET
4.	19-02-2019	05-03-2019	15 DAYS	HUGE DATASET AND LOCAL ENVIRONMENT
5.	06-03-2019	25-03-2019	18 DAYS	DEPLOYEMENT ON THE CLOUD
6.	27-03-2019	12-04-2019	16 DAYS	DOCUMENTATION AND VIVA-VOICE

Figure 44 : Break Down Analysis OF Work

Module 1

Exploring Sign Languages , technologies which can be used , and the efficient algorithms. We concluded with ASL.

Module 2

Checking the compatibility of the framework which stands more efficient among MATLAB and Python .

Module 3

Set up the framework and playing with small datasets of digits.

Module 4

Handling huge dataset such as that of alphabets and checking it in our local machines.

Module 5

Deploying it our cloud as it is more feasible, less time constraints and is more reliable.

Module 6

Compiling the entire work , its documentation with several supporting documents .

9. Market Analysis

Our project mainly focus on deaf and dumb peoples problem .Normally deaf and dumb people communicate among themselves and with common people using sign language or other expensive gadgets . They need to carry these gadgets everywhere and not all of them can afford these gadgets. So they mostly use sign language for communication. However our framework is cheap and can be accessed by anyone, anywhere easily. The number of deaf-mutes in the world are roughly calculated to be from 700,000 to 900,000" and of these 63 percent, are said to be born deaf, the others losing their hearing by different accidents. From our framework their lives can be enhanced.

10. Summary

Have you ever wondered how life would be without interacting with the ones who are nearby or let me frame it the other way what if you are unable to speak and hear, what if you can’t express your views, or what if you can’t share your ideas. So this is what is experienced by the deaf and dumb. The solution to this is sign language. Sign Language is the only way in which the deaf and dumb people can communicate to their different communities as well as with the common people the vice versa is also true. The only way common people can communicate to the deaf and dumb is sign language, which makes a successful communication or in other words effective communication. Now on analysing the current scenario we as group members have decided to work on the project which would be quite useful for the society.

So, our project is basically sign language recognition which would be quite beneficial for the deaf and dumb as well as common people. So, the methodology adopted in our project is to develop a framework that would carry on the entire process. The model used is CNN(Convolution Neural Networks).So initially we made the dataset by capturing images which comprises of alphabets and digits .The alphabets would be used for the formation of text and the digits would be used for the mathematical operations. So technically the first step is capturing custom images and matching it with the dataset, if the images matches we end up getting a letter or digit accordingly. Each letter is captured one by one thus forming texts and then voice accordingly. For the digit we can do the mathematical operations like addition, subtraction, multiplication and division as well thus bringing calculator into the picture. So this model basically helps us to get an image of better quality which does not hinders the further process thus giving better results.

From the entire framework we can conclude that the above framework is quite reliable, and the model gives the best results with accuracy 99.99% which is quite reliable and is cost effective as well. So we can use this model for the better livelihood of deaf and dumb people who can reciprocate as well to the common people.

11. References

[1] Zhengzhe Liu, Fuyang Huang, Gladys Wai Lan Tang, Felix Yim Binh Sze, Jing Qin, Xiaogang Wang, and Qiang Xu. “Real-time sign language-recognition with guided deep convolutional neural networks”. 2016 Symposium on Spatial User Interaction, pages-187–187. ACM, 2016.

[2] G. Anantha Rao, Pvv Kishore, D. Anil Kumar, and Ascs Sastry. “Neural network classifier for a continuous sign language recognition systemwith selfie video”.

[3] G. Ananth Rao and Pvv Kishore. “Selfie video based continuous indian sign language recognition system”. Ain Shams Engineering Journal, 2017.

[4] Chai, X., Li, G., Lin, Y., Xu, Z., Tang, Y., Chen, X., Zhou, M.: Sign Language Recognition and Translation with Kinect (2013), http://vipl.ict.ac.cn/sites/default/files/papers/files/2013 FG xjchai Sign Language Recognition and Translation with Kinect.pdf.

[5] Cooper, H., Ong, E.J., Pugeault, N., Bowden, R.: Sign language recognition using sub-units. The Journal of Machine Learning Research 13(1), 2205–2231 (2012).

[6] Escalera, S., Bar, X., Gonzlez, J., Bautista, M.A., Madadi, M., Reyes, M., Ponce, V., Escalante, H.J., Shotton, J., Guyon, I.: Chalearn looking at people challenge 2014: Dataset and results. In: ECCV workshop (2014)

[7] Nandy, Anup, Jay Shankar Prasad, Soumik Mondal, Pavan Chakraborty, and Gora Chand Nandi. "Recognition of isolated indian sign language gesture in real time." Information Processing and Management (2010): 102107.

[8] Bengio, Yoshua, Patrice Simard, and Paolo Frasconi. "Learning longterm dependencies with gradient descent is difficult " .

[9] Juha K., Panu K., Jani M., Sanna K., Giuseppe S., Luca J. and Sergio D. M. Accelerometer-based gesture control for a design environment,
Springer, Finland, 2005.

[10] Goodfellow, I.J., Bulatov, Y., Ibarz, J., Arnoud, S., Shet, V.: Multi-digit number recognition from street view imagery using deep convolutional neural networks. arXiv preprint arXiv:1312.6082 (2013).

[11] Goodfellow, I.J., Warde-Farley, D., Lamblin, P., Dumoulin, V., Mirza, M., Pascanu, R., Bergstra, J., Bastien, F., Bengio, Y.: Pylearn2: a machine learning research library. arXiv preprint arXiv:1308.4214 (2013), http://arxiv.org/abs/1308.4214

[12] Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580 (2012) .

[13] Ronchetti, Franco, Facundo Quiroga, César Armando Estrebou, and Laura Cristina Lanzarini. "Handshape recognition for argentinian sign language using probsom." Journal of Computer Science & Technology 16 (2016).

[14] Singha, Joyeeta, and Karen Das. "Automatic Indian Sign Language Recognition for Continuous Video Sequence." ADBU Journal of
Engineering Technology 2, no. 1 (2015).

[15] Tripathi, Kumud, and Neha Baranwal GC Nandi. "Continuous Indian Sign Language Gesture Recognition and Sentence
Formation." Procedia Computer Science 54 (2015): 523531.

[16] Jarrett, K., Kavukcuoglu, K.: What is the best multi-stage architecture for object recognition? Computer Vision, 2009 IEEE 12th International Conference on pp. 2146–2153 (2009), http://ieeexplore.ieee.org/xpls/abs all.jsp?arnumber=5459469

[17] Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Largescale video classification with convolutional neural networks. In: CVPR (2014)

[18] Krizhevsky, A., Sutskever, I., Hinton, G.: Imagenet classification with deep convolutional neural networks. Advances in Neural Information pp. 1–9 (2012), http://books.nips.cc/papers/files/nips25/NIPS2012 0534.pdf

[19] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, "Gradient-based learning applied to document recognition", *Proceedings of the IEEE*, pp. 2278-2324, 2000.

[20] A. Nair, V. Bindu, "A Review on Indian Sign Language Recognition", International Journal of Computer Applications, pp. 33-38, 2013.

[21] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short -term memory." Neural computation 9, no. 8 (1997): 1735¬1780.

[22] Ronchetti, Franco, Facundo Quiroga, César Armando Estrebou, Laura Cristina Lanzarini, and Alejandro Rosete. "LSA64: An Argentinian Sign Language Dataset." In XXII Congreso Argentino de Ciencias de la Computación (CACIC 2016). 2016.

[23] Kingma, Diederik, and Jimmy Ba. "Adam: A method for stochastic optimizations." 2014

[24] Y. Jia. Caffe: An open source convolutional architecture for fast feature embedding. http://caffe.berkeleyvision.org/, 2014.Lifeprint.com. American Sign Language (ASL) Manual Alphabet (fingerspelling) 2007.