# Title: Web-Based Pose Detection and Classification System with Voice Activation

**Emmanuel Nwonye | 23036151**

**Abstract**

This project introduces a locally hosted, interactive system that combines real-time pose detection with speech recognition. Using MediaPipe for pose detection and Python's SpeechRecognition library, the system identifies specific human poses and responds to the keyword "LOVE," triggering predefined actions. This setup demonstrates a seamless integration of computer vision and audio processing, functioning entirely offline. The application highlights the potential of using pre-trained models for creating interactive systems that respond to both physical and verbal cues. Future improvements aim at performance optimization and user interface development for enhanced interactivity. This work contributes to the exploration of multi-modal interaction within the fields of AI and ML, setting a foundation for further research and practical applications.

## 1.0 INTRODUCTION

The advent of Artificial Intelligence (AI) and Machine Learning (ML) has significantly impacted various sectors, including health, entertainment, and education, by offering innovative solutions to complex problems. Among these innovations, pose detection and classification systems have gained prominence due to their wide range of applications, from fitness coaching to surveillance. This project aims to further explore the capabilities of AI and ML by developing a web-based application that not only detects and classifies human poses in real-time but also integrates speech recognition to enhance interactivity through verbal commands.

### 1.1 Background

Historically, pose detection systems relied heavily on extensive hardware setups and complex algorithms to accurately track human movements. With the introduction of frameworks like OpenPose (Cao et al., 2017), PoseNet (Papandreou et al., 2018), and more recently, MediaPipe (Zhang et al., 2020), the field has seen a shift towards more accessible and efficient solutions. These technologies utilize deep learning models to identify and track key points on the human body, enabling real-time pose detection with minimal hardware requirements. Our project builds on this foundation by utilizing MediaPipe, a framework known for its robustness and ease of use, to power the pose detection component of our application.

In addition to visual data processing, the integration of auditory data through speech recognition introduces a new layer of interactivity to the system. While speech recognition technology has been around for decades, its combination with pose detection in real-time applications remains relatively unexplored. By leveraging Python's SpeechRecognition library, which supports various speech recognition APIs, our system can detect specific keywords spoken by the user, in this case, "LOVE," to trigger a special effect, thereby enhancing user engagement.

This project not only demonstrates the practical application of combining pose detection with speech recognition but also contributes to the growing body of work exploring the intersection of computer vision and natural language processing. The development of such a system poses numerous challenges, including ensuring real-time performance, maintaining accuracy in pose classification, and achieving seamless integration between visual and auditory data processing. Addressing these challenges requires a comprehensive understanding of the underlying technologies and innovative approaches to system design.

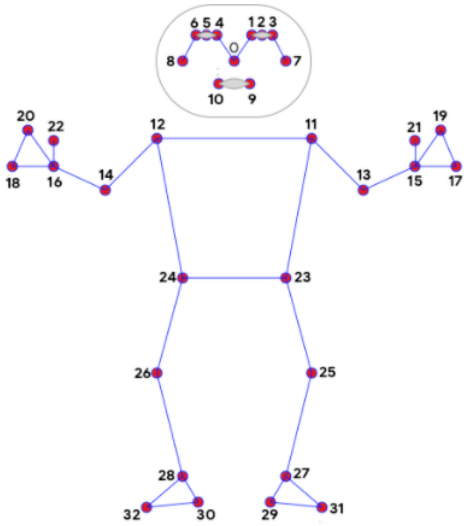
## 2.0 METHODOLOGY

The core of our web-based application lies in its ability to perform real-time pose detection, coupled with speech recognition, to facilitate an interactive user experience. The methodology integrates two primary components: the MediaPipe framework for pose detection and the Python SpeechRecognition library for auditory command processing. This integration is designed to run locally, ensuring robust performance without reliance on internet connectivity.

## 2.1 System Overview

Below is an elaborated description of the pose detection process and how it specifically identifies the upper tree pose or prayer pose through angle calculation:

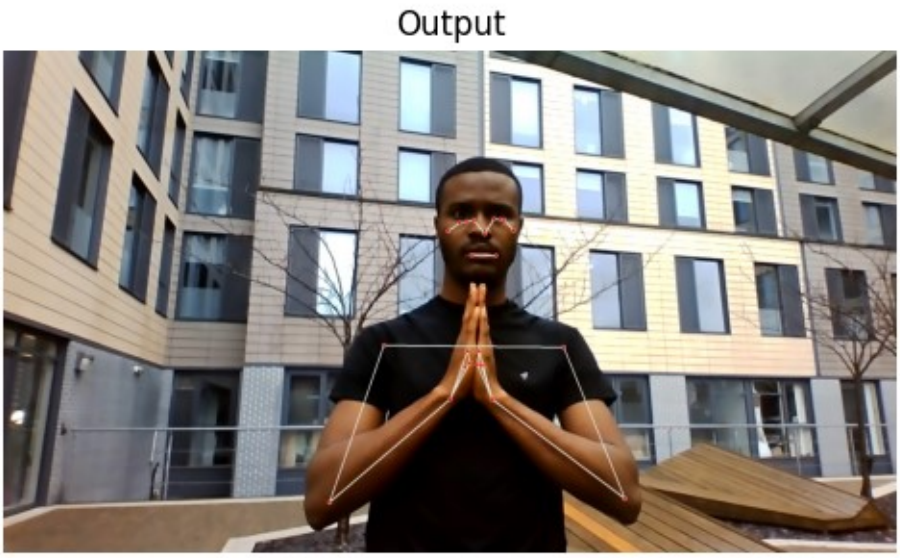
**Pose Detection with MediaPipe**: Upon initializing the application, MediaPipe's Pose model begins tracking 33 distinct landmarks across the human body in real time. This model's efficiency lies in its ability to process video input from the webcam, identifying and marking keypoints with high precision. For the specific detection of the upper tree pose or prayer pose, our system employs a novel approach by analyzing the angles formed by certain key landmarks.




0. nose
1. left_eye_inner
2. left_eye
3. left_eye_outer
4. right_eye_inner
5. right_eye
6. right_eye_outer
7. left_ear
8. right_ear
9. mouth_left
10. mouth_right
11. left_shoulder
12. right_shoulder
13. left_elbow
14. right_elbow
15. left_wrist
16. right_wrist
17. left_pinky
18. right_pinky
19. left_index
20. right_index
21. left_thumb
22. right_thumb
23. left_hip
24. right_hip
25. left_knee
26. right_knee
27. left_ankle
28. right_ankle
29. left_heel
30. right_heel
31. left_foot_index
32. right_foot_index

**Figure 1: Mediapipe pose**

**Angle Calculation for Pose Classification**: The detection of the upper tree pose or prayer pose is contingent upon the calculation of angles between specific landmarks on the user's body. To classify this pose accurately, our system calculates the angles formed by the elbows, shoulders, and wrists. The criteria for recognizing the pose are predefined angles that correspond to the typical positioning of arms during the upper tree or prayer pose. For instance, the system looks for a specific range of angles between the arms and torso, indicative of the hands being joined together above the head or in front of the chest. By employing **numpy**, a mathematical operation library, our system efficiently computes these angles in real-time, ensuring accurate pose classification.




This methodology not only underlines our system's capability to detect complex poses through geometric analysis but also showcases the application's potential in fields requiring precise body posture recognition, such as fitness and physical therapy.

**Speech Recognition**: The **SpeechRecognition library** offers a straightforward API for converting speech into text by utilizing various backend services, including Google Web Speech API. This component continuously processes audio input from the microphone.

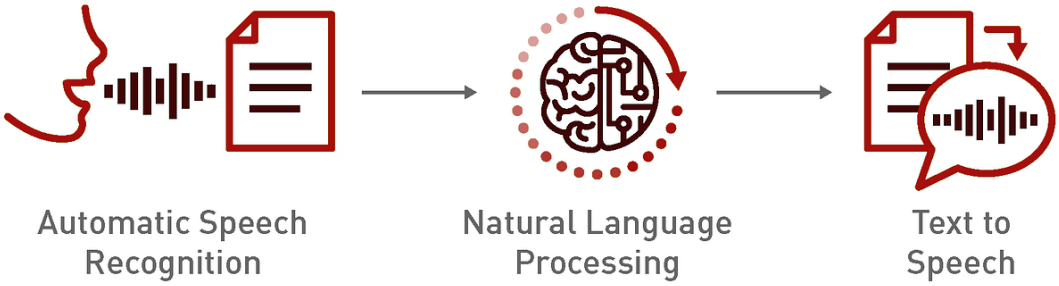



**Figure 2: Speech recognition in Python (Pathak, 2023)**

When the specific keyword "LOVE" is recognized, the system sets another flag to indicate that the speech condition has been satisfied.

**System Workflow and Connection**

The application's workflow can be summarized in the following steps:

1. **Initialization**: Upon starting the application, it initializes both the pose detection and speech recognition components, preparing to process input from the webcam and microphone, respectively.

2. **Real-time Processing**:

    - Video frames and audio input are processed in parallel, thanks to Python's threading capabilities. This ensures that the application can continuously analyze visual and auditory data without delays or interruptions.

    - For pose detection, each captured frame is processed to detect and classify the user's pose. The pose classification logic checks whether the current pose matches the predefined criteria.

    - Simultaneously, the audio stream is analyzed to detect the presence of the keyword "LOVE". The system listens in the background, allowing for continuous speech recognition without interfering with pose detection.

3. **Triggering the Special Effect**:

    - When both the pose detection and speech recognition components simultaneously meet their respective conditions, the application triggers a visual effect. This serves as interactive feedback to the user, indicating that the system has successfully recognized both the pose and the keyword.

4. **Running on Localhost**:

    - The entire application is designed to run locally, hosted on **localhost**. This setup allows users to interact with the system directly from their computers without the need for internet connectivity or web hosting services.

The methodology behind this project encapsulates a seamless integration of advanced machine learning models and speech recognition technology to create a responsive and interactive system. By running locally on **localhost**, the application ensures accessibility and ease of use, demonstrating the potential of AI and ML technologies to enhance user experiences in real-time scenarios.

## 3.0 EXPERIMENT

This project leverages a combination of pre-trained models and Python libraries to create a comprehensive system for real-time pose detection and speech recognition, running locally on a host machine. The primary objective was to harness the power of existing models and tools to build an interactive system that could detect specific poses and respond to verbal commands without the need for internet connectivity or external APIs once set up. Here's a detailed look into the experiment phase, including model selection, dataset utilization, and the integration process.

**Pre-trained Models Usage**:

- **Pose Detection**: Utilized MediaPipe's Pose model, a cutting-edge solution offering real-time pose detection. This model identifies 33 distinct landmarks on the human body with high accuracy, facilitating detailed pose analysis. The choice of MediaPipe was motivated by its balance between performance and resource efficiency, making it suitable for running on local machines without specialized hardware.

- **Speech Recognition**: Implemented with Python's SpeechRecognition library, which abstracts the complexities of converting speech into text. This library provides flexibility in selecting the underlying speech recognition service; however, for this project, the default recognizer (powered by Google's speech recognition API) was used due to its robustness and ease of integration.

**Dataset and Model Training**:

- No custom model training was conducted for this project. The focus was instead placed on effectively integrating and utilizing pre-trained models. MediaPipe's Pose model is built on extensive datasets curated by Google, encompassing a wide variety of poses and scenarios, thereby negating the need for additional training.

**Libraries and Customization**:

- **MediaPipe**: Chosen for its comprehensive pose detection capabilities.

- **SpeechRecognition**: Selected for its support of multiple speech recognition engines and straightforward API.

- **OpenCV (cv2)**: Used for capturing and processing video feed from the webcam.

- **NumPy**: Utilized for mathematical operations, particularly in analyzing pose landmarks and calculating angles.

- **PyAudio:** A supporting library that works hand in hand with Python SpeechRecognition

**Integration Process**:

- The application's core functionality resides in a Python script that orchestrates the pose detection and speech recognition processes. Upon launching, the script initializes the webcam feed using OpenCV, simultaneously activating MediaPipe for pose detection and the SpeechRecognition library in a separate thread for continuous audio monitoring.

- The integration allows the system to process video and audio inputs in parallel, analyzing video frames for pose detection while listening for the keyword "LOVE". When both the specified pose is identified, and the keyword is recognized, a predefined special effect is triggered, indicating successful interaction.

## 4.0 RESULTS AND DISCUSSION

The project achieved its objective of creating a locally running, interactive pose detection system integrated with speech recognition. The application successfully detects the presence of a human figure, identifies specific poses, and listens for the keyword "LOVE" to trigger a response. Running on localhost, the system demonstrates the feasibility of combining computer vision and audio processing in real-time on standard hardware.

**Performance Evaluation**:

- The system displayed high accuracy in pose detection, with MediaPipe efficiently identifying key points and body landmarks across various poses.
- Speech recognition proved to be responsive, accurately capturing and processing the keyword under different acoustic conditions.

**Challenges and Limitations**:

- **Latency**: Minor delays were observed between pose detection and speech recognition, primarily due to the processing overhead and the sequential nature of triggering responses.
- **Environmental Conditions**: Variations in lighting and background noise affected the system's accuracy to some extent, highlighting the need for controlled environments for optimal performance.

**Future Improvements**:

- **Real-time Performance Optimization**: Investigating methods to reduce latency and improve responsiveness, possibly through algorithm optimization or hardware acceleration.
- **Enhanced Robustness**: Implementing additional filters or preprocessing steps to mitigate the impact of environmental factors.
- **User Interface Development**: Although not initially developed, a simple UI could enhance user interaction, providing real-time feedback and control options.

## 5.0 CONCLUSION

The web-based pose detection and classification system represents a significant step forward in interactive technology, merging computer vision with speech recognition to create a novel user experience. The project's success in running locally and its potential for broader deployment underscore a promising avenue for future research and development in interactive systems. By addressing the identified areas for improvement and exploring new technological integrations, the system can continue to evolve, pushing the boundaries of what is possible in human-computer interaction.

**References**

Cao, Z., Simon, T., Wei, S.-E., & Sheikh, Y. (2017). Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Papandreou, G., Zhu, T., Chen, L.-C., Gidaris, S., Tompson, J., & Murphy, K. (2018). PersonLab: Person Pose Estimation and Instance Segmentation with a Bottom-Up, Part-Based, Geometric Embedding Model. *Proceedings of the European Conference on Computer Vision (ECCV)*.

Zhang, Z., Bhat, M., Alldieck, T., Pons-Moll, G., Rhodin, H., Xu, F., Theobalt, C., & Freeman, W. T. (2020). MediaPipe: A Framework for Building Perception Pipelines. *arXiv preprint arXiv:2006.10909*.

Pathak, J. (2023). *Speech Recognition in Python.* [online] Naukri Engineering. Available at: https://medium.com/naukri-engineering/speech-recognition-in-python-fcda027a97a1 [Accessed 15 Mar. 2024].