Project Report

on

BODY POSTURE RECOGNITION AND FITNESS TRAINING

Submitted to

Shri Ramdeobaba College of Engineering & Management, Nagpur

(An Autonomous Institute Affiliated to Rashtrasant Tukdoji Maharaj Nagpur University)

for partial fulfillment of the degree in

Bachelor of Engineering

(Information Technology)

Sixth Semester

by

HIMANSHU HIWARE

VINAY DONGRE

HRUSHIKESH DAGWAR

Under the Guidance of

Ms. Chetana Thaokar

Department of Information Technology

Shri Ramdeobaba College of Engineering & Management, Nagpur-13

2022-23

CERTIFICATE

This is to certify that the Project Report on

"BODY POSTURE RECOGNITION AND FITNESS TRAINING"

is a bonafide work and it is submitted to

Shri Ramdeobaba College of Engineering & Management, Nagpur

(An Autonomous Institute Affiliated to Rashtrasant Tukdoji Maharaj Nagpur University)

by

HIMANSHU HIWARE

VINAY DONGRE

HRUSHIKESH DAGWAR

For partial fulfillment of the degree in

Bachelor of Engineering in Information Technology,

Sixth Semester

during the academic year 2022-23

under the guidance of

Ms. Chetana Thaokar

Assistant Professor, Dept. of IT, RCOEM, Nagpur

Dr. P.D. Adane
Head, Department of Information Technology
RCOEM, Nagpur

Dr. R. S. Pande
Principal
RCOEM, Nagpur

RCOEM
Shri Ramdeobaba College of Engineering and Management, Nagpur

Department of Information Technology

Shri Ramdeobaba College of Engineering & Management, Nagpur-13

2022-23

ACKNOWLEDGEMENTS

We would like to express our sincere gratitude and deepest appreciation to several individuals who have played a pivotal role in the successful completion of our project. First and foremost, we extend our heartfelt thanks to our project mentor, **Ms. Chetana Thaokar**, for her exceptional guidance, constant support, and motivation throughout the project.

We would also like to acknowledge the invaluable support and encouragement of **Dr. P. D. Adane** HOD, IT, who not only provided us with the necessary resources but also stood by us during the entire project duration.

Furthermore, we would like to extend our thanks to the college for granting us the opportunity to collaborate with them and for providing the essential facilities for our project. We are also grateful to all the staff members of the department who have offered their assistance and encouragement, directly or indirectly, throughout our endeavor. Their involvement and interest have been instrumental in our progress.

Once again, we express our heartfelt appreciation to all those who have contributed to our project.

Name of the Projectees

HIMANSHU HIWARE
VINAY DONGRE
HRUSHIKESH DAGWAR

CONTENTS

CHAPTER 3
PROPOSED METHODOLOGY

CHAPTER 4
IMPLEMENTATION AND RESULTS

ABSTRACT

This project combines advanced technology and machine learning to focus on body posture detection and fitness training. By employing computer vision and machine learning algorithms, the system can accurately detect and analyze the user's body posture in real-time. This real-time analysis enables the system to provide personalized fitness recommendations and exercises tailored to each individual's needs, ultimately aiming to improve their overall health and well-being.

With interactive features, progress tracking, and a user-friendly interface, the system ensures accessibility and convenience for users. The interactive features engage users by providing visual feedback, exercise demonstrations, and personalized guidance. Progress tracking allows users to monitor their improvements over time, providing motivation and a sense of accomplishment.

By addressing the importance of proper posture and regular exercise, this project empowers individuals to take charge of their fitness journey and make positive lifestyle changes. Through the accurate detection of body posture and personalized fitness recommendations, users can learn proper form and techniques, reducing the risk of injuries and maximizing the effectiveness of their workouts.

The user-friendly interface and accessibility of the system make it suitable for people of all fitness levels and backgrounds. Whether users are beginners or experienced fitness enthusiasts, they can benefit from the system's personalized recommendations and exercises, tailored to their specific posture analysis.

Overall, this project has the potential to revolutionize the way people approach fitness training by leveraging advanced technology and machine learning algorithms, ultimately empowering individuals to achieve their fitness goals and improve their overall health and well-being.

List of Figures

CHAPTER-1

INTRODUCTION

Chapter -1

INTRODUCTION

Our project focuses on utilizing advanced technology, computer vision, and machine learning to revolutionize fitness training. By accurately detecting and analyzing body posture in real-time, we provide personalized fitness recommendations and exercises based on individual needs. With interactive features, progress tracking, and a user-friendly interface, our system aims to empower users to improve their overall health and well-being. By emphasizing the importance of proper posture and regular exercise, we strive to make positive lifestyle changes accessible and convenient for individuals, ultimately helping them take charge of their fitness journey.

1.1 Background

Almost everyone has experienced backpain due to incorrect posture. Incorrect posture seems to grow with time and becomes a chronic source of pain and bodily problems (Refer Figure 1.1).People are aware of the impacts of bad posture, yet do not make conscious efforts to change these bad habits.

Bad posture leads to not having confidence, which further leads to low self-esteem. This decreases the productivity of the employee, which is a great loss for the organization. We aim to develop a unique application that will support people to improve their posture by using the cell phone’s camera to monitor the user while they are seated and notify them whenever they get into a bad posture.

Although there are various other posture correction applications on app stores, almost all of them involve some expensive tracker placed on the body and do not seem to use the vast capabilities of machine learning and computer vision.

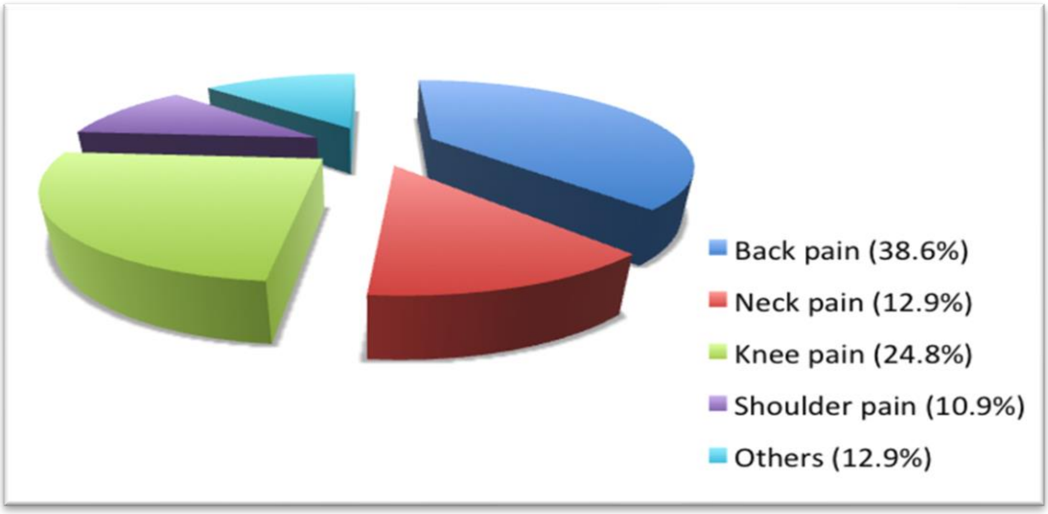

Figure 1.1 Body pain pie chart

1.2 Project Objective

The project aims to enhance the exercise experience and improve predictive accuracy by implementing a software-based view component. By leveraging advanced algorithms and computer vision techniques, the system provides users with a comprehensive assessment of their exercise form and technique.

The software captures video footage of the user performing exercises from different angles. Using computer vision algorithms, it analyzes the recorded video to evaluate the user's posture, body alignment, and movement patterns.

The mobile application utilizes the front and back cameras of smartphones or tablets to capture the user's exercise movements. The software analyzes the recorded video footage, allowing users to benefit from a multi-angle view of their exercise performance.

By implementing this software-based view feature, provides users with valuable feedback on their exercise form. The system detects deviations from optimal posture and technique, helping users make necessary adjustments to maximize the effectiveness and safety of their workouts.

Overall, the software-based view component enhances the project by providing users with a comprehensive assessment of their exercise performance. By leveraging computer vision algorithms, the software ensures accurate form analysis, promotes proper exercise technique, and helps users achieve their fitness goals effectively and safely as shown in Figure 1.2.

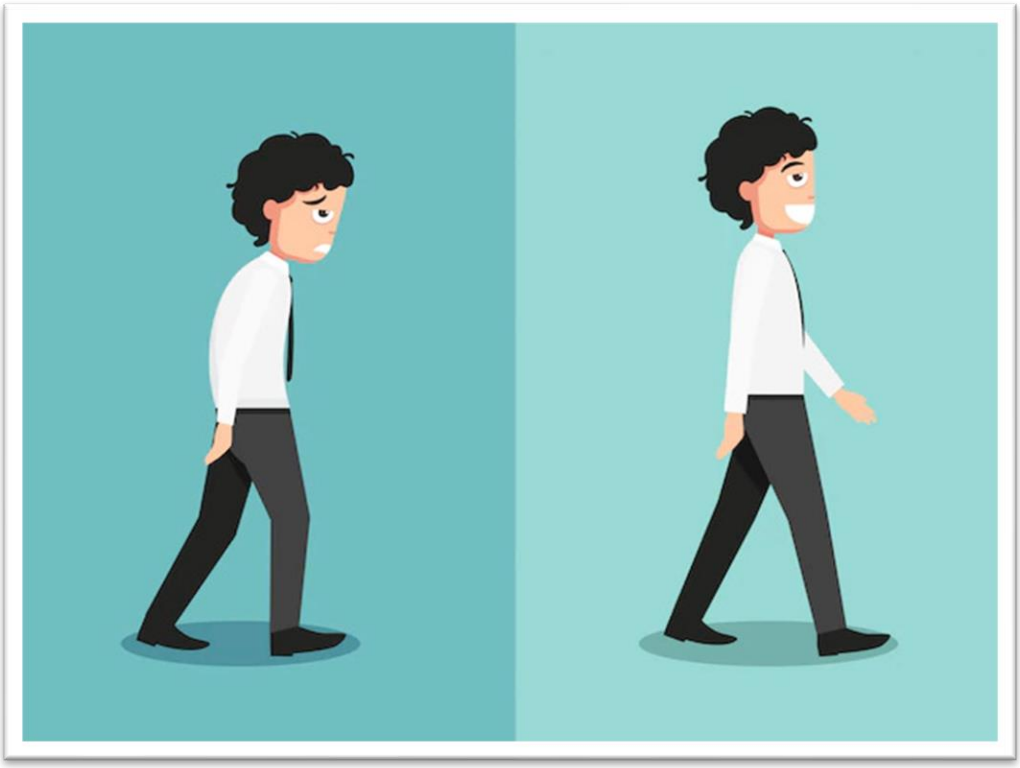

Figure 1.2 Incorrect and correct posture

1.3 Posture Detection

Posture detection, in the context of computer vision, refers to the process of automatically identifying and analyzing the body posture or positioning of individuals in images or videos. It involves using computer algorithms and techniques to extract relevant information about the spatial arrangement and alignment of different body parts.

Computer vision-based posture detection algorithms typically analyze visual data, such as images or video frames, to identify key landmarks or key points on the human body. These key points may include joints like shoulders, elbows, wrists, hips, knees, and ankles, as well as other significant body parts.

By detecting and tracking these key points over time, computer vision algorithms can analyze the relative positions, angles, and movements of body parts to infer the user's posture. For example, the algorithm may assess whether a person is sitting, standing, bending, or performing a specific exercise or movement.

Various techniques are employed in posture detection algorithms, including image processing, object recognition, feature extraction, and machine learning. These algorithms may use predefined models or be trained on large datasets to recognize specific postures or identify deviations from standard or optimal postures.

Posture detection has applications in a wide range of domains, including healthcare, fitness tracking, ergonomic analysis, sports training, and human-computer interaction. It can be used to monitor and provide feedback on proper body alignment during exercises, assess ergonomic risks in workplaces, detect abnormal or unhealthy postures for posture correction, and track movements for gesture-based interaction with devices or virtual environments.

Overall, posture detection in computer vision enables automated analysis of human body postures, providing valuable insights and applications for improving health, performance, and interaction in various fields.

1.4 Value Proposition

Our smartphone-based posture improvement and recognition system offers a unique value proposition by providing users with real-time feedback and guidance to improve their posture. Leveraging the capabilities of smartphones, our system utilizes advanced computer vision and machine learning algorithms to analyze the user's posture and provide personalized recommendations for correction.

Key Features:

Real-time Feedback: Our system uses the smartphone's camera to continuously monitor the user's posture in real-time. It analyzes the alignment and positioning of key body parts, such as the head, shoulders, spine, and hips, and provides instant feedback on any deviations from the ideal posture.

Personalized Recommendations: Based on the analysis of the user's posture, our system generates personalized recommendations to help improve posture. These recommendations may include corrective exercises, stretching routines, ergonomic adjustments, or posture reminders tailored to the user's specific needs and goals.

Progress Tracking: The system tracks the user's posture improvement over time, allowing them to monitor their progress and stay motivated. It provides visualizations and statistics to show the changes in posture alignment, helping users understand the impact of their efforts.

Customizability and Accessibility: Our system is designed to be highly customizable and accessible. Users can set their posture goals, adjust the sensitivity of the detection system, and choose the level of feedback they prefer. The system can be easily accessed through a smartphone app, making it convenient for users to improve their posture anywhere, anytime.

Educational Resources: In addition to feedback and recommendations, our system provides educational resources such as articles, videos, and tutorials on posture improvement. This empowers users with knowledge and awareness about the importance of good posture and the potential health benefits it offers.

Benefits:

Improved Posture: By providing real-time feedback and personalized recommendations, our system helps users develop better posture habits, reducing the risk of musculoskeletal issues, back pain, and related health problems.

Convenience and Accessibility: Our smartphone-based system allows users to work on their posture improvement at their convenience, without the need for additional devices or equipment. It can be easily integrated into their daily routines and lifestyles.

Motivation and Engagement: The progress tracking and visual feedback features of our system provide users with a sense of achievement and motivation, encouraging them to maintain consistent efforts towards improving their posture.

Empowering Users: Our system educates and empowers users by providing them with knowledge and resources to understand the importance of good posture and take proactive steps towards maintaining a healthy posture.

Overall, our feedback generation posture improvement and recognition system for smartphones offers a user-friendly and effective solution to enhance posture awareness, correct deviations, and promote overall well-being.

1.5 Tech Stack

1.5.1 Media Pipe

Media Pipe is a highly suitable choice for integrating real-time posture detection and analysis into smartphone applications. It offers a comprehensive range of pre-built models and components specifically designed for computer vision tasks, including body pose estimation. By leveraging Media Pipe’s accurate pose estimation algorithms, developers can detect and analyze users' postures in real-time.

The framework provides developers with the flexibility to customize the pipeline according to their specific requirements and seamlessly integrate it into smartphone applications. It offers robust tools and APIs for handling image and video data, performing pose estimation, and extracting key body landmarks. These features enable tracking of users' body movements, identification of posture deviations, and real-time provision of feedback or recommendations.

Media Pipe's adaptability makes it suitable for various smartphone platforms, including Android and iOS. Developers can leverage its pre-built models and components to expedite the development process and achieve accurate posture detection without the need for extensive algorithm development.

Moreover, Media Pipe's open-source nature allows for community contributions and continuous improvement, ensuring access to the latest advancements in computer vision and pose estimation techniques. This facilitates staying up-to-date with the latest developments and incorporating state-of-the-art features into posture detection systems.

In summary, Media Pipe offers a powerful and customizable framework for integrating real-time posture detection(Refer Figure 1.3) and analysis into smartphone applications. Its pre-built models, flexibility, and accuracy make it an ideal choice for developers seeking to create effective and robust posture improvement solutions.

Figure 1.3 Media Pipe

1.5.2 OpenCV Python

OpenCV is a versatile computer vision library that offers a wide range of functions and algorithms for image and video processing. It can be effectively utilized in smartphone applications for posture detection and analysis in conjunction with frameworks like Media Pipe or TensorFlow.

OpenCV provides features such as image preprocessing, feature extraction, object detection, and pose estimation, which are crucial for accurate posture detection. With OpenCV, developers can enhance image quality, extract relevant features from images, detect specific body parts or objects, and estimate body pose and alignment.

By combining OpenCV with frameworks like Media Pipe or TensorFlow, developers can leverage OpenCV's comprehensive image processing capabilities alongside the advanced machine learning and computer vision features of the other frameworks. OpenCV can handle preprocessing tasks, feature extraction, and facilitate the integration of posture detection algorithms provided by Media Pipe or TensorFlow.

Overall, OpenCV serves as a powerful tool for posture detection in smartphone applications, offering extensive support for various platforms and programming languages, and enabling developers to enhance the accuracy and effectiveness of their posture analysis solutions.

1.5.3 Android ML Kit

Android ML Kit is a mobile SDK (Software Development Kit) offered by Google, empowering developers to incorporate machine learning capabilities into Android applications with ease. It provides ready-to-use machine learning models and APIs, eliminating the need for extensive knowledge of machine learning algorithms.

Key features and benefits of Android ML Kit include pre-built machine learning models for various tasks like image labeling, text recognition, and face detection. It supports both on-device and cloud-based processing, offering real-time and resource-intensive capabilities. Integration with Android's camera and vision APIs enables seamless incorporation of machine learning features into camera-based applications.

Android ML Kit simplifies integration and usage through its intuitive API, reducing development time and complexity. It is seamlessly integrated with Firebase, allowing developers to combine machine learning with other Firebase services like authentication and cloud storage.

In summary, Android ML Kit empowers Android developers to effortlessly integrate machine learning functionalities as shown in Figure 1.4, into their applications, leveraging pre-built models, simplified integration, and seamless Firebase integration.

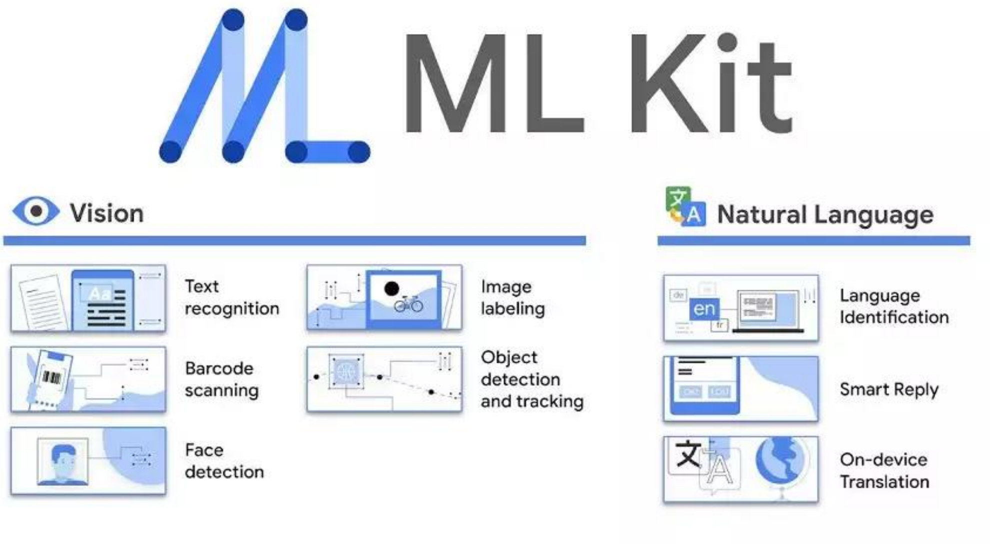

Figure 1.4 ML Kit

1.5.4 Flutter

Flutter is an open-source UI software development kit (SDK) created by Google. It enables developers to build natively compiled applications for multiple platforms, including mobile, web, and desktop, using a single codebase. Flutter uses the Dart programming language, which is also developed by Google.

Key features and advantages of Flutter:

Cross-Platform Development: Flutter allows developers to write code once and deploy it on multiple platforms. This saves development time and effort as there is no need to create separate codebases for each platform. With Flutter, you can build apps for Android, iOS, web, and desktop from a single codebase, ensuring consistent user experiences across platforms.

Fast and High-Performing Apps: Flutter utilizes a compiled and optimized rendering engine called Skia, which provides fast and smooth performance. Flutter apps are compiled to native ARM code, enabling them to run directly on the device's hardware. This results in highly performant and responsive applications.

Hot Reload: Flutter's hot reload feature allows developers to see the changes they make to the code almost instantly. These speeds up the development process and enables developers to experiment and iterate quickly, making it easier to build and refine UI designs.

Rich and Customizable UI: Flutter offers a rich set of pre-designed UI widgets that follow the Material Design and Cupertino (iOS) style guidelines. These widgets can be easily customized and combined to create visually appealing and responsive user interfaces. Flutter's flexible architecture allows for pixel-perfect UI designs and smooth animations.

Access to Native Features and APIs: Flutter provides a rich set of plugins and packages that allow developers to access native device features and APIs. This means you can seamlessly integrate with device-specific functionalities, such as camera, geolocation, sensors, and more, without sacrificing performance or user experience.

Strong Community and Ecosystem: Flutter has a vibrant and growing community of developers, which means there are numerous resources, tutorials, and packages available to assist in the development process. The Flutter ecosystem offers a wide range of libraries, tools, and extensions that extend the capabilities of Flutter and enable developers to build complex and feature-rich applications.

Integration with Firebase and Other Services: Flutter seamlessly integrates with Firebase, Google's backend platform, providing developers with ready-to-use backend services, including authentication, database, cloud storage, and more. Flutter also supports integration with other popular third-party services and APIs, allowing developers to leverage a wide range of functionalities and services in their apps.

1.5.5 Firebase Firestore

Firebase Firestore is a flexible and scalable NoSQL cloud database provided by Google as part of the Firebase platform. It allows developers to store, sync, and query data for their applications easily. Firestore is designed to seamlessly integrate with Firebase's other services, providing a comprehensive backend solution for mobile, web, and server development.

Key features and benefits of Firebase Firestore:

Document-Oriented NoSQL Database: Firestore is a document-oriented NoSQL database, which means data is stored in flexible, JSON-like documents rather than traditional tables. This makes it easy to organize and structure data, accommodating a wide range of data models and allowing for dynamic and evolving schemas.

Scalability and Performance: Firestore is designed to scale seamlessly as your application grows. It automatically handles sharing and distribution of data across multiple servers, ensuring high-performance data access regardless of the database size or number of concurrent users. Firestore also provides offline support, allowing applications to continue functioning even when there is no network connectivity.

Querying and Indexing: Firestore offers powerful querying capabilities, allowing you to retrieve data based on specific criteria. You can perform complex queries, filter data, and sort results. Firestore also supports composite indexes, making it efficient to query large datasets.

Security and Authentication: Firestore integrates with Firebase Authentication, providing robust security features to protect your data. You can easily configure access rules and permissions at the document or collection level, ensuring that only authorized users can read or modify data. Firestore also supports server-side security rules, giving you fine-grained control over data access.

Serverless and Automatic Scaling: Firestore is a serverless database, meaning you don't need to manage or provision servers. It automatically scales to handle traffic spikes, ensuring that your application remains performant and responsive. This allows developers to focus on building the application's features rather than worrying about infrastructure management.

1.5.6 Real Time Database

Firebase Realtime Database is a cloud-hosted NoSQL database by Google for real-time data synchronization across multiple clients. It offers offline support, automatic scaling, and utilizes a JSON data model.

Firebase Realtime Database offers key features and benefits:

Real-time Data Synchronization: Enables real-time updates across connected clients, ensuring all users have the latest data.

Offline Support: Allows apps to function offline by caching data locally and synchronizing changes once the network is available.

NoSQL JSON Data Model: Utilizes a flexible data model based on JSON, enabling easy storage and retrieval of structured data.

Automatic Scaling: Firebase automatically scales the database backend to handle increasing user demand, ensuring app performance.

Serverless Architecture: Eliminates the need to manage database servers, allowing developers to focus on app development rather than infrastructure management.

CHAPTER-2

LITERATURE REVIEW

Chapter-2

LITERATURE REVIEW

Through the literature review, we aim to examine studies, articles, and publications that discuss the use of advanced technology, computer vision, and machine learning algorithms in body posture detection and fitness training. This review helps us understand the advancements made in this field, the methodologies employed, and the effectiveness of different approaches.

2.1 Paper-1: Body Posture Detection Using Computer Vision

Introduction:

Computer Vision is an interdisciplinary field that deals with how computers can be made to gain high-level understanding from digital images and videos. Therefore, Human Body posture detection using computer vision deals with the analysis and high-level understanding of the body posture, detection of the presence of human body in a surrounding and the detailed analysis of the its movements with respect to the surrounding and the various human movements from digital images and videos. A wide spectrum of applications demands human motion recognition. The applications are spread over domains like sports, medical, surveillance, content-based video storage and retrieval, man-machine interfaces, video conferencing, art and entertainment and robotics.

Methodology:

The Technique used by our paper is Human Body Posture Detection in OpenCV Using Open Pose Mobile Net Technology. Mobile Nets are small, low-latency, low powered models which are used to meet the resource constraints of a variety of use cases. It comes under the family of mobile-first vision computer models designed to effectively maximize accuracy while being mindful of the resources for an on device or embedded application. It is a class of efficient modules for mobile and embedded vision applications. They are based on a streamlined architecture that uses depth-wise separable convolutions to build light weight Deep-Neural Networks (Refer Figure 2.1). It can be run on mobile devices with help of Tensor Flow Mobile, it enables on-device machine learning inference with low latency. With help of Android Neural Networks API. It focuses on optimizing for latency but also yields small networks. In the image below we detected the position and orientation of an object by detecting the key point locations that describe the object. We focused on human body posture estimation where it is required to detect and localize the major parts or joints of the body (e.g., Shoulders, ankle, knee, wrist etc.).

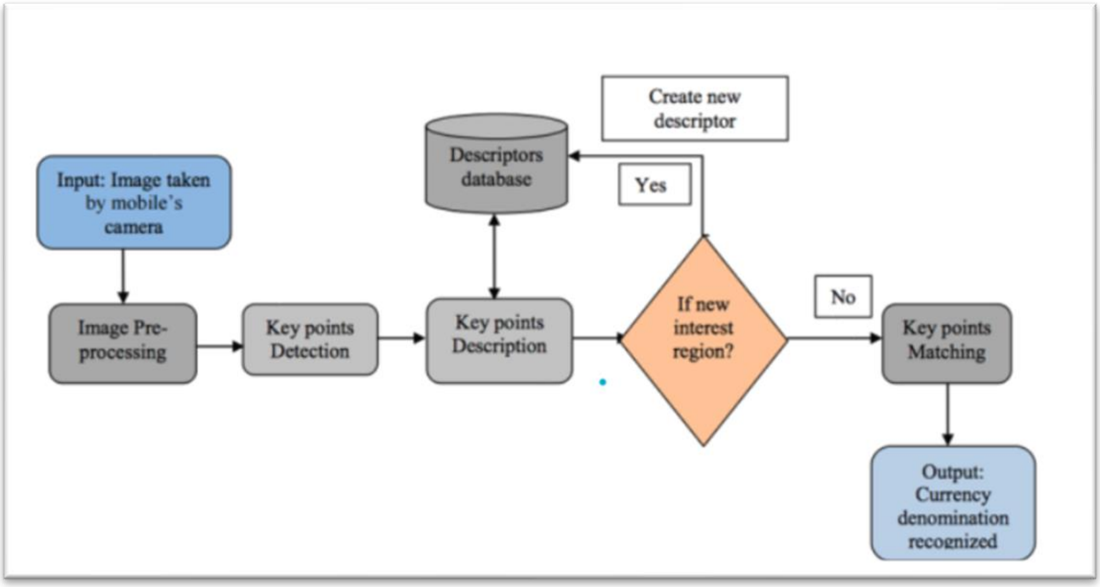

Figure 2.1 Flow Graph

Key Contribution:

Human posture detection using OpenCV is cheaper compared to other methods like using pyroelectric infrared sensors. Human posture detection using OpenCV does not require any external attachments on the body or external materials for its working. It only requires a computer system with required software's, webcam and the presence of the object u see consideration. It is more appropriate when it comes to daily practical application as compared to other methods. Requires minimum number of materials as compared to other methods. It is more effective in practical applications as compared.

Finding and discussion

Conclusion:

This paper provides an overview of human body posture recognition using computer vision, focusing on the detection of posture through OpenCV and Open Pose Mobile Net Technology. The approach is deemed reliable and robust, with potential applications in surveillance, entertainment, and healthcare. However, challenges remain to be addressed, particularly in achieving real-time performance and refining the model for further research and advancements in the field.

2.2 Paper-2: <u>360 Degree Posture Detection in Mobile and Desktop version of Aasan</u>

Introduction:

Aasan is a unique telehealth service that sets itself apart by eliminating excuses for users to neglect posture analysis and correction. Unlike other services, Aasan offers a desktop version for those who prefer not to use a mobile application. Additionally, a lighter model was developed to address concerns about phone overheating and computational power. Aasan aims to be the next generation application that helps people worldwide improve their posture, leading to increased productivity and a better quality of life. With a comprehensive 360-degree solution and access to professional advice, Aasan has the potential to bring about significant positive changes in society.

Methodology:

The plan for Aasan involves leveraging deep learning and machine learning techniques, specifically Google's ML Vision Kit, to address various back problems. By incorporating machine learning models for posture recognition, Aasan aims to offer comprehensive treatments, notifications, and analyses to users. These functionalities can be utilized in several ways, including consultation with professionals and gaining a deeper understanding of posture maintenance as shown in Figure 2.2.

The implementation of 360-degree posture assessing algorithms, through an Android and web application, is a significant aspect of this project. This cross-platform approach to posture detection and analysis is expected to greatly assist individuals and have a notable societal impact.

In terms of the literature review, the team has collected relevant data and developed analysis tools to conduct a comprehensive assessment. The objective is to identify the interrelationships between different back problems and their potential to lead to more serious issues if left unaddressed. Additionally, the team has investigated the associated side effects for each condition. These efforts contribute to improved posture analysis and the development of suggestive corrective measures that can enhance the algorithm's predictive capabilities.

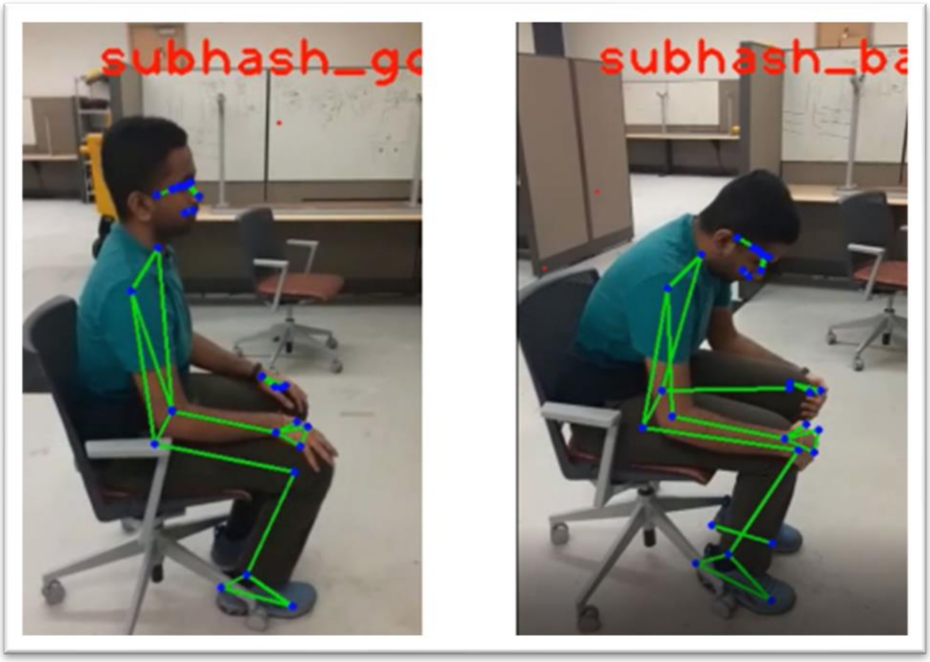

Figure 2.2 Output

Key Contribution:

The key contributions of the project "360 Degree Posture Detection in Aasan" include the development of a mobile and desktop application that utilizes machine learning and computer vision for real-time, 360-degree posture monitoring. The application provides notifications, feedback, and recommendations to help users correct their posture effectively. It integrates Firebase and ML Kit Vision for enhanced functionality and includes a lighter model for the desktop version, optimizing performance and usability.

Conclusion:

The project was a success, providing a comprehensive and user-friendly solution for posture analysis and correction. The combination of Android Studio, Stream lit, and Firebase empowered the team to create a functional, secure, and efficient application. The implementation of various machine learning models and the integration of multiple video input sources contributed to the versatility and accuracy of the system. Overall, Aasan has the potential to make a positive impact on individuals' posture and well-being.

2.3 Paper-3: Yoga Pose Estimation and Feedback Generation Using Deep Learning

Introduction:

The introduction of the research paper emphasizes the significance of practicing yoga poses accurately and highlights the potential risks associated with improper posture. It acknowledges the importance of having an instructor present during yoga sessions to ensure correct form and to prevent injuries. However, it also acknowledges that in today's fast-paced lifestyle, it is not always feasible to have a dedicated instructor or join regular yoga classes.

To address this challenge, the research paper proposes an AI-based system that can serve as a virtual instructor, helping users identify and improve their yoga poses. By leveraging advanced computer vision techniques and deep learning algorithms, the system aims to accurately detect and classify yoga poses, providing users with real-time feedback and suggestions to enhance their practice.

Methodology:

The methodology for conducting a literature review involves a systematic approach to gather and analyze relevant research. It begins by defining the objectives and scope of the review. Next, appropriate databases and search keywords are identified to conduct a comprehensive search. The initial search is followed by a refining process to narrow down the results. Screening and selection criteria are applied to choose relevant sources for inclusion. Key information from selected articles is then extracted and analyzed to identify common themes and patterns. The review is structured with an introduction, main body, and conclusion, presenting a synthesis of the findings. Careful editing and revision ensure clarity and coherence. Lastly, the literature review should be periodically updated to incorporate new research and maintain relevance. This methodology ensures a rigorous and organized approach to reviewing the existing body of knowledge on a particular topic. Refer Figure 2.3 and Figure 2.4.

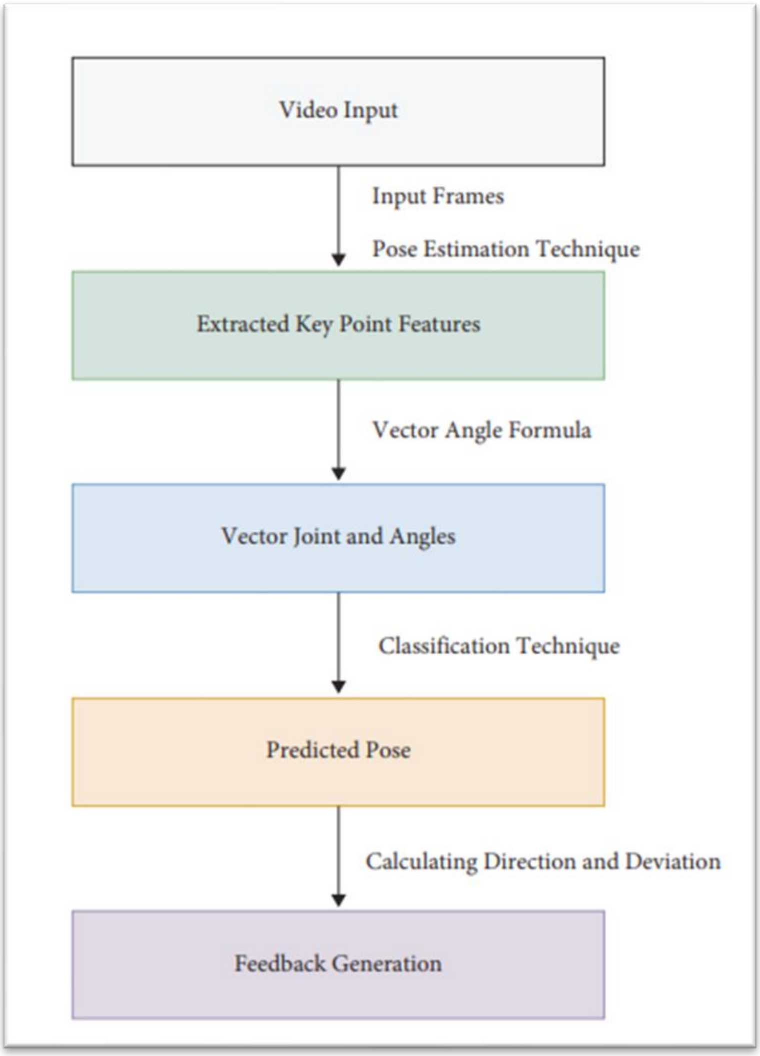

Figure 2.3 Flow Graph

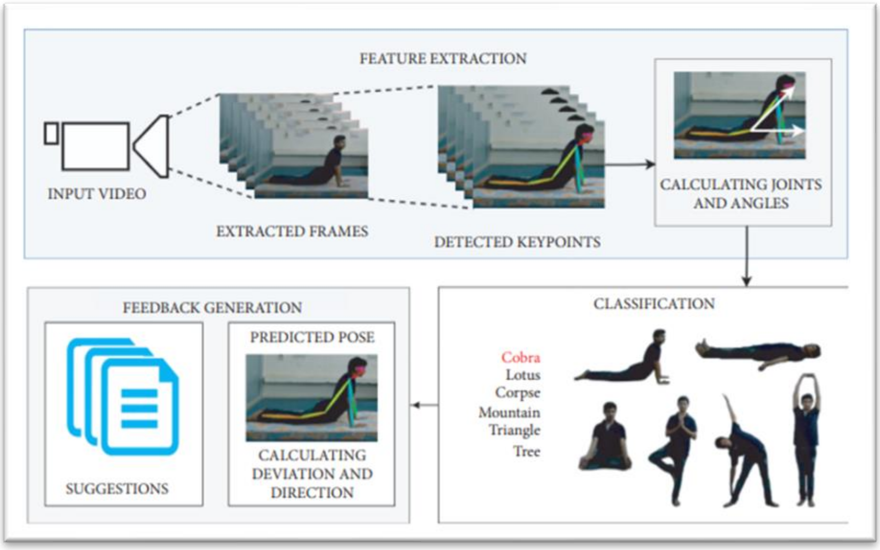

Figure 2.4 Implementation

Key Contribution:

The key contributions of the research lie in the development of a deep learning-based system for yoga pose estimation and improvement. The primary contribution is the introduction of a methodology that utilizes neural networks, particularly the multilayer perceptron (MLP), to accurately classify yoga poses and provide personalized feedback. This approach significantly enhances the accuracy of pose detection compared to existing methods.

Another significant contribution is the emphasis on feature extraction, specifically the calculation of angles between body parts. By considering these angles, the system can detect even slight rotations in key points, leading to improved accuracy in identifying incorrect postures. This feature extraction technique enhances the overall effectiveness of the system.

Finding and discussion:

The report presents key findings on the deep learning-based system for detecting and improving yoga postures. The study showcases the effectiveness of the multilayer perceptron (MLP) model in accurately classifying yoga poses and providing personalized feedback. Feature extraction using angles enhances the system's ability to detect subtle posture variations. The system offers convenience for self-yoga learning, allowing users to upload videos for personalized feedback without the need for an instructor. The findings highlight the system's potential in enhancing posture and preventing long-term issues associated with incorrect poses.

Conclusion:

This research introduces deep learning-based methods for detecting and giving feedback on incorrect yoga postures with high accuracy. The proposed approach shows promising results while maintaining computational efficiency. Future work includes expanding the dataset, exploring real-time predictions on mobile devices, and applying the system in crowded environments. The methodology of extracting angles as features has potential for other applications. Overall, the system provides a practical solution for individuals to enhance their yoga practice and avoid health issues associated with incorrect postures.

CHAPTER-3

PROPOSED METHODOLOGY

Chapter-3

PROPOSED METHODOLOGY

The methodology used in this project on body posture detection and fitness training involves a combination of computer vision techniques and machine learning algorithms. The following is an overview of the methodology employed as shown in Figure 3.1.

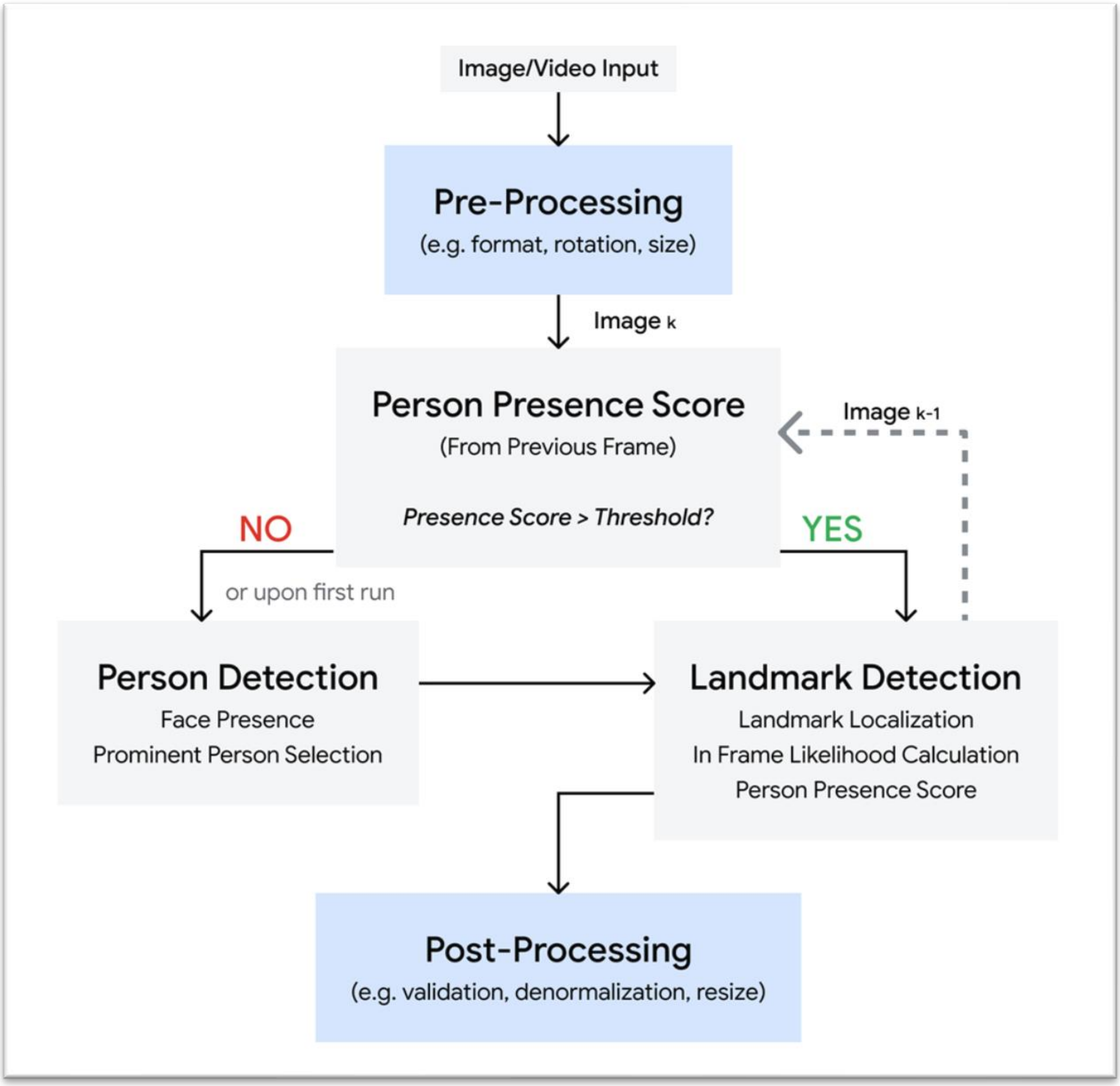

Figure 3.1 Methodology Flow chart

3.1 Data Collection

The methodology for data collection is a crucial step in developing an accurate and robust posture detection model. The following details the methodology for collecting the necessary data:

Define Data Requirements: Determine the specific data requirements for training and evaluating the posture detection model. This includes identifying the types of postures to be detected, the number of different postures, and any specific criteria for selecting the samples.

Source Selection: Identify the sources from which the data will be collected. This can include publicly available datasets, proprietary datasets, or collecting data through user participation or surveys. Ensure that the selected sources align with the defined data requirements.

Sample Size and Diversity: Determine the appropriate sample size based on the complexity and diversity of the postures. Aim for a sufficient number of samples to represent a wide range of postures and variations.
Consider factors such as age, gender, body types, and any other relevant characteristics to ensure diversity in the dataset.

Data Collection Process: Develop a systematic process for collecting the data. This may involve capturing images or videos of individuals performing different postures using cameras or mobile devices. Ensure that the data collection process is consistent and standardized to minimize biases and variations.

Annotation and Labeling: Annotate the collected data by labeling the postures correctly. This involves manually marking the key points or regions of interest in the images or videos that correspond to different postures. Use appropriate tools or software to facilitate the annotation process and ensure accuracy.

3.2 Model Development

The methodology for model development is a crucial step in creating an accurate and reliable posture detection system. The following outlines the methodology for developing the posture detection model:

Data Preparation: Preprocess the collected data to make it suitable for training the posture detection model. This may involve resizing the images or videos, normalizing pixel values, and organizing the data into appropriate formats for model training.

Model Selection: Choose the appropriate machine learning or deep learning model for posture detection. Consider models such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs) that have shown effectiveness in computer vision tasks. Select a model architecture that can capture the spatial relationships and patterns relevant to posture detection.

Training and Validation: Split the preprocessed data into training and validation sets. Use the training set to train the posture detection model, adjusting the model's parameters and weights through an optimization algorithm. Validate the model's performance on the validation set to ensure it generalizes well and does not overfit to the training data.

Model Evaluation: Evaluate the trained model's performance using appropriate evaluation metrics such as accuracy, precision, recall, F1 score, or mean average precision (MAP). Compare the model's performance against baseline or industry-standard models to assess its effectiveness in posture detection.

Fine-tuning and Optimization: Fine-tune the model to improve its performance. This may involve adjusting hyperparameters, exploring different network architectures or regularization techniques, or employing transfer learning by leveraging pre-trained models on related tasks. Continuously optimize the model based on feedback and evaluation results.

Performance Monitoring: Continuously monitor and evaluate the model's performance in real-world scenarios. Collect feedback from users and address any issues or limitations. Monitor the model's accuracy, robustness, and scalability and make necessary updates or improvements as required. Refer Figure 3.2

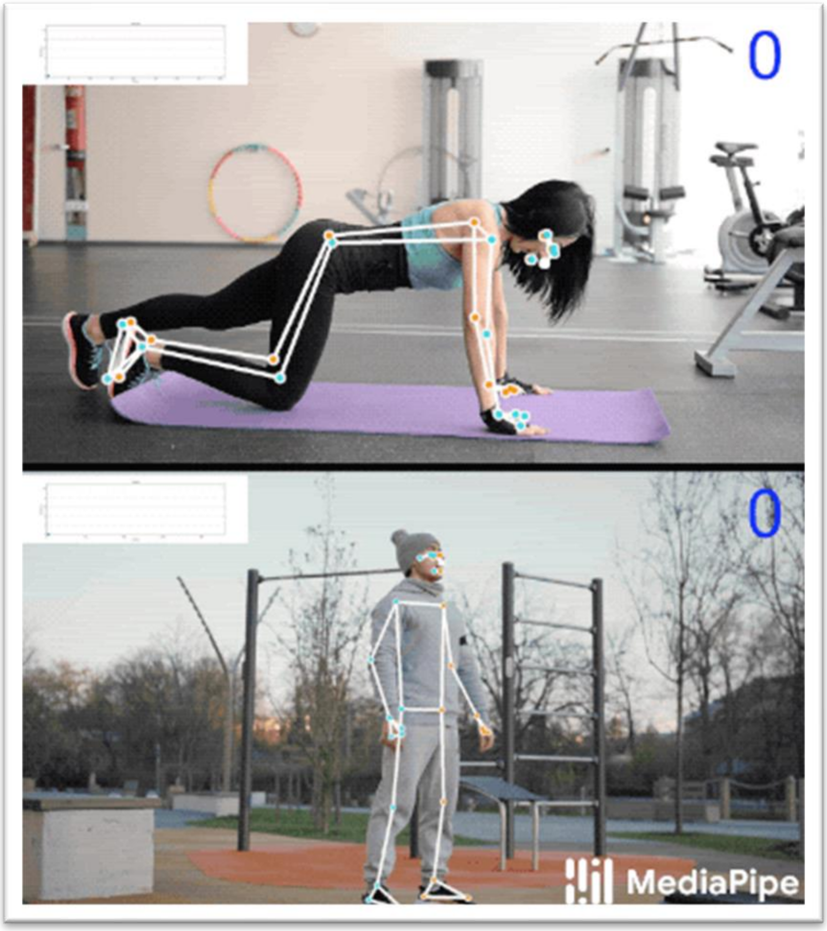

Figure 3.2 Media Pipe Model

3.3 Application Development

The methodology for application development involves creating a user-friendly and functional application that integrates the posture detection model. The following outlines the methodology for developing the posture detection application:

User Interface Design: Design the user interface (UI) of the application, keeping in mind the ease of use and visual appeal. Consider user experience (UX) principles to ensure a smooth and intuitive interaction for users. Create wireframes and prototypes to plan the layout and flow of the application.

Platform Selection: Determine the platform for application development, such as Android, iOS, or cross-platform frameworks like Flutter or React Native. Consider factors like target audience, platform popularity, and development resources to make an informed decision.

Framework and Tools: Choose suitable development frameworks and tools based on the selected platform. For example, if developing for Android, use Android Studio and Java or Kotlin. If using a cross-platform framework, select the appropriate tools like Flutter or React Native.

Integration of Posture Detection Model: Integrate the trained posture detection model into the application. Use the APIs and libraries provided by the machine learning framework used during model development. For example, if the model was developed using TensorFlow, utilize TensorFlow Lite for mobile applications.

Real-time Processing: Implement real-time image or video processing capabilities in the application to capture input from the device's camera. Use the device's camera API or libraries like OpenCV to capture frames and pass them through the posture detection model for inference.

Posture Visualization: Visualize the detected postures in a user-friendly manner. Display the posture information on the screen, such as highlighting body parts, showing correct posture alignment, or providing feedback on posture quality. Use appropriate UI components and visual cues to enhance user understanding.

Testing and Debugging: Conduct thorough testing and debugging to ensure the application functions as intended. Test various scenarios and edge cases to validate the accuracy and performance of the posture detection functionality. Fix any issues or bugs that arise during testing.

User Feedback and Iteration: Gather user feedback and iterate on the application based on the feedback received. Consider conducting user testing sessions to collect insights on usability, performance, and user satisfaction. Use the feedback to make necessary improvements and updates to the application.

Deployment and Distribution: Prepare the application for deployment to the respective application stores or distribution platforms. Follow the guidelines and requirements of the chosen platform to ensure successful submission. Monitor and address user feedback and reviews after deployment.

3.4 Evaluation

Accuracy: Measure the accuracy of the posture detection system by comparing the detected postures with ground truth annotations. Calculate metrics such as precision, recall, and F1 score to assess how well the system identifies correct postures and minimizes false positives and false negatives.

Real-time Performance: Evaluate the real-time performance of the posture detection system. Measure the system's ability to process frames or video streams in real-time, ensuring that it can provide instantaneous feedback on posture without significant delays.

Robustness: Test the robustness of the posture detection system by evaluating its performance under different conditions and variations. Assess how well the system handles variations in lighting, camera angles, clothing, and different body types. Ensure that the system can accurately detect postures across a diverse range of scenarios.

User Experience: Gather user feedback to evaluate the user experience of the posture detection system. Conduct user testing sessions or surveys to assess factors such as ease of use, clarity of instructions or feedback, and overall satisfaction with the system. Incorporate user feedback to make improvements and enhance the user experience.

Comparison with Baseline: Compare the performance of the developed posture detection system with baseline or state-of-the-art models in the field. Evaluate how the system's accuracy, speed, and robustness stack up against existing solutions. Identify the unique contributions and advantages of the developed system.

By considering these points during the evaluation of the posture detection system, you can gain insights into its accuracy, real-time performance, robustness, user experience, and its position in relation to existing solutions.

3.5 Additional Features

Progress Tracking: Enable users to track their yoga practice progress and monitor performance over time.

Exercise Recommendations: Provide personalized recommendations for yoga poses, sequences, or guided classes based on the user's skill level, goals, and preferences.

Posture Improvement Tips: Offer real-time feedback and tips to help users improve their posture during yoga practice.

Personalized User Profiles: Allow users to create profiles with information such as fitness level and specific considerations, enabling customized recommendations and modifications.

CHAPTER-4

IMPLEMENTATION AND RESULTS

Chapter-4

IMPLEMENTATION AND RESULTS

The objective of this project is to develop a general exercise counting system using the Media Pipe library and Python. The program leverages computer vision techniques to detect and track human poses in real-time video input. Based on the identified poses, the system counts the number of exercises performed by an individual. This report provides an overview of the implementation details and the results obtained from the program.

4.1 Implementation

The exercise counting program is implemented through the following steps:

Library Installation: The Media Pipe library is installed using the pip package manager. This library offers pretrained models and APIs for pose estimation.

Pose Estimation Configuration: A Pose object from the Media Pipe Pose module is created with appropriate configuration parameters. The minimum detection confidence and minimum tracking confidence are set to 0.5, ensuring reliable pose estimation.

Input Source: The program supports two input sources: a video file or the default webcam. If no video file is provided, the program captures frames from the webcam.

Pose Detection and Exercise Counting: The program enters a loop where it continuously reads frames from the input source. Each frame is converted to RGB format and passed to the pose estimator. The pose estimator processes the frame and returns the pose landmarks if a person is detected.

Next, the program analyzes the detected pose landmarks to identify specific exercise movements. This can involve tracking joint angles (Figure 4.1), body positions, or specific gestures associated with the exercise being targeted.

If an exercise movement is completed, the program increments the exercise count by one. It keeps track of the previous exercise position to avoid counting the same exercise multiple times.

Visualization: For each frame, the program draws the detected pose landmarks and displays the exercise count on the frame. This provides real-time visual feedback to the user.

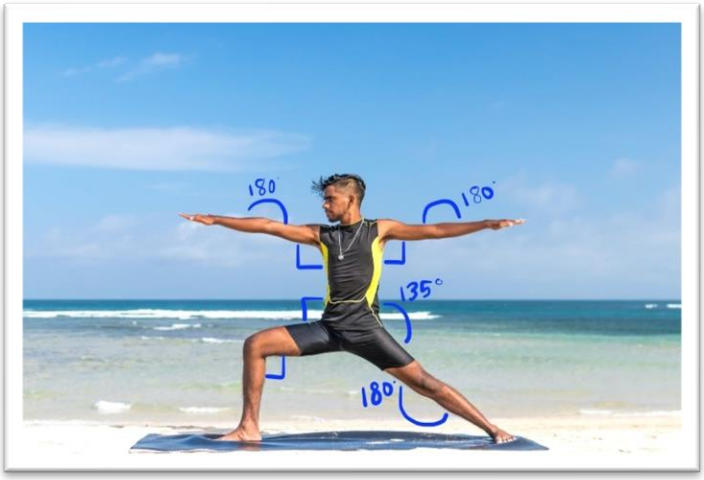

Figure 4.1 Angle Heuristic

Termination: The program can be terminated by pressing the "q" key.

4.2 Results

The exercise counting program was tested using various exercise routines and input sources, including video files and live webcam feeds. The following observations were made:

Accuracy and Reliability: The accuracy and reliability of the exercise counting system depend on factors such as the quality of the input video, lighting conditions, and the adherence of the individual to proper exercise form. In optimal conditions, the program demonstrates satisfactory accuracy in counting exercises.

Real-time Performance: The implemented program achieves real-time performance, processing and analyzing video frames at a speed that allows for smooth visualization and continuous counting.

User Feedback and Visualization: The program provides valuable visual feedback by drawing the pose landmarks and displaying the exercise count on each frame. This feature enhances the user experience and assists in monitoring exercise progress.

Limitations and Future Improvements: The program may face challenges in scenarios where occlusions occur or when the pose estimation fails to accurately detect the relevant body landmarks. Additionally, it assumes a standardized exercise form and may not accurately count variations in exercise movements.

Future improvements may involve incorporating machine learning techniques to train custom models for specific exercises, addressing the limitations mentioned above, and providing a more robust and versatile exercise counting system.

The aim is to map the key points for the bicep curl and the repetitions as shown in Figure 4.2.

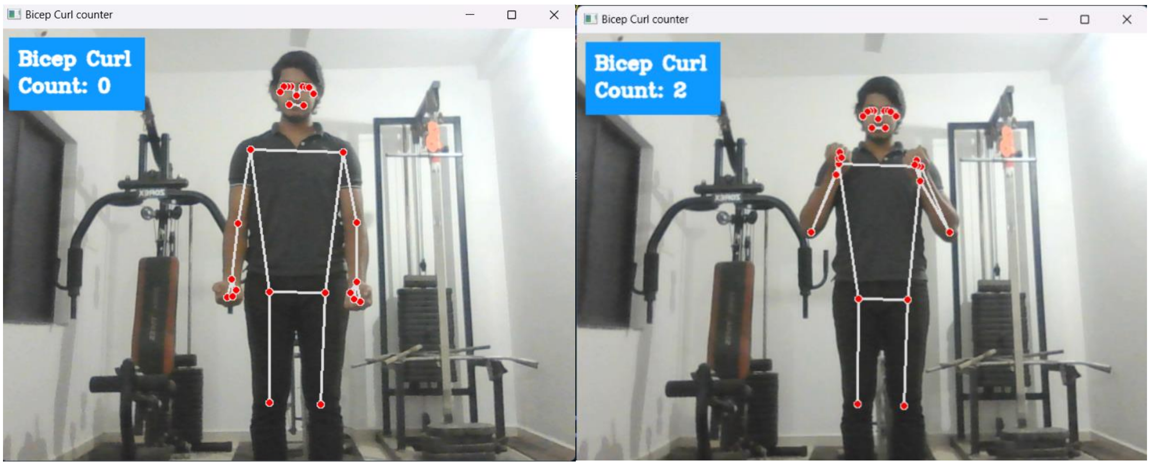

Figure 4.2 Implementation of Bicep Curl

The aim is to map the key points for the shoulder press and the repetitions as shown in Figure 4.3.

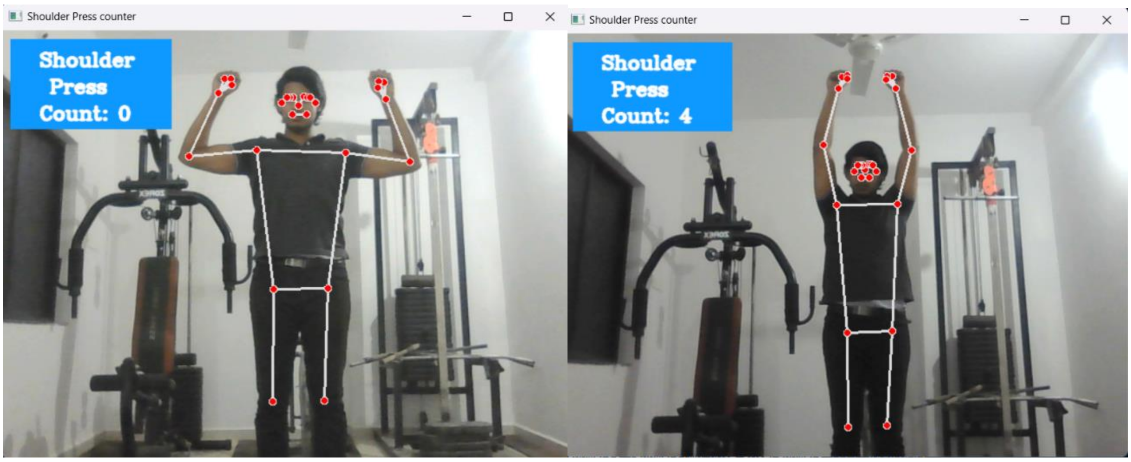

Figure 4.3 Implementation of Shoulder Press

BIBLIOGRAPHY

Paper Reference

[1] Malik, D. "360 Degree Posture Detection in Mobile and Desktop Version of Aasan." (2022) Research Commons Home, The University of Texas at Arlington. Retrieved from https://rc.library.uta.edu/uta-ir/handle/10106/30337

[2] Thoutam, V. A., Srivastava, A., Badal, T., Mishra, V. K., Sinha, G. R., Sakalle, A., Bhardwaj, H., & Raj, M. "Yoga Pose Estimation and Feedback Generation Using Deep Learning." (2022) Computational Intelligence and Neuroscience, 2022, Article ID 4311350. https://doi.org/10.1155/2022/4311350

[3] Upadhyay, A., Chaudhari, K., Bhere, P., & Thomas, J. "Body Posture Detection Using Computer Vision." (2020). SSRG International Journal of VLSI & Signal Processing, 7(1), 6-10. https://doi.org/10.14445/23942584/IJVSP-V7I1P102

Site Reference

[1] Google Developers. (n.d.). Pose Detection. Retrieved from https://developers.google.com/ml-kit/vision/pose-detection

[2] Google Developers. (n.d.). Classifying Poses. Retrieved from https://developers.google.com/ml-kit/vision/pose-detection/classifying-poses