# Project Report

# Project Name: Live Human Activity recognition

Group ID :- 3

| NAME | ROLL NO |
|---|---|
| Chauhan Smit Ghanshyambhai | 4085 |
| Fanse Manav Samirbhai | 4096 |
| Chauhan Shikha Dineshbhai | 4163 |

**Supervisor**: Krupali Busa

Department Name : MSc(CA & IT)

College Name : K.S. school of business management

University Name : Gujarat University

## TABLE OF CONTENT

# 1: INTRODUCTION

Live human activity detection refers to the process of identifying and monitoring human movements in real-time.Human activity recognition has received increasing attentions in computer vision and has made significant progress in recent years. The most common approach involves two steps: feature extraction and classification. The aim of this project is to provide an overview of the current state-of-the-art techniques for live human activity detection. We can even use it to train a new employee to correctly perform a task or practice your moves when dancing or working out. Moreover, this technology can be used in many different use cases like gaming controllers, human-robot interactions, and even virtual reality scenarios.

In Python, there are many powerful libraries and tools available for building and training deep learning models, such as TensorFlow, Keras, and OpenCV. These models can be trained on datasets of human images or videos to improve their accuracy. These libraries can be used to create and train models that can recognize human faces, body parts, and even movements. A human recognition project involves several steps, such as preprocessing and cleaning the dataset, building and training the model, evaluating its performance, fine-tuning it, and testing it on a separate dataset to measure its accuracy.

## Important Things About This Project:

- This is a laptop or desktop application.
- In this project we have used LSTM algorithm to develop this project.
- In this project we collect the numpy array to trace the activity and create dataset.
- Application compare the activity detection keypoints to the dataset numpy array and whichever is the most relevant activity, it will appear on the screen.
- We are using mediapipe holistic to extract keypoints from body. We used keras and tensorflow to create LSTM model.

## 2: OBJECTIVES

Live human detection has several objectives, including:

Security: One of the primary objectives of live human detection is to improve security by identifying and tracking human presence in given area. This can be useful in detecting and preventing unauthorized access, theft, or vandalism.

Safety: Another objective is to ensure the safety of individuals in high-risk environments, such as construction sites, hazardous material storage facilities, or areas with potential safety hazards. Live human detection can help to ensure that people are not in harm's way when potentially dangerous situations arise.

Monitoring: Live human detection can be used for monitoring purposes, such as counting the number of people in a specific area, monitoring their movements and behaviors, or tracking crowd dynamics for event management and safety.

Automation: Live human detection can be used to automate various processes, such as turning lights on or off based on human presence, controlling HVAC systems, or adjusting security systems.

Marketing: Live human detection can also be used for marketing purposes, such as analyzing customer traffic patterns in stores to optimize store layouts and improve sales.

Overall, live human detection can provide a range of benefits, including improved security, safety, monitoring, automation, and marketing.

# 3: METHODOLOGY

## Algorithm explaination:

LSTM stands for Long Short-Term Memory, which is a type of recurrent neural network (RNN) architecture used in deep learning. LSTMs are designed to overcome the vanishing gradient problem, which is a common issue in traditional RNNs that makes it difficult to learn long-term dependencies in sequential data.

The basic idea behind LSTMs is to introduce a memory cell that can store information over time and selectively forget or update this information based on input data. The memory cell is composed of three main components: an input gate, a forget gate, and an output gate.

The input gate controls how much new information is added to the memory cell. The forget gate determines how much old information should be removed from the memory cell. The output gate regulates how much information should be passed on to the next time step or the final output.

During each time step, the LSTM receives an input vector and a hidden state vector from the previous time step. The input vector is multiplied by the input gate, and the result is added to the current memory cell state. The forget gate determines which parts of the current memory cell state to keep and which parts to discard. The resulting memory cell state is then passed through the output gate to generate the output vector for the current time step.

Overall, the LSTM algorithm is a powerful tool for processing sequential data, such as natural language text or time series data. Its ability to store and selectively update information over time makes it particularly useful for tasks that require long-term dependencies, such as language translation, speech recognition, and sentiment analysis.

## Approach To The Problem:

1. Extract holistic keypoints
2. Collect keypoints and values for training and testing
3. Build and train LSTM neural network
4. Make predictions
5. Evaluation using confusion matrix and accuracy

## Inbuilt libraries Used In Project:

Opencv: opencv is a computer vision library that allows us to work with your webcams and makes it little bit easier.

Mediapipe : To extract keypoints from human body.

Scikit-learn : we used this for evalution metrics and leverage a training and testing split.

Numpy: It helps us to work with arrays.

Os: To work with file paths.

Keras: It is used to make implementation of neural networks easily.

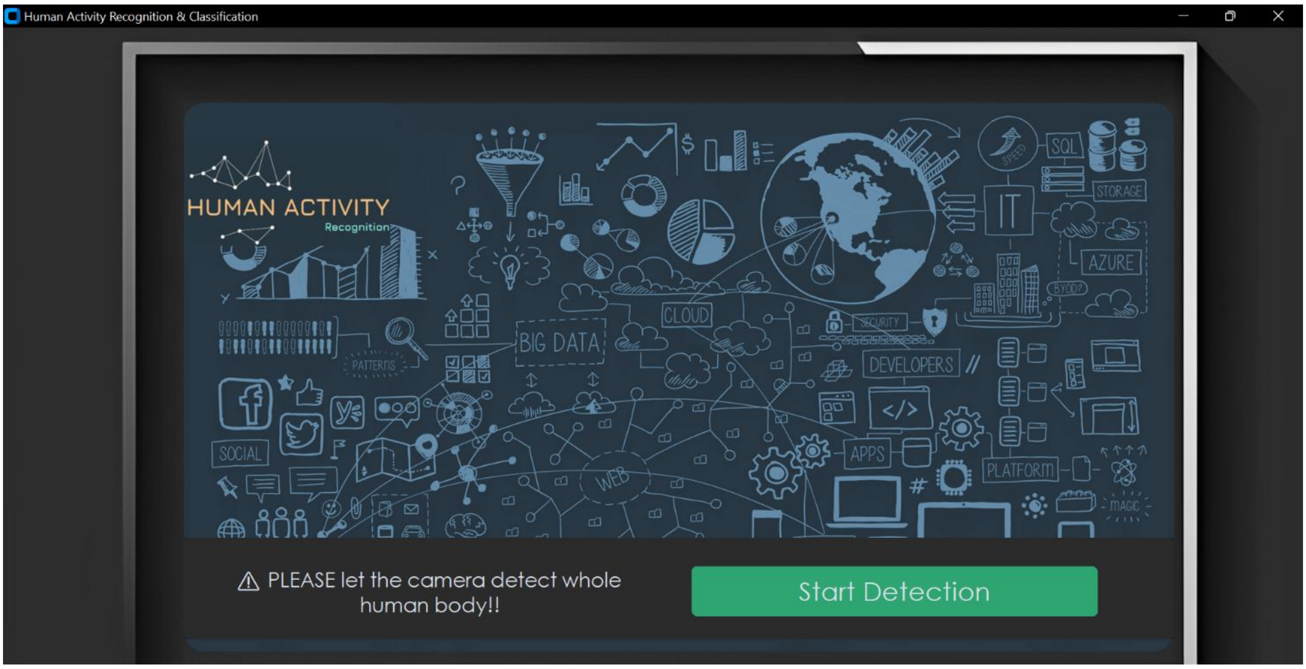
Tkinter: This library is used to create GUI.

PIL: It deals with images.

# 4: INPUT AND OUTPUT

## INPUT:

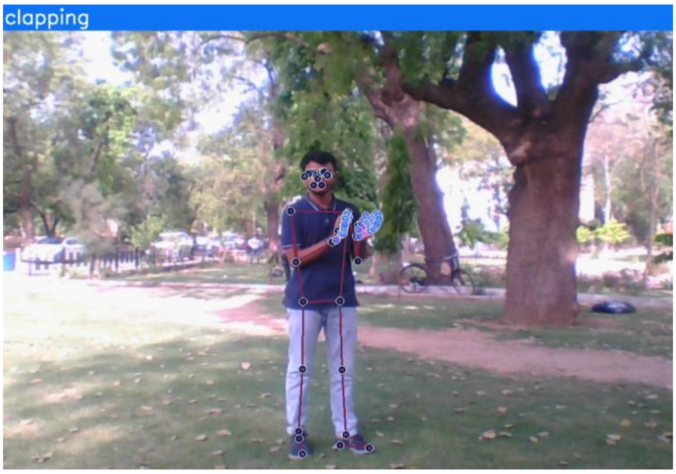
First welcoming interface.

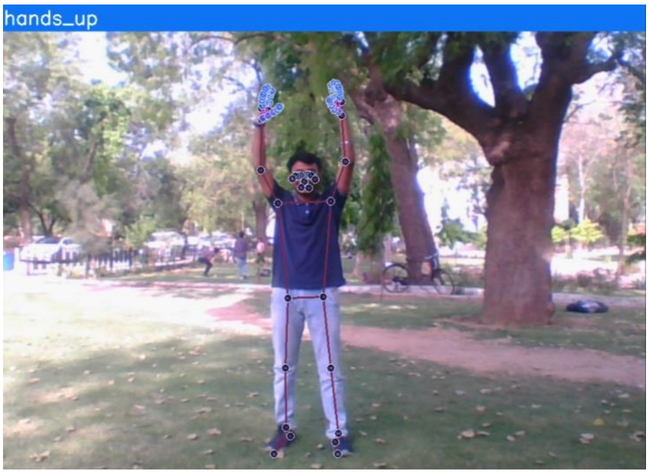


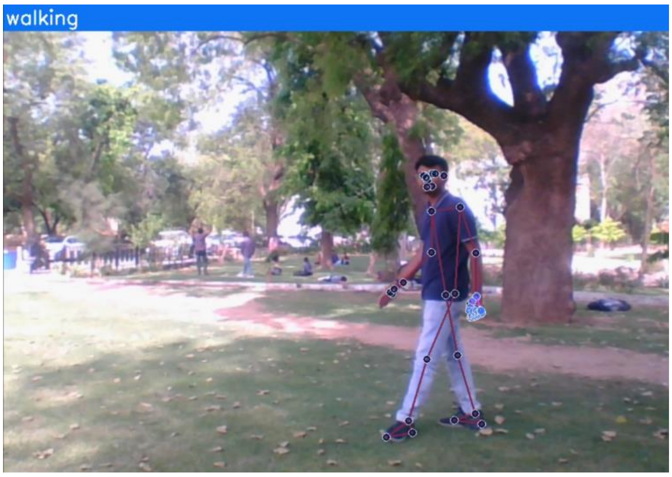
⚠ PLEASE let the camera detect whole human body!!

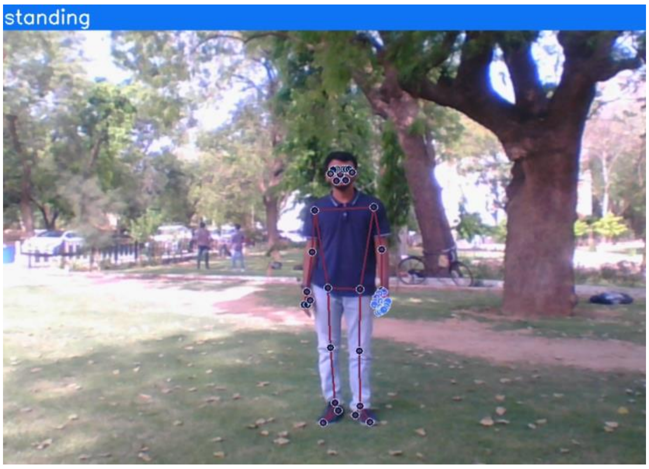
Start Detection

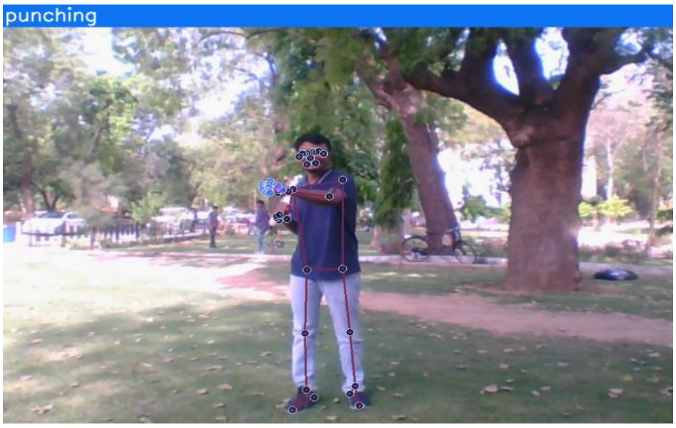
We took keypoints of the human activity as input.

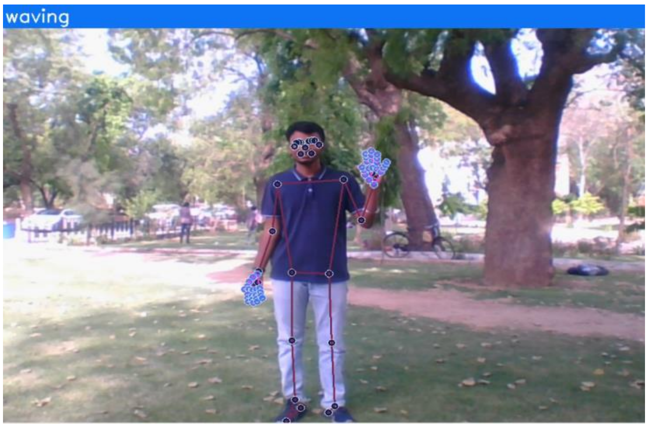






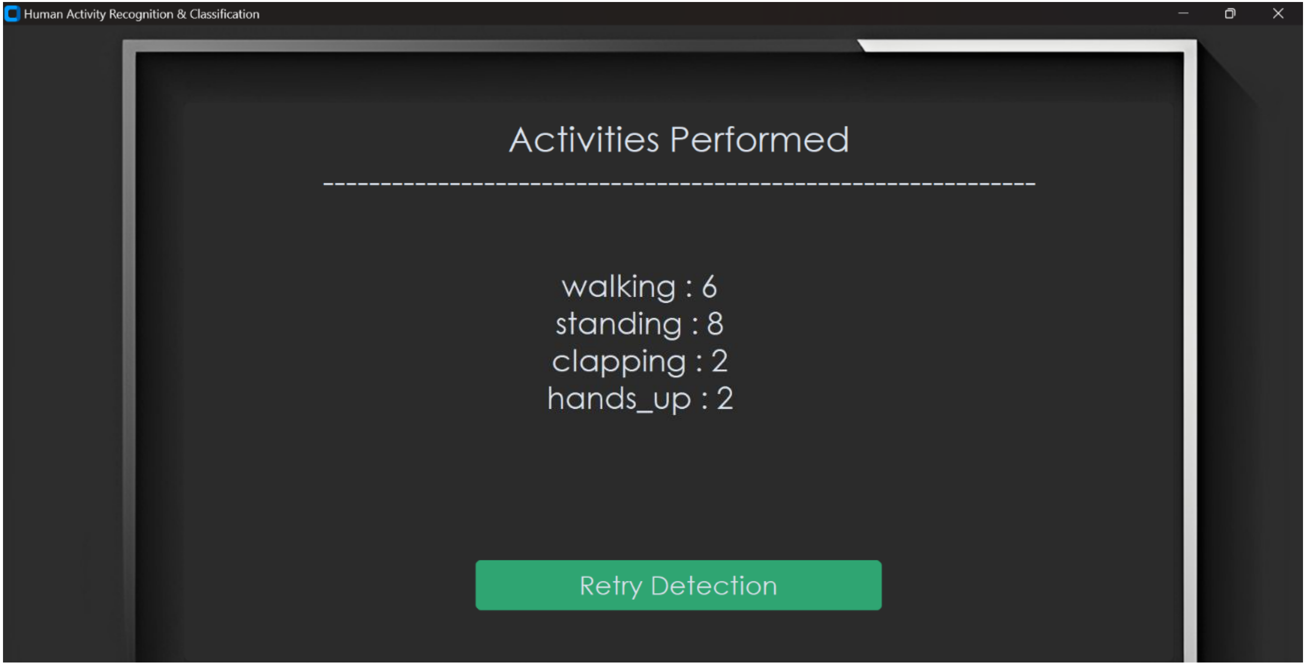






## OUTPUT

This screen will shows that the how many times each activity is performed. Click button retry detection to start process again.



Activities Performed

---------------------------------------------------------------

walking : 6
standing : 8
clapping : 2
hands_up : 2

## 5: BIBLIOGRAPHY

- Machine Learning: Step-by-Step Guide To Implement Machine Learning Algorithms with Python
  by Rudolph Russell

- Python Programming. Python Programming for Beginners, Python Programming for Intermediates
  by Adam Stewart

# 6: CONCLUSION

## Things That Have Been Learned:

- concept of heuristic keypoints.
- how to use AI algorithm for activity detection.
- Adding multiple screens in a single project.
- Use of inbuilt python libraries according to project requirement.

## Future Work:

- As of now we have created seven different activities in this project. which we will develop till 100 activities.
- If any unusual activity is tracked in the system, it will capture that video and will give some alert.

# 7: REFERENCES

https://youtu.be/doDUihpj6ro

https://github.com/nicknochnack/ActionDetectionforSignLanguage.git

https://www.python.org/

https://www.github.com/