



ACM-W 5<sup>th</sup> National Hackathon – 2020

# Hand Talk

## Technocrats



# Team Introduction

## TEAM TECHNOCRATS

Sri Sivasubramaniya Nadar College of Engineering

Lakshmi Priya B



Nivedhitha D



Sakthi Sairaj



# Problem Statement

## 17. Gesture Recognition for Human Computer Interaction

2

- Alternative way of effectively interacting with a computer system by using hand gestures for input and controls
- Simulating the functions of the keyboard alphabets, cursor and common controls like switching slides, etc using hand gesture motions
- Help users interact with the computer from a considerable distance without requiring any device like a mouse or keyboard
- Eliminate or at least decrease the hardware requirement for effective human-computer interaction

# Probable Domains of Impact

Gesture controlled interaction has a plethora of professional, recreational and supportive applications

3

- In the prevailing pandemic situation, it prevents contact with surfaces, for example, billing stations and manufacturing warehouses
- It facilitates interaction with a computer from a considerable distance, for, example, changing slides while taking a presentation
- No need for extra hardware and mimics real actions, heavy machinery operations and game playing

Support for deaf and mute people to communicate with a computer generated voice

A decorative pattern of hexagons in various shades of blue and teal on the left side of the slide. Some hexagons contain icons: a lightbulb, a thumbs up, a smartphone, a magnifying glass, and a gear. A large teal hexagon in the center of this pattern contains the number 4.

# 4

## Implementation Plan

- Plan of Action
- Technology Stack
- Workflow
- Product Pipeline



# Plan of Action

## STEP 1

Neural Network module to load serialized face detector and face embedding model from disc to authorize the user

## STEP 2

Define the classification categories for gestures and collect training images to create a clean dataset

## STEP 3

Train and test the gesture recognition CNN classifier using dlib from scikit-learn and OpenCV

## STEP 4

The gesture selected is used to perform a select action


## STEP 5

The corresponding phrase or keywords associated with the gesture is recognized by the speech module

## STEP 6

The keywords associated with the gestures are converted to complete sentences using BERT


# Technology Stack



Packages or tools	Components used	Purpose
Python 3.9	<ul style="list-style-type: none"><li>• Interpreter</li><li>• Related packages pywin32</li></ul>	Primary language of development of the solution
OpenCV, scikit-learn	<ul style="list-style-type: none"><li>• dnn</li><li>• blobFromImage</li></ul>	<ul style="list-style-type: none"><li>• Deep Neural Network module to load serialized face detector and face embedding model from disc</li><li>• To construct a blob from image</li></ul>
Numpy	Arrays and vectors	Vectorization and image representation
sentence_transformers	Sentence Transformer	For BERT model
Keras	SqueezeNet	Pre-trained CNN Algorithm for classification



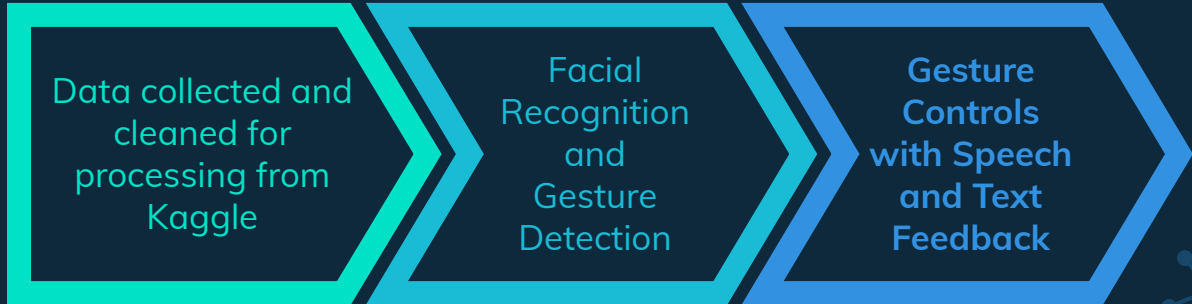
# Workflow

1. Collect live web camera feed
  2. Run face identification and gesture recognition processes parallelly
  3. If the person performing the gestures is not authorized, end both processes
  4. Else, perform the necessary gesture recognition and conversions
- 





# Product Pipeline



Data collected and  
cleaned for  
processing from  
Kaggle

Facial  
Recognition  
and  
Gesture  
Detection

Gesture  
Controls  
with Speech  
and Text  
Feedback

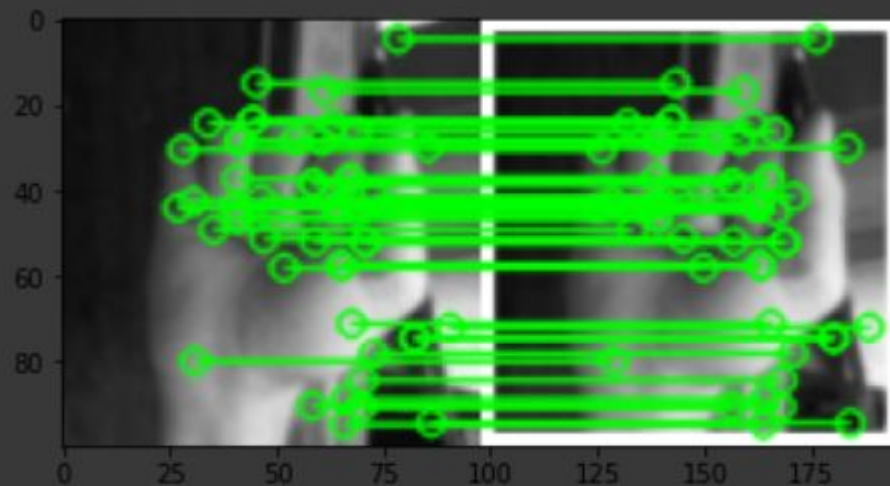
A decorative pattern of hexagons in various shades of blue and cyan on the left side of the slide. Some hexagons contain icons: a lightbulb, a thumbs up, a network diagram, a smartphone, a magnifying glass, a gear, and a speech bubble.

5

## Demo

Code and results obtained during the development lifecycle  
highlighting the successes and bugs

# Approach 1



['a/a\_1.jpg', 'a/a\_1.jpg']

**MATCH!!!!**

# Approach 1

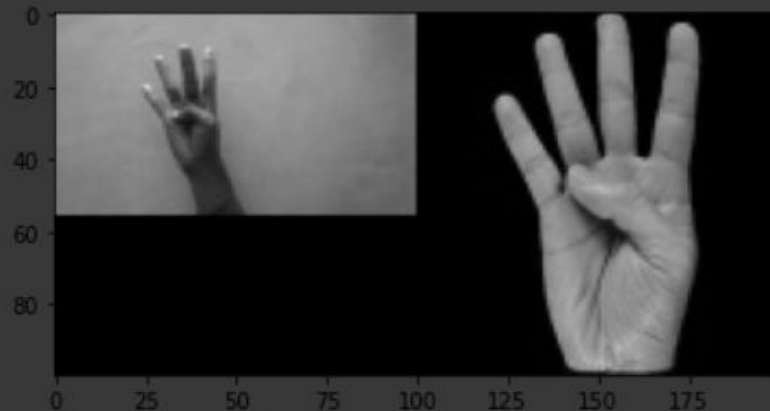
## Problem

Using just a SIFT detector and FLANN matcher didn't deliver significant gesture recognition performance

## Resolution

A CNN or convolutional neural networks algorithm was employed for gesture image classification

Not enough matches are found - 0/20



['b/b\_1.jpg', 'b/b\_100.jpg']

NO MATCH

TESTING

\*\*\*\*\*CONSOLIDATED TESTING REPORT\*\*\*\*\*

Testcase 0 : PASSED: MATCH

Testcase 1 : FAILED: NO MATCH



# Approach 2

- Facial recognition for authorization and security using a SVM
  - To identify and verify the presence and location of a human face in an image and authorize
  - To extract the 128-d feature vectors (called “embeddings”) that quantify each face in an image
- Gesture classification and controls
  - A CNN image classifier takes the video feed and classifies the selected frame into one of the possible gesture categories that it was trained to identify
- The gesture will produce the corresponding action along with speech and text feedback



Spyder (Python 3.7)

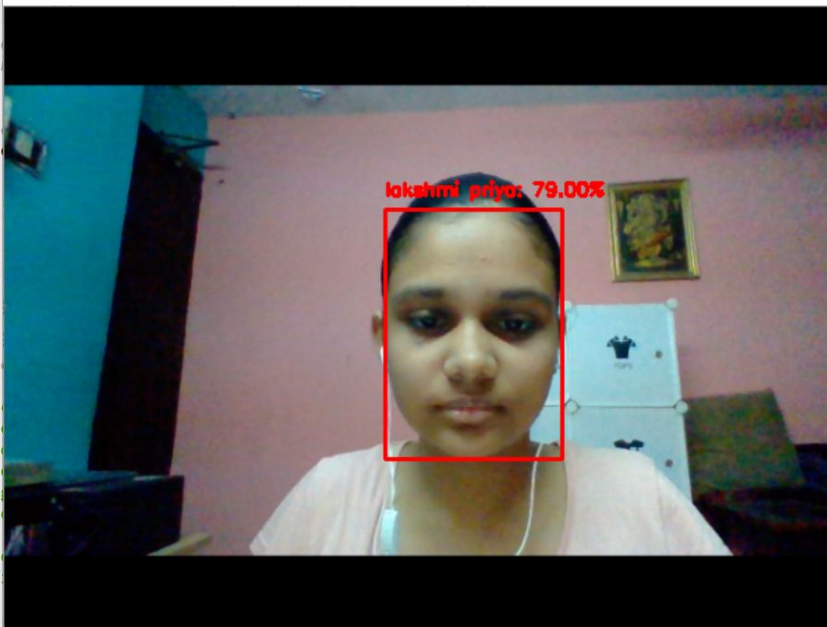
File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\Lakshmi Priya\Documents\Python\opencv-face-recognition\recognize\_video.py

test.py acm.py texttospeech.py

```
1 # USAGE
2 # python recognize_video.py -
3 # --embedding-model openfac
4 # --recognizer output/recog
5 # --le output/le.pickle
6
7 # import the necessary packag
8 from imutils.video import Vid
9 from imutils.video import FPS
10 import numpy as np
11 import argparse
12 import imutils
13 import pickle
14 import time
15 import cv2
16 import os
17
18 # construct the argument pars
19 ap = argparse.ArgumentParser(
20     ap.add_argument("-d", "--dete
21         help="path to OpenCV's de
22     ap.add_argument("-m", "--embe
23         help="path to OpenCV's de
24     ap.add_argument("-r", "--reco
25         help="path to model train
26     ap.add_argument("-l", "--le",
27         help="path to label encod
28     ap.add_argument("-c", "--conf
29         help="minimum probability
30     args = vars(ap.parse_args())
31
32 # Load our serialized face detector from disk
33 print("[INFO] loading face detector...")
34 protoPath = os.path.sep.join([args["detector"], "deploy.prototxt"])
35 modelPath = os.path.sep.join([args["detector"],
```

Frame



Message

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our [tutorial](#)

Variable explorer Help Plots Find

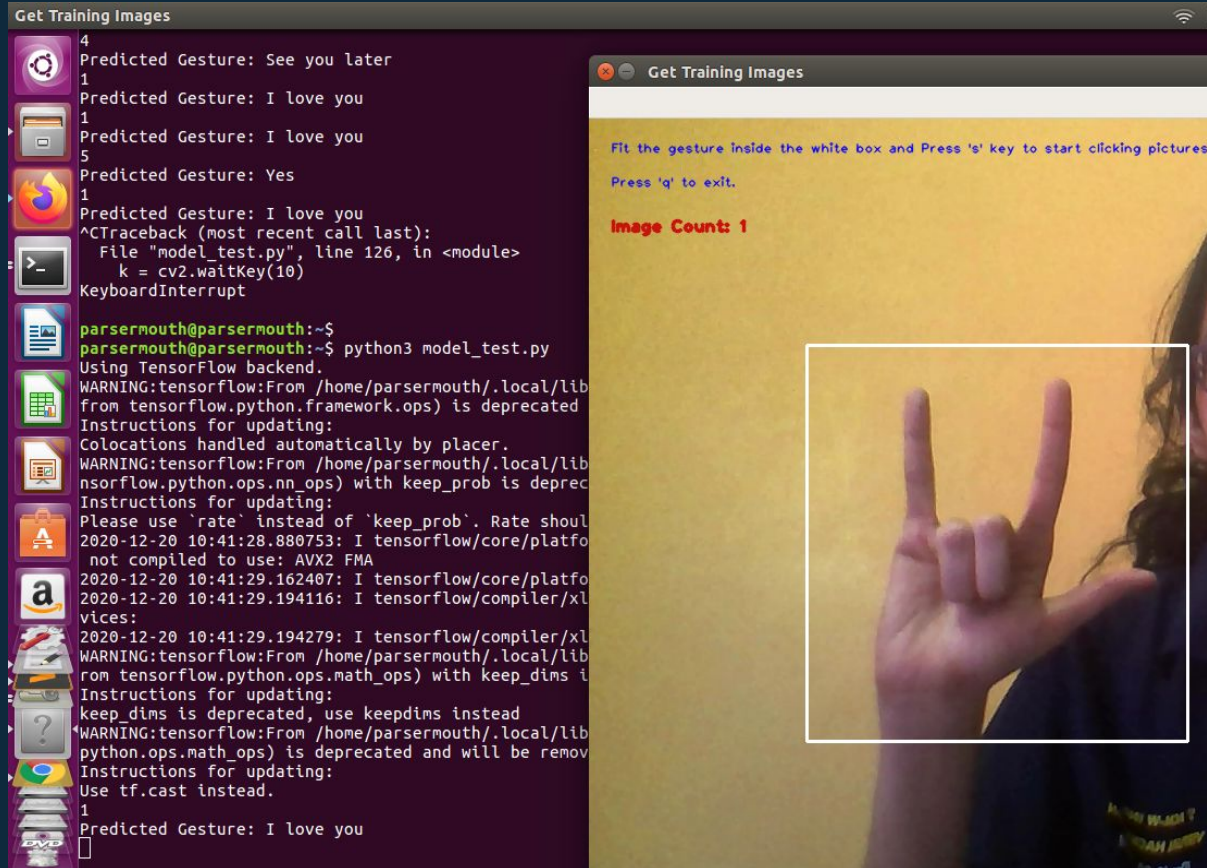
```
recognize_video.py', args='-d face_detection_model
all2.v1.t7 -r output/recognizer.pickle -l output/

e detector...
e recognizer...
riya\AppData\Roaming\Python\Python37\site-packages
06: UserWarning: Trying to unpickle estimator SVC
1 when using version 0.21.2. This might lead to
nvalid results. Use at your own risk.

riya\AppData\Roaming\Python\Python37\site-packages
06: UserWarning: Trying to unpickle estimator
LabelEncoder from version 0.19.1 when using version 0.21.2. This
might lead to breaking code or invalid results. Use at your own risk.
UserWarning)
[INFO] starting video stream...
```

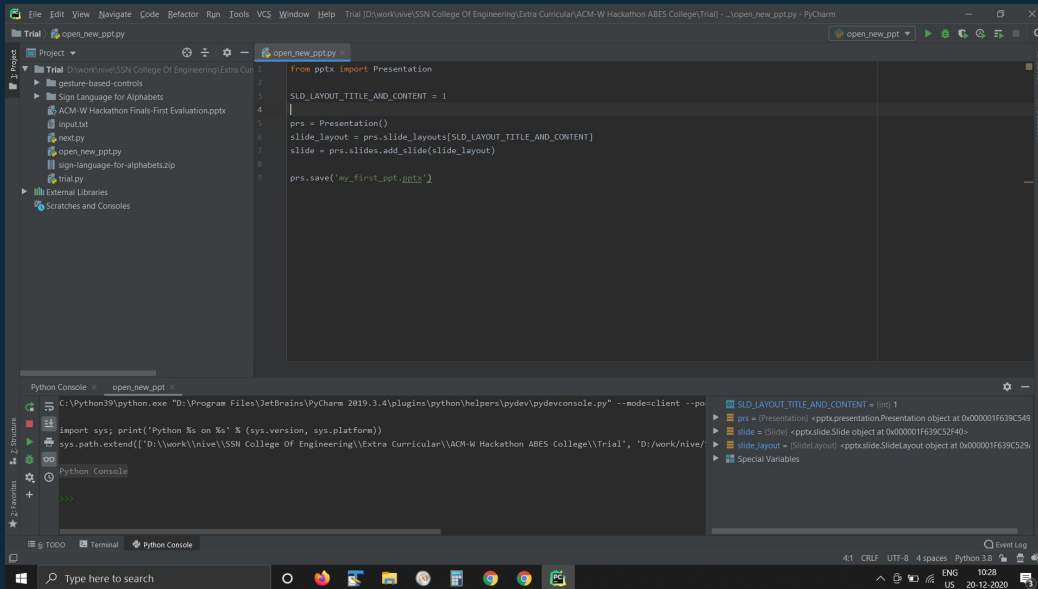
IPython console History

Kite: indexing conda: base (Python 3.7.6) Line 142, Col 10 ASCII LF RW Mem 73%





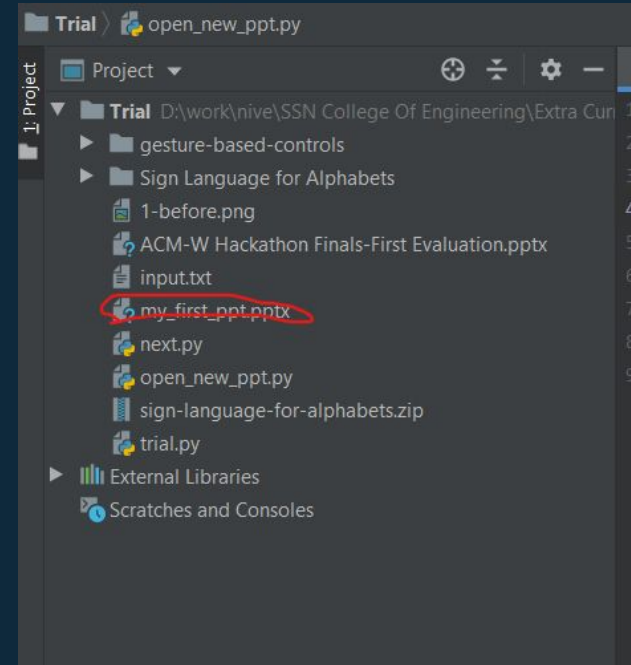
# Code to open a PPT from python



```
1 from pptx import Presentation
2
3 SLD_LAYOUT_TITLE_AND_CONTENT = 1
4
5 prs = Presentation()
6 slide_layout = prs.slide_layouts[SLD_LAYOUT_TITLE_AND_CONTENT]
7 slide = prs.slides.add_slide(slide_layout)
8
9 prs.save('my_first_ppt.pptx')
```

Python Console

```
C:\Python39\python.exe "D:\Program Files\JetBrains\PyCharm 2019.3.4\plugins\python\helpers\pydev\pydevconsole.py" --mode=client --po
import sys; print("Python %s on %s" % (sys.version, sys.platform))
sys.path.extend(['D:\work\nive\SSN College Of Engineering\Extra Curricular\ACM-W Hackathon ABES College\Trial', 'D:\work\nive/
```





# USP and Novelty

6

- Gesture to Speech conversion is still not extensively deployed in the market as support software for the deaf and mute, but, big tech giants have published algorithms for the same
- Our model binds together various well-established modules like face identification, image classification, text to speech conversion, text commands to action stimulation into one cohesive whole and uses all these to create the “Hand Talk” for the disabled

A decorative graphic on the left side of the slide. It features a large cyan hexagon with the number '7' inside. Surrounding this central hexagon are several smaller hexagons of varying shades of blue and cyan. Some of these smaller hexagons contain white icons: a lightbulb, a thumbs-up, a smartphone, a magnifying glass, and a gear. There is also a network-like icon with a central node and several smaller nodes connected by lines.

7

# Scope

Further improvements and other ideas



# Scope

- Using infrared sensors for face recognition
- Cursor movement





# Thank You

Any Questions?

