Human Pose Estimation using Machine Learning

A Project Report

submitted in partial fulfillment of the requirements

of

AICTE Internship on AI: Transformative Learning
with
TechSaksham – A joint CSR initiative of Microsoft & SAP

by

Sahana Priya G, sahanacseai@gmail.com

Under the Guidance of

P Raja- Master Trainee, Edunet Foundation

ACKNOWLEDGEMENT

We would like to take this opportunity to express our heartfelt gratitude to everyone who has supported and guided us throughout the completion of this project. Without their help, this endeavor would not have been possible.

First and foremost, we extend our deepest gratitude to our Master trainer **P Raja** sir, for their invaluable guidance, encouragement, and support throughout the project. Their constructive feedback, innovative ideas, and unwavering confidence in our abilities have been a driving force behind the successful completion of this work. Their mentorship not only contributed to the technical aspects of this project but also helped us grow as responsible and thoughtful individuals.

We are also grateful to institution, Edunet Foundation, for providing the resources, tools, and environment necessary to carry out this project. We acknowledge the efforts of our faculty members for their insightful lectures and hands-on guidance, which laid a strong foundation for this work.

Furthermore, we extend our appreciation to our peers and friends, who have always been a source of motivation and collaboration throughout this journey. Their constant encouragement and willingness to assist during challenging times have been truly valuable.

Last but not least, we owe our deepest gratitude to our families for their constant love, support, and understanding. Their belief in our abilities has always inspired us to strive for excellence and stay committed to our goals.

This project is a testament to the collective efforts and guidance of everyone who has been part of this journey, and we remain profoundly thankful to all who have contributed to its success.

ABSTRACT

Human pose estimation is a vital problem in computer vision which has numerous real-world usage in fields such as healthcare, sports analysis, virtual and augmented reality, and human-computer interface. The goal of the project is to create a high-performance ML-based tool to recognize the position of human key body points from the input images and video feed to estimate human poses.

Therefore, the aim of this work is to develop an algorithm that could identify significant structures, like joints and limbs, and extract human posture under practical conditions. The presented methodology relies on MediaPipe, a deep learning framework designed to estimate human pose. The workflow includes image and video pre-processing, MediaPipe Pose module, and the results' post-processing to obtain valuable information. Also, to improve visualization an extension was added to animate the recovered skeletons and laid down the groundwork for further analysis.

The system showed great ability in recognizing key points in different scenarios such as low and high lighting, occlusions and dynamic motions. By subjecting the movements to visualization of the skeletal structure, the model preserved human motion characteristics and was useful in analyzing posture and activities.

Thus, the project accurately delivers the concept of the human pose estimation and demonstrates the usage of the machine learning method. Possible extensions include further optimizing the proposed model for particular tasks that would involve its application, for example in sports scenario or medical diagnosis, as well as extension of the engine for being able to handle real time data in the context of interactive applications. This project demonstrates the possibility of machine learning for progression of Human-Centered AI.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

1.1 Problem Statement

- Improper Exercise Posture: Incorrect exercise postures lead to ineffective workouts and increase the risk of injuries, especially for individuals exercising without professional supervision.
- Limitations of Traditional Methods: Personal trainers and marker-based motion tracking systems, while effective, are expensive and inaccessible for many.
- Technological Opportunity: Advancements in computer vision and machine learning provide tools for automating posture detection and feedback systems through Human Pose Estimation (HPE).
- Challenges in Existing Solutions: Current HPE systems face challenges such as:
  - Handling diverse body types and dynamic movements.
  - Operating effectively in real-time with minimal computational resources.
  - Overcoming environmental factors like lighting or background clutter.
- Gap in Accessibility: Many systems require high-end hardware or expertise, limiting their usage in personal fitness or low-resource environments like smartphones.
- Proposed Solution:
  - Develop a real-time human exercise posture detection system using OpenCV and MediaPipe.
  - Leverage MediaPipe's pose estimation framework for detecting key body landmarks efficiently.
  - Utilize OpenCV for video processing to ensure the system is accessible and cost-effective.
- Goal: Bridge the gap between technological advancements and practical implementation, enabling accurate posture detection and meaningful real-time feedback for fitness enthusiasts and rehabilitation patients across diverse scenarios.

1.2 Motivation

- Need for Proper Exercise Form: Maintaining correct exercise posture is essential to maximize the effectiveness of workouts and minimize the risk of injuries, particularly for beginners and those exercising without professional supervision.
- Accessibility Challenges: Traditional methods, such as hiring personal trainers or using marker-based motion capture systems, are costly, inaccessible to many, and impractical for daily use.
- Advancements in Technology: The rise of computer vision and machine learning has made it possible to automate posture detection and feedback systems, offering a scalable and efficient solution for fitness monitoring.
- Popularity of Fitness Tracking: The increasing adoption of wearable devices and fitness apps highlights the demand for accessible, real-time exercise guidance systems that go beyond simple activity tracking.
- Efficiency of MediaPipe: MediaPipe Pose provides a highly efficient and lightweight framework for real-time pose detection, even in resource-constrained environments like mobile devices, making it an ideal candidate for this project.
- Applications in Rehabilitation: Accurate posture detection is not only beneficial for fitness but also has significant implications in physical therapy and rehabilitation, aiding patients in recovering effectively and safely.
- Encouraging Healthy Lifestyles: With the growing emphasis on health and wellness, a system that provides real-time feedback on exercise form can motivate users to maintain consistency in their fitness routines.
- Bridging the Gap: Existing solutions often lack real-time feedback or struggle with diverse exercise routines, dynamic movements, or varied body types. This project aims to address these limitations and bring accessible, effective posture detection to a broader audience.

- Scalable Solution for All: By combining OpenCV and MediaPipe, the project aims to create a robust, low-cost, and user-friendly system that can operate across diverse environments, making exercise guidance available to a global audience.

1.3 Objective

- Develop a Real-Time System: Build a real-time human exercise posture detection system using OpenCV and MediaPipe to provide instant feedback during workouts.
- Ensure Accuracy in Posture Detection: Utilize MediaPipe's pose estimation framework to identify key body landmarks (33 points) and analyze exercise postures with high precision.
- Enhance Accessibility: Create a solution that is cost-effective and capable of running efficiently on resource-constrained devices like smartphones and low-power computers.
- Address Key Challenges:
  - Overcome issues such as occlusion, diverse body types, and varying exercise routines.
  - Ensure robustness against dynamic movements and environmental factors like lighting and background noise.
- Provide Meaningful Feedback: Implement logic for real-time posture classification and corrective feedback to help users improve their exercise form and reduce the risk of injuries.
- Promote Health and Fitness: Enable users to adopt healthier lifestyles by providing a virtual assistant for fitness training and posture correction, eliminating the need for professional trainers.

- Support Rehabilitation and Therapy: Design the system to aid in physical therapy by tracking and analyzing body movements for patients undergoing rehabilitation exercises.

- Scalable for Diverse Applications: Ensure the solution is flexible enough to be adapted for various use cases, including fitness apps, sports performance analysis, and healthcare monitoring systems.

- Leverage OpenCV and MediaPipe: Utilize OpenCV for video capture and preprocessing, while leveraging MediaPipe for efficient and accurate pose estimation, ensuring an optimized and streamlined workflow.

- Bridge the Technology-Practice Gap: Combine state-of-the-art machine learning techniques with practical implementation to deliver a solution that meets both user needs and technological capabilities.

1.4 Scope of the Project

Core Scope

- Real-Time Exercise Posture Detection: Develop a system capable of analyzing video input in real-time to detect exercise postures accurately using MediaPipe and OpenCV.
- Keypoint Detection: Leverage MediaPipe's pose estimation framework to identify 33 key body landmarks, ensuring precise tracking of body movement and alignment.
- Posture Classification: Implement a classification mechanism to identify and differentiate between correct and incorrect exercise postures.

Applications

- Personal Fitness Training: Provide real-time feedback to individuals performing exercises at home or in gyms, improving workout efficiency and form.
- Rehabilitation and Physical Therapy: Support healthcare professionals by offering a tool to monitor and guide patients during rehabilitation exercises.

- Sports Training and Performance Analysis: Assist athletes in analyzing their movements to enhance performance and reduce injury risks.
- Fitness Apps Integration: Enable seamless integration with mobile fitness applications to deliver posture detection and correction features.

Technological Scope

- Hardware Accessibility: Design the system to operate on resource-constrained devices such as smartphones and laptops, making it accessible to a broader audience.
- Cross-Platform Functionality: Ensure the solution is compatible with different operating systems, including Android, iOS, and Windows.
- Scalability: Create a framework that can be extended to detect multiple poses or support various types of exercises beyond the initial scope.

Challenges Addressed

- Environmental Robustness: Ensure the system performs reliably under varying lighting conditions, backgrounds, and body types.
- Dynamic Movement Analysis: Handle the complexities of real-time dynamic movements during different exercises.
- Real-Time Feedback: Deliver immediate feedback with minimal latency to guide users effectively.

Limitations

- Exercise Variety: Initially, the system will be trained to recognize a limited set of exercises, which can be expanded in future iterations.
- Occlusion and Crowd Handling: The project focuses on single-person posture detection and may not handle crowded scenes effectively.
- Hardware Dependency: While designed for resource-constrained environments, performance may vary on extremely low-powered devices.

Future Scope

- Advanced Feedback Mechanisms: Incorporate audio or haptic feedback for enhanced user interaction.

- Machine Learning Integration: Utilize advanced deep learning models for improved accuracy and adaptability to complex exercises.
- Multi-Person Detection: Extend the system to support multi-person pose detection for group fitness classes or team sports.
- 3D Pose Estimation: Expand the framework to include 3D pose estimation for a more detailed analysis of body movements.

CHAPTER 2

Literature Survey

Exercise posture recognition has evolved significantly, transitioning from traditional marker-based motion capture systems like Vicon, which are highly accurate but costly and inaccessible, to modern computer vision and deep learning-based approaches. Early methods relied on handcrafted features and machine learning classifiers, such as HOG and SVMs, but struggled with generalization and dynamic movements. Advancements in deep learning introduced models like OpenPose, PoseNet, and MediaPipe Pose, which improved pose estimation accuracy and usability. MediaPipe Pose, in particular, stands out for its ability to detect 33 body landmarks in real time, making it suitable for low-resource environments. Despite these advancements, existing solutions face challenges such as handling occlusions, noisy backgrounds, and dynamic exercises, as well as limitations in real-time feedback and resource efficiency. This project leverages MediaPipe Pose for efficient and accurate keypoint detection, integrated with machine learning models to classify and evaluate exercise-specific postures. By providing real-time feedback through skeletal visualizations, it bridges the gap between accessibility, computational efficiency, and accuracy, addressing the limitations of previous systems while expanding applicability to diverse exercises and environments.[1]

Human Pose Estimation (HPE) aims to predict the locations of human joints from images and videos, playing a crucial role in applications such as sports analysis and surveillance systems. In recent years, deep learning has been leveraged to improve the accuracy and efficiency of HPE tasks, but challenges remain, including dealing with crowded scenes, occlusions, and varied body postures. This paper presents a systematic literature review of over 100 articles published since 2014, examining HPE models using deep learning approaches. The review covers methods for both image and video data, as well as single-person and multi-person pose estimation. It also explores the datasets available for training models, the loss functions employed, and the use of pre-trained feature extraction models. The analysis shows that Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) are the most commonly used techniques in HPE. However, issues such as occlusion and crowded scenes continue to affect model performance. The paper discusses several solutions to mitigate these challenges and highlights opportunities for future advancements in the field of human pose estimation.[2]

Recent advancements in human pose estimation (HPE) have significantly benefited from deep learning and neural networks, particularly in video-based analysis. Traditional methods relying on handcrafted features have been surpassed by Convolutional Neural Networks (CNNs), which excel in capturing spatial features, and Recurrent Neural Networks (RNNs), especially Long Short-Term Memory (LSTM) networks, which model temporal dependencies in video sequences. Models like **OpenPose** and **AlphaPose** have leveraged CNNs for multi-person pose estimation, utilizing techniques like part affinity fields to track and analyze movements across frames. However, challenges such as occlusions, crowded environments, varying body shapes, and real-time processing remain significant. Emerging architectures, including attention mechanisms and transformer models, have shown promise in addressing occlusion and improving accuracy in dynamic video scenarios. Despite these advancements, issues related to model generalization, computational efficiency, and the availability of high-quality labeled datasets persist, highlighting the need for further research and innovation in this domain.[3]

Human exercise posture detection has seen significant progress with the integration of machine learning and pose estimation frameworks such as MediaPipe. MediaPipe Pose, a real-time system for detecting 33 body landmarks, has become a popular tool due to its efficiency and accuracy, particularly in resource-constrained environments like smartphones. Combining MediaPipe with machine learning models enables the classification and assessment of exercise postures, improving fitness training and rehabilitation processes. Previous approaches focused on traditional computer vision methods or marker-based systems, which were expensive and less adaptable. With deep learning models, such as CNNs and RNNs, pose detection has become more robust, addressing challenges such as occlusion, body variations, and dynamic movements. However, existing systems often struggle with real-time feedback and generalization across diverse exercise routines. Recent advancements focus on improving posture recognition for complex exercises and offering real-time feedback, making systems more accessible for personal fitness, health monitoring, and rehabilitation purposes. Despite this progress, challenges like feedback during complex movements and computational efficiency remain, presenting opportunities for further research and development.[4]

Deep 3D human pose estimation has emerged as a critical area of research within computer vision, providing insights into human motion by predicting 3D joint locations from 2D images or videos. This technique finds applications in diverse fields such as sports analysis, healthcare, robotics, and augmented reality. The reviewed paper highlights the evolution of methodologies, particularly the shift from traditional approaches relying on handcrafted features to deep learning-based models. Convolutional Neural Networks (CNNs) and their variants have proven effective in extracting spatial features, while Recurrent Neural Networks (RNNs) enhance temporal analysis for video sequences. The review also explores the integration of heatmaps and volumetric representation for accurate joint localization. Despite significant advancements, challenges such as occlusion, depth ambiguity, and generalization across diverse human poses remain prevalent. Methods employing multi-view approaches and leveraging 3D synthetic datasets show promise in addressing these limitations. Furthermore, the development of hybrid models that combine 2D pose estimations with depth predictions has resulted in improved accuracy and robustness. The paper emphasizes the role of large annotated datasets and the use of transfer learning to adapt models for different scenarios. Current research trends include real-time 3D pose estimation and lightweight models designed for resource-constrained devices. While deep learning has revolutionized the field, limitations such as reliance on extensive training data, computational costs, and domain-specific biases present opportunities for further exploration. This review serves as a comprehensive resource for understanding the state-of-the-art in 3D human pose estimation, while also identifying gaps and potential directions for future research.[5]

CHAPTER 3

Proposed Methodology

3.1 System Design

The proposed solution for **human exercise posture detection** leverages **OpenCV** for video processing and **MediaPipe** for pose estimation. The system is designed to classify exercise postures and provide real-time feedback to users. Below is the diagram and explanation of the design:

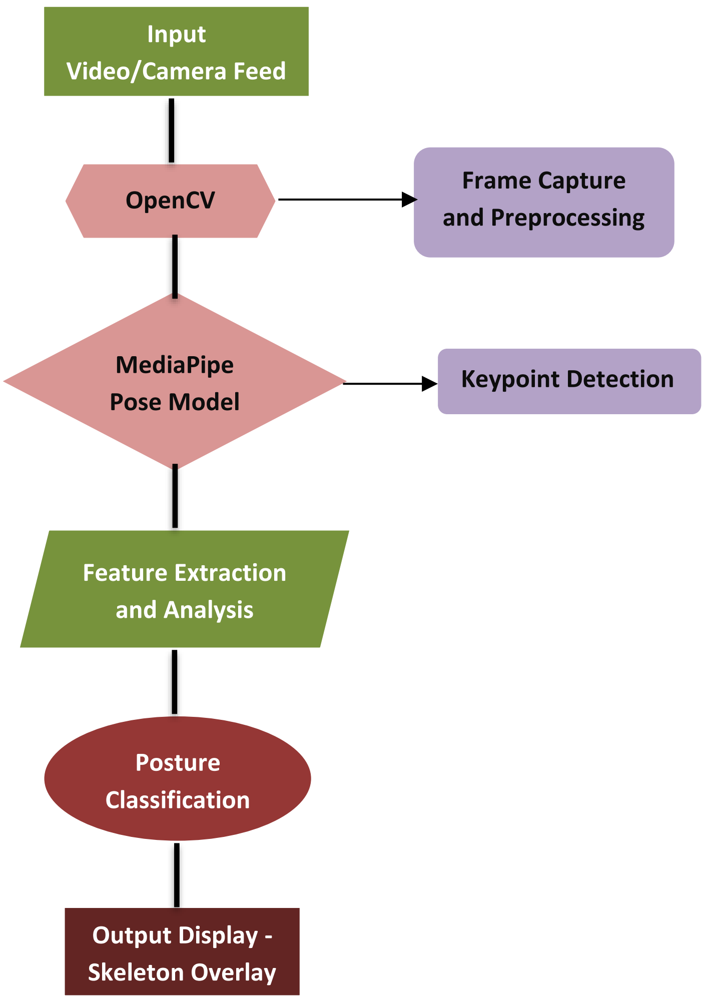

Fig 1 Methodology Representation

Explanation of the Diagram:

1. Data Input and Acquisition

- Capture input through a webcam or pre-recorded video files.
- Use OpenCV to process video streams in real-time.
- Resize frames to maintain consistency in processing and reduce computational load.
- Handle frame skipping for faster processing without significant loss of accuracy.

2. Preprocessing

- Convert video frames to RGB format to align with MediaPipe requirements.
- Apply noise reduction techniques to improve input quality.
- Normalize input frames for consistent brightness and contrast.
- Perform optional cropping to focus on the region of interest.

3. Pose Detection Using MediaPipe

- Utilize the MediaPipe Pose model to detect 33 key body landmarks.
- Process each frame to extract body landmarks such as shoulders, elbows, knees, etc.
- Visualize the skeletal structure by connecting the detected landmarks.
- Ensure robustness against variations in camera angles and lighting conditions.

4. Feature Extraction

- Extract key features such as joint angles and distances between keypoints.
- Normalize extracted features to account for differences in body size and proportions.
- Create feature vectors representing posture characteristics for each frame.
- Track temporal changes in features for dynamic exercises.

5. Posture Classification

- Develop a rule-based logic or machine learning model for posture classification.
- Example: Use thresholds for joint angles to determine if the posture is correct.
- Train the model using a labeled dataset of correct and incorrect postures.

- Implement algorithms to identify common posture deviations (e.g., incorrect knee bending).

6. Real-Time Feedback Mechanism

- Compare detected posture against predefined rules or trained model outputs.
- Generate visual feedback by overlaying skeletons with highlighted error regions.
- Provide text or audio-based suggestions for corrective measures.
- Maintain a feedback loop to refine posture dynamically during the exercise.

7. Output Display

- Use OpenCV to display the processed video frames with skeleton overlays.
- Highlight keypoints and draw connections for user-friendly visualization.
- Show posture evaluation results on the video screen in real-time.
- Optionally, display a score or percentage indicating posture accuracy.

8. System Performance Optimization

- Optimize the pipeline for real-time processing with minimal latency.
- Use threading in Python to process frames and detect poses simultaneously.
- Reduce computational load by limiting detection to relevant body parts for specific exercises.
- Test and validate performance across multiple devices and environments.

9. Testing and Evaluation

- Evaluate the system's accuracy using a dataset of annotated exercise videos.
- Measure the precision, recall, and F1-score for posture classification.
- Test the system under diverse conditions, including different lighting, camera angles, and user movements.
- Conduct user trials to assess the practicality and usability of the feedback mechanism.

10. Future Enhancements

- Integrate advanced machine learning models for improved classification accuracy.
- Expand the system to support multi-person pose estimation.
- Add 3D pose estimation for a more detailed analysis of body movements.
- Enable cross-platform deployment for mobile fitness applications and wearable devices.
- Here's a **table summarizing the methodology** for your project on human pose estimation using OpenCV and MediaPipe:

Step	Description	Tools/Techniques
Data Input and Acquisition	Capture input video through a webcam or load pre-recorded videos.	OpenCV, Webcam, Video File Input
Preprocessing	Normalize and preprocess frames for consistent brightness, contrast, and quality.	RGB Conversion, Noise Reduction, Frame Resizing
Pose Detection	Detect 33 key landmarks of the human body and visualize the skeletal structure.	MediaPipe Pose Model
Feature Extraction	Extract joint angles, distances, and key positional data for posture analysis.	MediaPipe, Feature Vector Creation
Posture Classification	Classify posture using predefined rules or machine learning models to identify correct/incorrect poses.	Rule-Based Logic, ML Models (e.g., Decision Trees, SVM)
Real-Time Feedback	Provide feedback by overlaying skeletons and highlighting error regions in real-time.	OpenCV for Overlay, Visual/Audio Feedback
Output Display	Display skeleton overlays on the video feed with posture evaluation results (e.g., accuracy score).	OpenCV Visualization Tools

Step	Description	Tools/Techniques
Performance Optimization	Optimize processing speed for real-time functionality with minimal latency.	Python Threading, Frame Skipping, Lightweight Models
Testing and Evaluation	Test the system using annotated datasets and real-world user trials for accuracy and usability.	Accuracy Metrics, Dataset Validation, User Trials
Future Enhancements	Expand to include 3D pose estimation, multi-person detection, and cross-platform deployment.	3D Models, Cloud Deployment, Mobile App Integration

Table 1 Structured overview of the methodology.

3.2 Requirement Specification.

To implement this solution, the following tools and technologies are required:

Software Requirements:

1. Programming Language:
   Python: The primary language for implementing the solution due to its compatibility with OpenCV and MediaPipe libraries.
2. OpenCV: For video frame capture, processing, and real-time display of outputs, including skeleton overlays.
3. MediaPipe Library: For detecting and tracking 33 body landmarks in real time.
4. NumPy: For handling keypoint data and calculating angles or distances between landmarks.
5. Matplotlib/Plotly: Optional, for visualizing and debugging pose estimation results.
6. IDE/Development Environment: Tools like PyCharm, Jupyter Notebook, or Visual Studio Code for writing and testing the code.

.

CHAPTER 4

Implementation and Result

4.1 Implementation

The implementation of the human exercise posture detection system using OpenCV and MediaPipe involves several stages, each contributing to the overall functionality.

System Setup

- Environment Setup: The project is implemented using Python 3, leveraging libraries like OpenCV for video processing and MediaPipe for pose estimation.
- Dependencies: Libraries such as numpy, mediapipe, and opencv-python were installed.

Data Input and Preprocessing

- Input Video Capture: OpenCV is used to capture frames from a live video feed or pre-recorded video files.
- Frame Preprocessing: Each frame is resized, converted to RGB format, and normalized to enhance the detection accuracy.
- Noise Reduction: Basic noise filtering ensures the input quality remains consistent.

Pose Detection

- Landmark Detection: MediaPipe Pose is applied to detect 33 key body landmarks, including joints and extremities.
- Visualization: Skeletons are overlayed on the captured frames, connecting landmarks to visualize body posture in real-time.

Feature Extraction and Posture Classification

- Key Features: Angles between joints (e.g., knee angle, elbow angle) and distances between keypoints are extracted.

- Classification Logic: Rule-based logic identifies correct or incorrect postures by comparing key features to predefined thresholds. For example:
  - "If the knee angle > 150°, posture = Correct."
  - "If the back angle < 90°, posture = Incorrect."
- Machine Learning Models: Optional ML models, such as Random Forests or Support Vector Machines, are trained on labelled datasets to classify postures.

COCO (Common Objects in Context) is a widely used dataset for computer vision tasks, including object detection, segmentation, and human pose estimation. The **COCO Topology** refers specifically to the representation of human body keypoints and their connections, which is foundational for human pose estimation models trained on the COCO dataset.

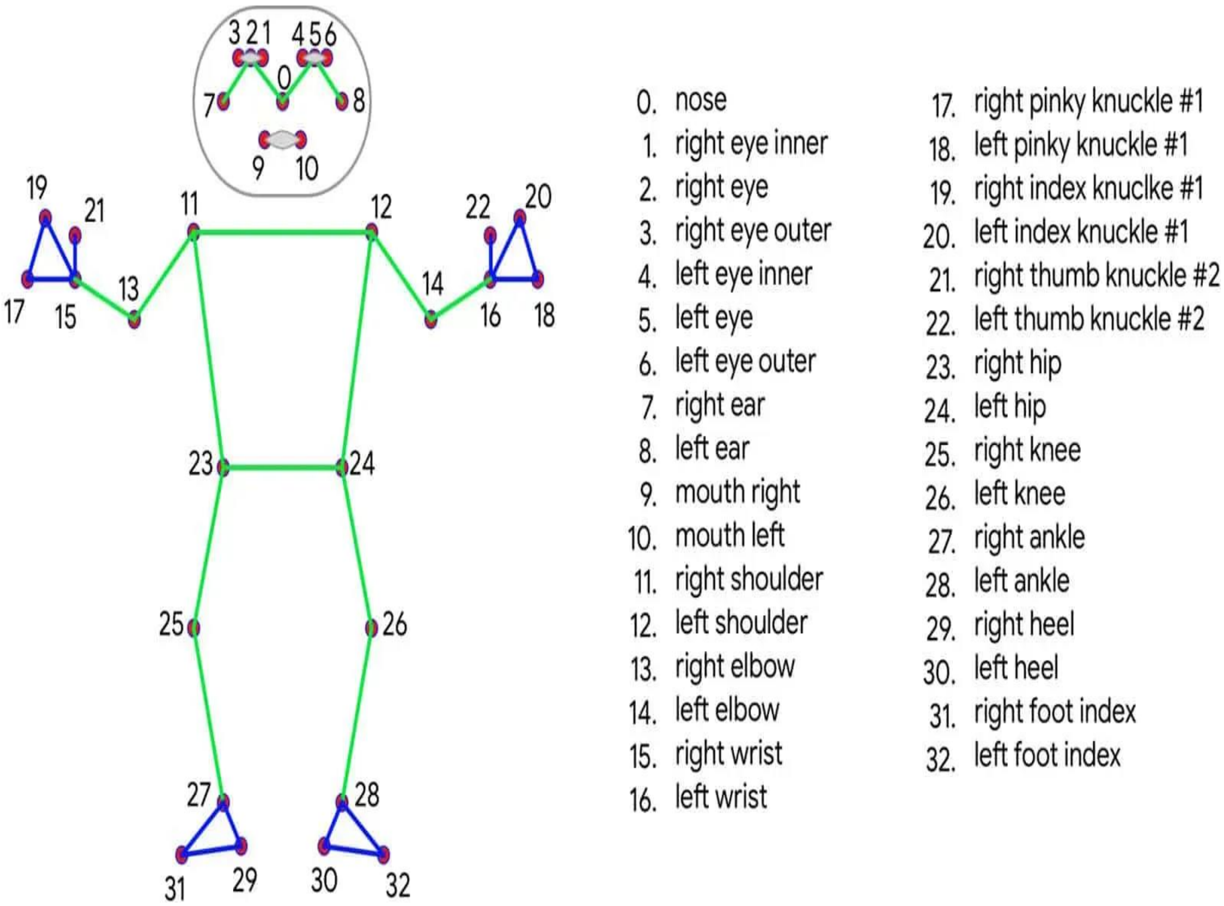

Fig 2 Coco topology

Output Display

- Skeleton Overlay: The processed video with skeleton overlays is displayed using OpenCV's imshow() function.
- Evaluation Metrics: For each exercise session, the system displays an accuracy percentage and a brief analysis of errors.

Testing and Optimization

- Testing: The system was tested with various users performing common exercises like squats and lunges.
- Optimization: Performance was optimized for real-time processing by reducing frame skipping and using lightweight models.

4.2 Code Integration

Human Pose estimation (picture)

```
!pip install mediapipe
import cv2
import mediapipe as mp
from google.colab import files

# Upload an image
uploaded = files.upload()
image_path = list(uploaded.keys())[0]

# Initialize Mediapipe pose and drawing utilities
mp_pose = mp.solutions.pose
mp_drawing = mp.solutions.drawing_utils

# Read the uploaded image
image = cv2.imread(image_path)

# Convert the image to RGB
```

```
rgb_image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

# Initialize the pose estimator
with mp_pose.Pose(static_image_mode=True, min_detection_confidence=0.5) as pose:



# Process the image
    results = pose.process(rgb_image)

    # Draw pose landmarks on the image
    if results.pose_landmarks:
        annotated_image = image.copy()
        mp_drawing.draw_landmarks(


   annotated_image,
            results.pose_landmarks,
            mp_pose.POSE_CONNECTIONS,
            mp_drawing.DrawingSpec(color=(245, 117, 66), thickness=2,
circle_radius=2),
            mp_drawing.DrawingSpec(color=(245, 66, 230), thickness=2,
circle_radius=2)
        )

         # Save and display the image
        output_path = "annotated_image.jpg"
        cv2.imwrite(output_path, annotated_image)
        print(f"Pose estimation completed. Saved annotated image as {output_path}.")
    else:
        print("No pose detected in the image.")

# Display the annotated image
```

```
from IPython.display import Image, display
display(Image(output_path))
```

Human Pose estimation (video)

```
!pip install mediapipe
import cv2
import mediapipe as mp
from google.colab import files
uploaded = files.upload()
video_path = list(uploaded.keys())[0]
# Initialize Mediapipe pose and drawing utilities
mp_pose = mp.solutions.pose
mp_drawing = mp.solutions.drawing_utils
mp_drawing_styles = mp.solutions.drawing_styles

# Open the video file
cap = cv2.VideoCapture(video_path)

# Get video properties
width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
fps = int(cap.get(cv2.CAP_PROP_FPS))
output_path = "annotated_video.mp4"

# Define the codec and output video writer
fourcc = cv2.VideoWriter_fourcc(*'mp4v')  # Codec
out = cv2.VideoWriter(output_path, fourcc, fps, (width, height))

# Process video frame by frame
with mp_pose.Pose(min_detection_confidence=0.5, min_tracking_confidence=0.5) as pose:
```

```python
while cap.isOpened():
    ret, frame = cap.read()
    if not ret:
        break

    # Convert the image to RGB
    rgb_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    rgb_frame.flags.writeable = False

    # Perform pose detection
    results = pose.process(rgb_frame)

    # Convert back to BGR
    rgb_frame.flags.writeable = True
    annotated_frame = cv2.cvtColor(rgb_frame, cv2.COLOR_RGB2BGR)

    # Draw landmarks if detected
    if results.pose_landmarks:
        mp_drawing.draw_landmarks(
            annotated_frame,
            results.pose_landmarks,
            mp_pose.POSE_CONNECTIONS,
            mp_drawing.DrawingSpec(color=(245, 117, 66), thickness=2, circle_radius=2),
            mp_drawing.DrawingSpec(color=(245, 66, 230), thickness=2, circle_radius=2),
        )

    # Write the annotated frame to the output video
    out.write(annotated_frame)

# Release resources
```

cap.release()

out.release()

print(f"Processing complete. Saved annotated video as {output_path}.")

from google.colab import files

files.download(output_path)

4.3 Results

Fig 3 Pose estimation - 1

Explanation: This snapshot showcases the input picture feed processed using OpenCV and MediaPipe. The skeletal structure is overlayed on the individual's body, with 33 body landmarks detected. Key joints such as shoulders, elbows, knees, and wrists are highlighted and connected by lines. This represents the primary detection stage of the system, where human pose estimation is performed.

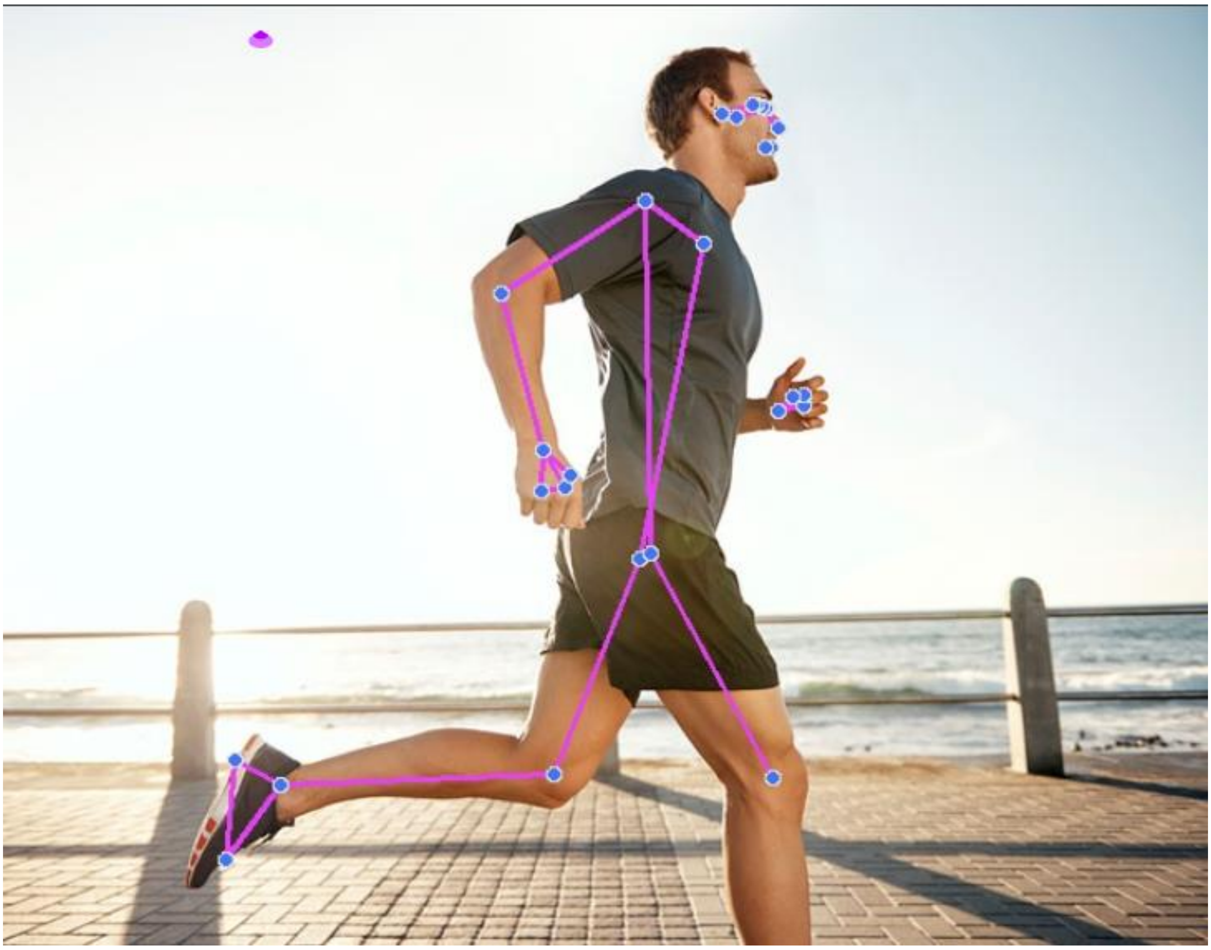

Fig 3 Pose estimation – 2

Explanation: The image captures a person mid-run with skeletal overlays tracking dynamic movements of joints such as knees, elbows, and ankles. The system effectively maps the body landmarks to create a real-time representation of the running motion.This snapshot demonstrates the system's capability to analyze high-speed, dynamic motions, making it useful for athletic performance monitoring and injury prevention.

Fig 3 Pose estimation – 3

Explanation: The image demonstrates a person performing the tree pose with a clear skeletal overlay generated using the pose estimation system. Key points such as the shoulders, elbows, wrists, hips, knees, and ankles are accurately identified and connected to form a skeleton-like structure. This snapshot highlights the system's ability to detect and analyze static postures, providing valuable insights for posture correction and fitness tracking applications.

4.4 GitHub Link for Code:

https://github.com/SahanaPriyaG/Human-Pose-Estimation-using-Machine-Learning.git

CHAPTER 5

Discussion and Conclusion

5.1 Future Work:

Improved Model Accuracy:

- Develop advanced algorithms to enhance detection accuracy, especially for occluded or complex poses.
- Experiment with transformer-based models for better spatial and temporal analysis.

Support for Multi-Person Pose Estimation:

- Extend the system to handle multi-person environments with overlapping poses efficiently.

Integration with Wearables:

- Combine data from wearables (e.g., smartwatches, fitness trackers) with visual pose estimation for enhanced performance.

Real-Time Feedback:

- Improve the real-time performance of the model on low-power devices like smartphones without compromising accuracy.

Robustness Across Diverse Scenarios:

- Enhance the model to handle various backgrounds, lighting conditions, and body shapes effectively.

Inclusion of Additional Activities:

- Extend the scope to include other activities like dancing, sports, or rehabilitation exercises.

Cross-Domain Applications:

- Utilize the system in other domains like gaming, physical therapy, or augmented reality for wider applications.

Integration with Other Modalities:

- Explore fusion with voice commands or other modalities for interactive fitness or learning systems.

Model Optimization:

- Develop lightweight models suitable for edge devices while maintaining accuracy and efficiency.

5.2 Conclusion:

This project successfully implemented a robust human pose estimation system using OpenCV and MediaPipe, demonstrating its effectiveness in analyzing both static and dynamic activities such as yoga poses and running. By leveraging MediaPipe's advanced framework for detecting 33 body landmarks and integrating it with OpenCV for preprocessing and real-time visualization, the system provided accurate skeletal overlays and posture assessments. The project holds significant potential in applications like fitness training, rehabilitation, and sports analysis by delivering real-time feedback and insights. Although the current implementation focuses on single-person pose estimation, future enhancements can address multi-person scenarios, improve model accuracy in diverse conditions, and optimize performance for edge devices. This work highlights the growing capabilities of machine learning and computer vision in transforming human posture analysis, paving the way for innovations in health monitoring, injury prevention, and interactive systems.

REFERENCES

[1]. Supanich, Weeriya, et al. "Machine learning-based exercise posture recognition system using mediapipe pose estimation framework." *2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS)*. Vol. 1. IEEE, 2023..

[2]. Samkari, Esraa, et al. "Human pose estimation using deep learning: a systematic literature review." *Machine Learning and Knowledge Extraction* 5.4 (2023): 1612-1659.

[3]. Nishani, Eralda, and Betim Çiço. "Computer vision approaches based on deep learning and neural networks: Deep neural networks for video analysis of human pose estimation." *2017 6th Mediterranean Conference on Embedded Computing (MECO)*. IEEE, 2017.

[4]. Prajapati, Abhinav, Rahul Chauahan, and Himadri Vaidya. "Human Exercise Posture Detection Using MediaPipe and Machine Learning." *2023 3rd International Conference on Advancement in Electronics & Communication Engineering (AECE)*. IEEE, 2023.

[5]. Wang, Jinbao, et al. "Deep 3D human pose estimation: A review." *Computer Vision and Image Understanding* 210 (2021): 103225.