# GESTURE BASED COMMUNICATION USING PYTHON PROGRAMMING


GUIDED BY:

Mr. RAJESH KUMAR.D
(Assistant Professor)

PRESENTED BY:

VTU21007- PRIYANSI
VTU21484- RAJIV CHAURASIYA
VTU21217- T. Mounika

# ABSTRACT

In a hand sign language-based communication system, concepts are conveyed through a variety of hand gestures. Similar to spoken language, hand signs represent emotions like happiness, sadness, anger, and fear, as well as abstract notions such as time, space, and direction. To interpret these gestures effectively, machine learning algorithms are employed. These algorithms analyze the distinctive features of hand gestures to classify them into meaningful categories. For instance, they can be trained to distinguish between different hand signs representing emotions or abstract concepts. Python programming, utilizing libraries like OpenCV, Keras, and TensorFlow, facilitates the implementation of these machine learning algorithms. These libraries offer robust tools for image, as well as neural network modeling and training, enabling accurate interpretation and communication through hand sign language.

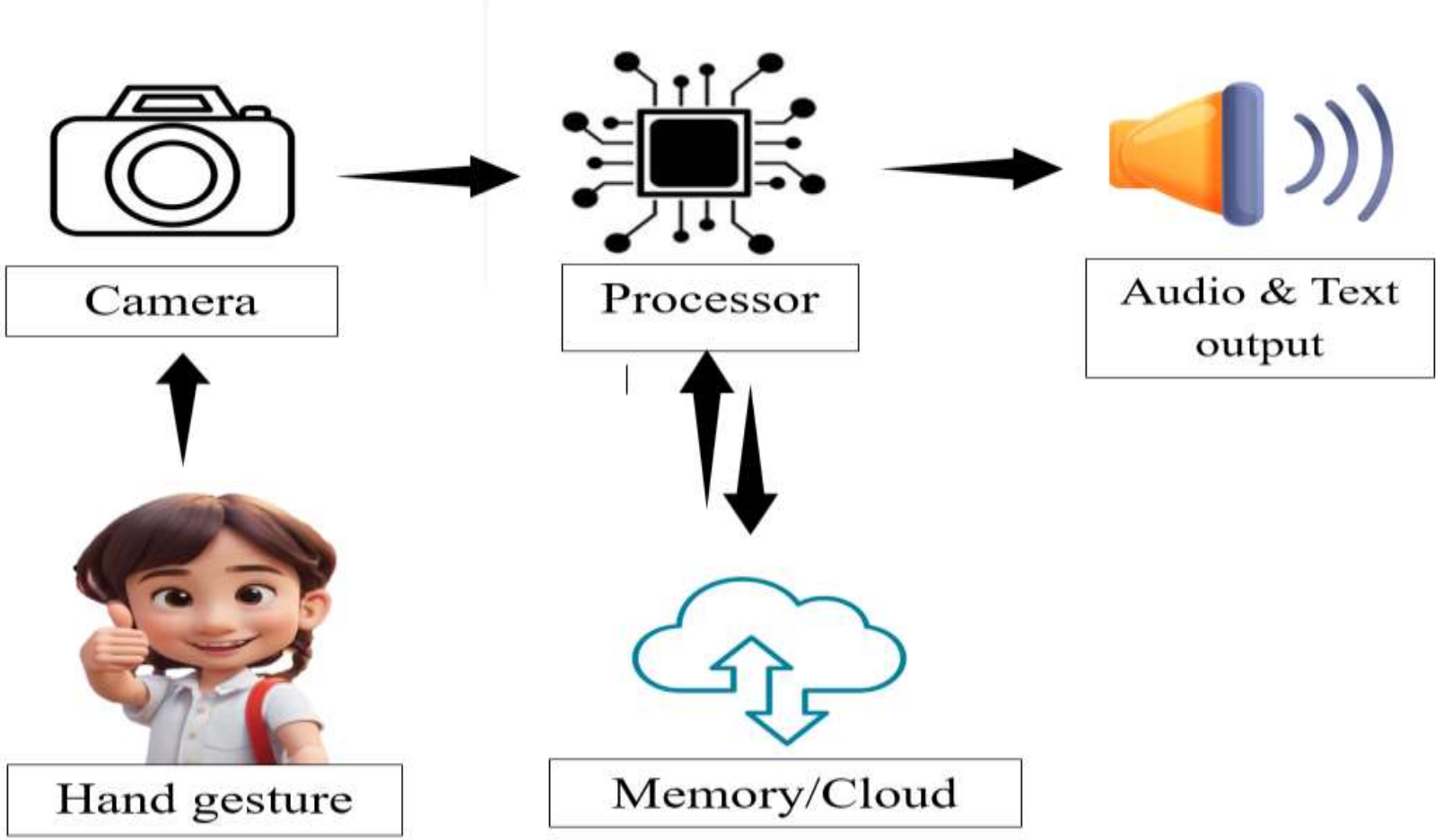
# OBJECTIVE

- The objective is to develop a software which will help mute individuals to communicate to normal people.
- Generally mute people need to learn hand sign-language to express themselves, but normal individuals are not familiar to sign language.
- To address this issue we are developing a software which will convert hand sign gestures to texts and audio signals.

# INTRODUCTION

In a world that thrives on communication, there exists a significant barrier for individuals who are dumb, hindering their ability to connect with the world around them. AAC is an option for nonverbal mode of communication for the persons with communication deficit. In India, 6.3 % of the whole population are sensory impaired so we are developing a software to address this issue. Gesture-based communication systems can be very helpful for people who are dumb. These systems can recognize hand gestures and translate them into text or speech, allowing for more effective communication..

# BLOCK DIAGRAM



# LITERATURE SURVEY

| S. No. | Title of the Paper | Journal name and year | Inference |
|---|---|---|---|
| 1. | Deep Learning-Based Standard Sign Language Discrimination | IEEE Access, Oct 2023 | The primary focus of the study is the usage of RCNN and FPN model with accuracy of 95.5%. |
| 2. | Helping Hearing-Impaired in Emergency Situations: A Deep Learning-Based Approach | IEEE Access, Jan 2022 | This study employs deep learning-based models to accurately predict emergency signs in Indian Sign Language, achieving 82% and 98% prediction accuracies with classification models and 99.6% mean average precision with a YOLO-based object detection model. |
| 3. | Hand Gesture Translation System based on Multi Sensor Fusion | 2023 The 8th International Conference on Integrated Circuits and Microsystems | The Author used CNN and LSTM model anf received accuracy of 97.2% |
| 4. | Real-Time Dynamic Hand Gesture Recognition | IEEE Access, Dec 2014 | This study showcases a real-time hand gesture recognition system, achieving a 85% accuracy rate, through YCbCr color space transformation and OpenCV implementation. |

# METHODOLOGY

1. **Collecting data set:** Gather a diverse and well-labeled dataset of gestures. Include variations in lighting, backgrounds, and hand poses to improve the model's generalization.

1. **Data processing:** Preprocess the collected data, including resizing images, normalizing pixel values, and performing other necessary transformations. Consider techniques like data augmentation to increase the diversity of your dataset.

1. **Model Selection:** Choose a suitable machine learning or deep learning model for gesture recognition. Convolutional Neural Networks (CNNs) , RNN and LSTM are commonly used for image-based tasks. Transfer learning with pre-trained models can be beneficial, especially when working with limited data.

1. **Model training:** Split your dataset into training, validation, and testing sets. Train your chosen model on the training set and fine-tune hyper parameters based on performance on the validation set. Monitor for overfitting and adjust accordingly.7. *Evaluation:*

# TECHONOLOGY USED

## Machine learning

Machine learning can be applied to gesture recognition by training models on datasets of labeled gestures. Algorithms, such as neural networks, can learn patterns and features from these examples to identify and classify different gestures. This enables devices, like cameras or sensors, to interpret and respond to specific hand or body movements, enhancing human-computer interaction in applications like gaming, virtual reality, or touchless interfaces.

## Deep learning

Deep learning in gesture recognition typically involves using neural networks, particularly Convolutional Neural Networks (CNNs) for image-based tasks.

## Image processing

Image processing is a crucial step in gesture recognition, as it helps extract relevant features from input images to enable accurate recognition.

# DATA COLLECTION

Data collection in deep learning for images involves gathering a diverse set of images relevant to the task at hand.

**TECHNOLOGY EMPLOYED:**

1. **Opencv (cv2)**
   OpenCV (Open Source Computer Vision Library) is a powerful open-source library for computer vision and image processing tasks.

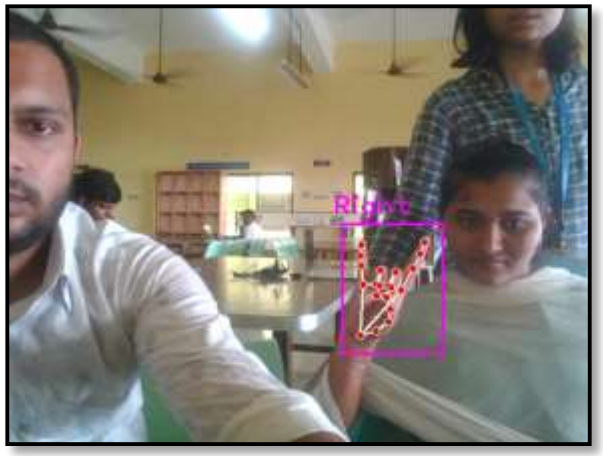
**2.Cvzone**

CVZone offers features like hand tracking, face detection, object recognition, pose estimation, and gesture recognition, among others. These functionalities are often utilized in projects ranging from real-time applications like augmented reality and gesture-based interfaces to tasks such as object detection and tracking in videos.

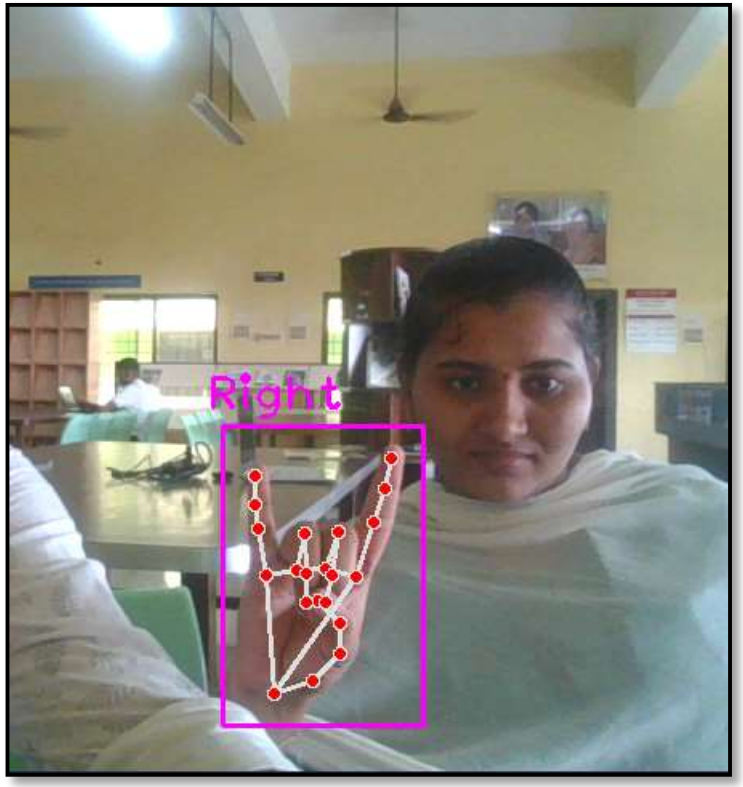
**3. Numpy**

**4.Hand-detector**

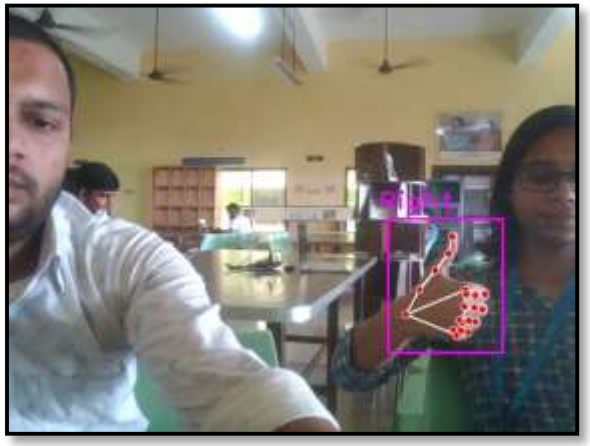
# DATA COLLECTION











# LIBRARY USED

**1.OS**

The Python os module provides a wide range of functions for interacting with the operating system, including file and directory operations, process management, and environment variables.

**2.PANDAS**

Pandas is a powerful and widely used Python library for data manipulation and analysis.

**3.MATPLOTLIB**

Matplotlib is a popular Python library for creating static, interactive, and animated visualizations. It provides a wide range of features and functionalities for creating high-quality plots and charts.

**4.PIEL**

PIL (Python Imaging Library), also known as Pillow, is a Python library that provides extensive capabilities for opening, manipulating, and saving many different image file formats. It is widely used for image processing tasks and provides a range of features for working with images in Python.

# CODE

```
mport cv2
from cvzone.HandTrackingModule
import HandDetector
import numpy as np
import time
capture = cv2.VideoCapture(0)
detector = HandDetector(maxHands=2)
folder = "Data/A"
counter = 0
while True:
 success, img = capture.read()
  if not success:
   print("Failed to read from camera.")
    break
  hands, img = detector.findHands(img)
```

```
if hands:
   if len(hands) >= 2:
     hand1, hand2 = hands[:2]
     x1, y1, w1, h1 = hand1['bbox']
     x2, y2, w2, h2 = hand2['bbox']

         # Crop and save both hands
   hand1_img = img[y1:y1+h1, x1:x1+w1]
  hand2_img = img[y2:y2+h2, x2:x2+w2]
 cv2.imwrite(f'{folder}/
Hand1_{time.time()}.png', hand1_img)
    cv2.imwrite(f'{folder}/
Hand2_{time.time()}.png', hand2_img)
     counter += 2

   elif len(hands) == 1:
       hand1 = hands[0]
     x1, y1, w1, h1 = hand1['bbox']
```

```
# Crop and save the single hand
    hand_img = img[y1:y1+h1, x1:x1+w1]
   cv2.imwrite(f'{folder}/
Hand_{time.time()}.png', hand_img)
      counter += 1

cv2.imshow("Image", img)
key = cv2.waitKey(1)
 if key == ord("S") or key == ord("s"):
  counter += 1
   cv2.imwrite(f'{folder}/Image_{time.time()}.png', img)
 print(counter)
 if key == ord("q"):
    breakcapture.release()cv2.destroyAllWindows()
```

# Program to find label of image (data set)

```
import os
import pandas as pd
from matplotlib.image import imread
import numpy as np
rom PIL import Image
import random
dim1 = []
dim2 = []
for i in range(0,27):
  labels = '/content/h_sign_dataset/rajiv/
Train' + '/{0}'.format(i)
 image_path = os.listdir(labels)
  for x in image_path:
    img = imread(labels + '/' + x)
    dim1.append(img.shape[0])
  dim2.append(img.shape[1])
print("Dimension 1 Mean :
```

```
if hands:
   if len(hands) >= 2:
     hand1, hand2 = hands[:2]
     x1, y1, w1, h1 = hand1['bbox']
     x2, y2, w2, h2 = hand2['bbox']

          # Crop and save both hands
   hand1_img = img[y1:y1+h1, x1:x1+w1]
  hand2_img = img[y2:y2+h2, x2:x2+w2]
 cv2.imwrite(f'{folder}/
Hand1_{time.time()}.png', hand1_img)
    cv2.imwrite(f'{folder}/
Hand2_{time.time()}.png', hand2_img)
     counter += 2

   elif len(hands) == 1:
       hand1 = hands[0]
     x1, y1, w1, h1 = hand1['bbox']
```

# TIMELINE PLAN 

- **Week 1:** Literature review & software selection.
- **Week 2:** Software setup & library integration.
- **Week 3-5:** Algorithm development & code implementation.
- **Week 6-7:** Testing & debugging.
- **Week 8-9:** System optimization.
- **Week 10-11:** Final testing, demonstration & project completion.
- **Week 12:** Preparation of Report

# REFERENCES

**Gesture Recognition Survey:**

https://ieeexplore.ieee.org/document/10310202

**IEE Base Papers:**


1. MENGLIN ZHANG, SHUYING YANG , AND MIN ZHAO (2013). "Deep Learning-Based Standard Sign Language Discrimination".

2. Guozhang Hao1 and Yang Li. (2023). "Hand Gesture Translation System based on Multi Sensor Fusion"

# THANK YOU