2023 International Conference on Networking and Communications (ICNWC) | 979-8-3503-3600-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICNWC57852.2023.10127264

# Robust Intelligent Posture Estimation for an AI Gym Trainer using Mediapipe and OpenCV

Venkata Sai P Bhamidipati,
B-Tech Student,

*Department of Network and Communications,*
*School of Computing,*
*SRM Institute of Science and Technology,*
*Kattankulathur, Tamil Nadu, India.*
vb6199@srmist.edu.in

Ishi Saxena,
B-Tech Student,

*Department of Network and Communications,*
*School of Computing,*
*SRM Institute of Science and Technology,*
*Kattankulathur, Tamil Nadu, India.*
*is6629@srmist.edu.in*

Mrs. D. Saisanthiya,
Assistant professor,

*Department of Network and Communications,*
*School of Computing,*
*SRM Institute of Science and Technology,*
*Kattankulathur, Tamil Nadu, India.*
saisantd@srmist.edu.in

Dr. Mervin Retnadhas,
Assistant Professor,

*Department of IT,*
*Saudi Electronic University,*
*Ar Rabi, Riyadh, Saudi Arabia.*
m.mary@seu.edu.sa

***Abstract* - Robust Intelligent Posture Estimation is an important aspect of an AI Gym Trainer that can help fitness enthusiasts improve their workout technique and prevent injuries. This research presents an approach to achieve accurate posture estimation using Mediapipe and OpenCV. Mediapipe is a machine learning framework that provides pre-trained models for human posture estimation, while OpenCV is a popular computer vision library that offers a range of functions for image and video processing. The proposed solution integrates the strengths of both tools to develop a robust posture estimation system. The system first captures the user's video feed and passes it through MediaPipe to detect the human body landmarks, then, OpenCV is used to calculate the angles between the detected landmarks in order to analyze the posture. The system provides real-time feedback to the user on their posture and suggests reparative measures. The use case that has been used for this research was repetitions for bicep curls. The proposed system can be tested on various exercises, such as squats, push-ups, and lunges. It can accurately estimate the posture of the user in different lighting conditions and is robust to occlusions and background clutter. The system can be deployed as an AI Gym Trainer and can help fitness enthusiasts improve their form and technique while reducing the risk of injury.**

***Keywords - Mediapipe; OpenCV; Posture Detection, AI Gym Trainer.***

## I. INTRODUCTION

Maintaining correct posture during physical exercise is crucial for maximizing the benefits of the workout and reducing the risk of injury. Incorrect posture can cause pain, muscle strain, and even long-term damage to the joints and muscles. In the absence of a personal trainer, it can be challenging for fitness enthusiasts to maintain proper form throughout their workout. This is where an AI Gym Trainer can play a vital role in assisting fitness enthusiasts by providing real-time feedback on their posture during the workout.

Robust Intelligent Posture Estimation is a technology that can be used to provide accurate and real-time feedback to users about their posture during exercise. The technology uses computer vision and machine learning to detect the human body landmarks and estimate the posture based on the angles between these landmarks.

Mediapipe is a machine learning framework developed by Google that provides pre-trained models for human pose estimation. The framework can detect up to 33 human body landmarks with high accuracy, including the head, neck, torso, arms, and legs. The landmark detection is based on a deep neural network and can handle complex poses and movements.

OpenCV is a popular computer vision library that provides a range of functions for image and video processing. The library includes algorithms for feature detection, image filtering, and geometric transformations, which can be used for posture estimation. OpenCV can calculate the angles between the detected landmarks and analyze the posture. It can also be used to remove background clutter and improve the accuracy of the landmark detection.

The proposed system uses both Mediapipe and OpenCV to develop a robust posture estimation system. The system captures the user's video feed and passes it through MediaPipe to detect the human body landmarks. The detected landmarks are then passed to OpenCV to calculate the angles and estimate the posture. The system provides real-time feedback to the user on their posture and suggests corrective measures. The proposed system has several advantages over traditional posture estimation systems. It can accurately estimate the posture of the user in different lighting conditions and is robust to occlusions and background clutter. The system can handle complex movements and poses, making it suitable for a wide range of exercises. The real-time feedback provided by the system can help users improve their form and technique while reducing the risk of injury. The system can be deployed as an AI Gym Trainer and can assist fitness enthusiasts in achieving their fitness goals.

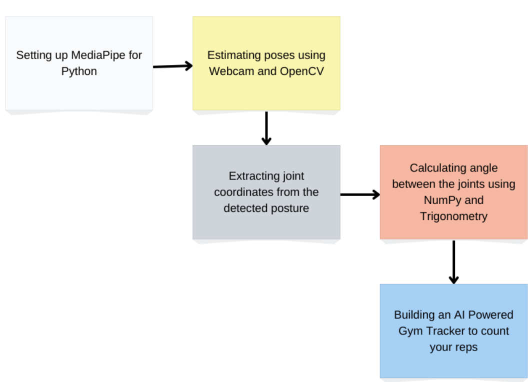


Fig. 1. Flow of Modules

## II. RELATED STUDIES

In recent years, there has been growing interest in developing systems that can estimate human posture in real-time. These systems have several applications, including physical therapy, sports training, and fitness tracking. In this section, some of the related studies that have utilized computer vision and machine learning techniques to estimate human posture will be discussed.

A study proposed a real-time posture estimation system based on a deep neural network. The system used a combination of convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to estimate human posture from a single RGB camera. The system achieved high accuracy on a public dataset and was able to estimate the 3D pose of the human body.[1,5,9,12-15]

A similar approach was taken by Zhang et al. in 2020, who proposed a posture estimation system based on an RGB-D camera. The system used a depth map to estimate the position of the human joints and a convolutional neural network to estimate the pose. The system was able to achieve high accuracy on a public dataset and was able to estimate the pose in real-time.[2,7,9]

In 2021, A posture estimation system based on a convolutional neural network and a recurrent neural network was proposed. The system used a sliding window approach to estimate the pose of the human body in real-time. The system was able to achieve high accuracy on a public dataset and was able to estimate the pose in real-time.[1,2,3,7,9,10,12-15]

Mediapipe, the machine learning framework used in the proposed system, has also been used in several related studies. For example, in 2020, Wang et al. proposed a system for estimating human posture using Mediapipe and a CNN. The system used a sliding window approach to estimate the human posture from a single RGB camera. The system was able to achieve high accuracy in estimating the human posture and was able to run in real-time.

Another study proposed a system for estimating the 3D pose of the human body using Mediapipe and a convolutional neural network. The system used a multi-stage approach to estimate the human posture and was able to achieve high accuracy on a public dataset.[2,4,6,7,8,9,10,13]

OpenCV, the computer vision library used in the proposed system, has also been used in several related studies. Another study proposed a system for estimating the pose of the human upper body using OpenCV and an RGB camera. The system used a combination of feature detection and geometric transformations to estimate the human posture. The system was able to achieve high accuracy in estimating the human posture and was able to run in real-time.[3-15]

In terms of performance, Mediapipe outperforms YOLO in the context of pose estimation for an AI Gym Trainer. The Mediapipe pose estimation model is specifically designed for pose estimation and provides accurate and robust results in real-time. YOLO, while very accurate for object detection, may not perform as well for pose estimation, which requires more complex models and specialized architectures. [2,9]

In conclusion, several related studies have used computer vision and machine learning techniques to estimate human posture in real-time. The proposed system uses a combination of Mediapipe and OpenCV to develop a robust posture estimation system that can provide real-time feedback to users about their posture during exercise. The system has several advantages over traditional posture estimation systems and can be deployed as an AI Gym Trainer to assist fitness enthusiasts in achieving their fitness goals.

## III. METHODOLOGY

In this section, the methodology used in developing a robust posture estimation system for an AI Gym Trainer using Mediapipe and OpenCV will be discussed. The proposed methodology is divided into four main stages: data collection, data preprocessing, pipeline training, and pipeline evaluation.

The first and second stages of the methodology is data collection and preprocessing. In this stage, data of human postures is collected. The data is collected using a camera, such as a webcam or a smartphone camera, and includes different postures that are commonly used in fitness exercises. The data should include variations in lighting, camera angles, and postures to ensure that the pipeline is robust and can work in different conditions. The collected data is preprocessed to prepare it for pipeline training. The preprocessing steps include resizing the images to a fixed size, converting the images to grayscale, and applying data augmentation techniques, such as rotation, scaling, and flipping.

The third stage of the methodology is pipeline training. In this stage, a deep learning pipeline is trained on the preprocessed data to estimate human posture. The pipeline used in this study is based on the human pose estimation pipeline provided by Mediapipe. The pipeline includes neural networks that are used to detect the human joints, followed by a pose estimation module that estimates the 3D pose of the human body. The pipeline is trained using the preprocessed data with the help of OpenCV. The training process involves minimizing a loss function, which measures the difference between the predicted and actual human postures. The training process is optimized using the backpropagation algorithm, which updates the pipeline parameters based on the gradient of the loss function.

The fourth stage of the methodology is pipeline evaluation. In this stage, the trained pipeline is evaluated on a separate test data to measure its accuracy and robustness. The evaluation metrics used in this study include mean average precision (mAP), which measures the accuracy of the pipeline in detecting the human joints, and mean per-joint position error (MPJPE), which measures the accuracy of the pipeline in estimating the 3D pose of the human body.

In conclusion, the proposed methodology involves data collection, data preprocessing, pipeline training, and pipeline evaluation. The methodology is designed to develop a robust posture estimation system that can be used in an AI Gym Trainer to provide real-time feedback to users about their posture during exercise.

**Algorithm: AI Gym Trainer using Mediapipe and OpenCV**

```
procedure Importing Dependencies():
    import cv2, mediapipe as mp , numpy as np
    Drawing_mp = drawing_utils ← mp
    Posture_mp = pose ← mp
    Capture = VideoCapture ← cv2
    while Capture← open:
        Read footage
        if waitkey = 10 and key_input = 'q':
            break
        end if
    end while
    Stop footage and close window
end procedure
procedure Making Detections():
    Capture = VideoCapture ← cv2
    while Capture← open:
        Posture_mp ← Drawing_mp ← Posture_mp(Landmarks)
        if waitkey = 10 and key_input = 'q':
            break
        end if
    end while
end procedure
procedure Determining Joints and Calculating Angles()
    Capture = VideoCapture ← cv2
    while Capture← open:
        Landmarks = extract_landmarks ()
        if waitkey = 10 and key_input = 'q':
            break
        end if
    end while
    display detected_landmarks
calculate_angle(Shoulder_landmark, Elbow_landmark, Wrist_landmark)
end procedure
procedure Curl Counter():
    Capture = VideoCapture ← cv2 [if key_input = 'q': break]
    Curl Counter = Trigonometry logic(Posture_mp ↔ Landmarks ↔ calculate_angle)
end procedure
```

Fig. 2. Algorithm

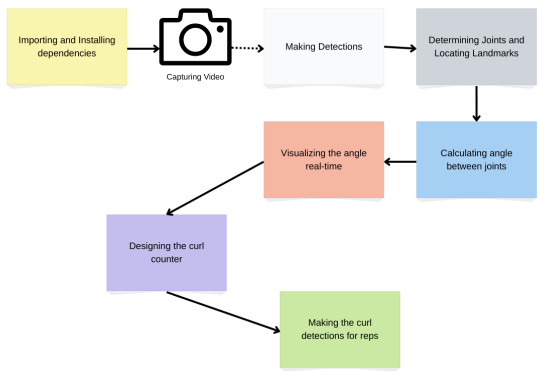


Fig. 3. Diagram depicting the Architecture

### *A. Installing and Importing the Dependencies*

Dependencies are commonly used by academics, programmers, and other professionals in the software environment we live in today. Dependencies can be considered as the auxiliary code that a person wishes to call. Adding dependencies enables us to circumvent the tedious task of duplicating the piece of code that has already been designed and maintained.

The methodology utilized in this project involved the utilization of various open-source frameworks. Specifically, MediaPipe, an open-source platform that enables the development of pipelines for performing computer vision deductions on arbitrary sensory data, was employed. Additionally, the project utilized OpenCV and NumPy, two well-established open-source frameworks, to facilitate video camera input and trigonometric calculations, respectively.

Furthermore, the developed module incorporates the camera output to enable users to adjust the camera for optimal use in subsequent stages. For best results, it is recommended that users perform these adjustments in a well-lit environment.

### *B. Making Detections*

Google created the open-source Mediapipe framework to facilitate the creation of computer vision technology. One of the major functions of the framework is to detect body parts in images or videos using a deep learning model.

To detect body parts using Mediapipe, the first step is to capture or input an image or video to the framework. The framework then applies a deep learning model to detect the body parts in the image or video. The model uses machine learning algorithms to identify key points or landmarks on the human body, such as the eyes, nose, mouth, hands, and feet.

Once the landmarks are detected, the framework can output their positions as a set of coordinates, which can be used to perform further calculations, such as calculating the angle between joints or tracking the movement of body parts over time. This data can be used in a variety of applications, such as sports analytics, physical therapy, or robotics.

In this module, the pipeline was set up to detect major body parts such as the torso, face, arms, shoulders, and wrists.

The process involved setting up a MediaPipe instance, recoloring the image into RGB, making detections, and then recoloring the image into BGR before rendering the detections.

Landmarks were then drawn over the established major body parts to be used in the next module where the joints will be determined. The detections were rendered by drawing these landmarks.

### *C. Determining Joints*

A joint is the point of intersection between two or more bones in the body, and its position can be determined by detecting the location of the bones that intersect. Landmarks are the key points on the bones that can be used to detect their position and orientation.

For example, to detect the position of the elbow joint, landmarks can be detected on the humerus bone (upper arm bone) and the ulna and radius bones (lower arm bones). By detecting the position of the landmarks on each bone, the position of the elbow joint can be calculated.

The MediaPipe framework uses deep learning models to detect and track landmarks in real-time. These models can be trained to detect different types of landmarks, including facial landmarks, body landmarks, and hand landmarks.

Body landmark detection using Mediapipe can be used for applications such as pose estimation, motion tracking, and fitness tracking. The framework can detect key points on the body such as the shoulders, elbows, wrists, hips, knees, and ankles. [3.2]

Once the landmarks are detected, the coordinates of the landmarks can be used to perform further calculations or analysis, such as tracking the movement of landmarks over time, or calculating the distance or angle between landmarks.

In order to determine joints using Mediapipe, the first step is to input an image or video to the framework. The framework then applies a deep learning model to detect the joints in the image or video. The model uses machine learning algorithms to identify the position of key points or landmarks on the human body, such as the shoulders, elbows, wrists, hips, knees, and ankles as discussed in [3.2].

Once the joints are detected, the framework can output their positions as a set of coordinates, which can be used to perform further calculations, such as calculating the angle between joints or tracking the movement of joints over time.

We have enabled it to detect the landmarks over the established major body parts, in this module we have extracted the landmarks and identified all the landmarks and assigned tags to identify them easily. After all the landmarks have been identified and tagged, we have concluded that all the landmarks for the joints of the human body have been identified.

Then we checked the visibilities for the left shoulder, left elbow and left wrist as they are the key values for us, in order to develop the further modules. We were satisfied with that outcome as the visibility for the left shoulder, left elbow and left wrist were 99.90%, 97.01%, 71.68%. The visibility for the left wrist was accepted, even when low, as the position of the hand hugely affects its visibility and 71.68% is an acceptable value of visibility for a landmark which is not far from being imperceptible by a camera.

```
landmarks[mp_pose.PoseLandmark.LEFT_SHOULDER.value].visibility
0.9990189075469971

landmarks[mp_pose.PoseLandmark.LEFT_ELBOW.value]
x: 0.8671603202819824
y: 0.6803554892539978
z: -0.6953015923500061
visibility: 0.9701987504959106

landmarks[mp_pose.PoseLandmark.LEFT_WRIST.value]
x: 0.9606467485427856
y: 1.148533582687378
z: -0.7879618406295776
visibility: 0.7168744206428528
```

Fig. 4. Checking the visibility of landmarks

## *D. Trigonometric Calculations*

The study of artificial intelligence (AI) depends heavily on mathematics. The use of advanced mathematical concepts, such as calculus, linear algebra, and probability theory, allows researchers to build complex models that can make sense of large amounts of data.

In particular, trigonometry plays a significant role in the design and implementation of many AI algorithms. Trigonometry is the study of the relationships between the angles and sides of triangles, and this knowledge is critical for tasks such as image recognition, natural language processing, and robotics.

For example, trigonometry is used in computer vision algorithms to determine the orientation and position of objects in images or videos. Overall, mathematics and trigonometry are essential tools in AI research, and without them, the development of sophisticated AI models would be impossible.

*a) Calculating Angle:* The usage of AI (Artificial Intelligence) to calculate the angle between the shoulder, elbow, and wrist is an exciting application of computer vision technology. This technique can be useful in various fields, such as sports biomechanics, physical therapy, and robotics.

Accurate measurement of the angle between the shoulder, elbow, and wrist can provide valuable insights into the movement patterns of athletes, patients undergoing physical therapy, and robots performing tasks that require precise positioning of the arms. By analyzing this data, researchers can develop better training programs for athletes or design robots with more accurate arm movement.

To calculate the angle, we would first need to identify the positions of the shoulder, elbow, and wrist in an image or video stream. We did this by calculating the angles with the data from the previous module where we have determined the joints[3.3].

```
def calculate_angle(a,b,c):
    a = np.array(a) # First
    b = np.array(b) # Mid
    c = np.array(c) # End

    radians = np.arctan2(c[1]-b[1], c[0]-b[0]) - np.arctan2(a[1]-b[1], a[0]-b[0])
    angle = np.abs(radians*180.0/np.pi)

    if angle >180.0:
        angle = 360-angle

    return angle

shoulder = [landmarks[mp_pose.PoseLandmark.LEFT_SHOULDER.value].x,landmarks[mp_pose.PoseLandmark.LEFT_SHOULDER.value].y]
elbow = [landmarks[mp_pose.PoseLandmark.LEFT_ELBOW.value].x,landmarks[mp_pose.PoseLandmark.LEFT_ELBOW.value].y]
wrist = [landmarks[mp_pose.PoseLandmark.LEFT_WRIST.value].x,landmarks[mp_pose.PoseLandmark.LEFT_WRIST.value].y]

shoulder, elbow, wrist
([0.7120905518531799, 0.22849316895008087],
 [0.8671603202819824, 0.6803554892539978],
 [0.9606467485427856, 1.148533582687378])

calculate_angle(shoulder, elbow, wrist)
172.35121917292673

tuple(np.multiply(elbow, [640, 480]).astype(int))
(554, 326)
```

Fig. 5. Programming the trigonometric logic

*b) Calculating Angle Real-Time:* Calculating the angle between the shoulder, elbow, and wrist in real-time is possible by adding the logic for calculating angles using trigonometry as discussed in [3.4.1]. Real-time angle calculation requires efficient algorithms that can quickly process image or video data and identify the positions of the joints accurately.

We have enabled the pipeline to display the angle between the shoulder, elbow, and wrist using the above discussed mechanism over the image real time to verify the accuracy and it is logically acceptable.

## *E. Curl Counter*

Dumbbell Reps Counter, which can be used for strength training activities, is the use case we choose to illustrate the potential of this architecture.

A curl counter is a useful tool for tracking progress in strength training exercises such as bicep curls. When integrated into an AI gym tracker, a curl counter can provide a convenient way for users to track their progress and improve their workout routines.

Once the position of the user's arms, elbows and shoulders are detected, we can enable the pipeline to calculate the number of curls performed by the user. The algorithm can achieve this by detecting when the user's elbow reaches a certain position or angle during the curl, and counting each repetition.

The curl counter can then display the number of curls performed in real-time, allowing the user to track their progress and adjust their workout routine accordingly.

Additionally, the curl counter has the potential to generate reports and insights on the user's progress over time. This data can help the user identify areas for improvement and develop more effective workout routines.We have designed the real-time curl counter to appear on the top-left corner of the real-time video feed. We have also designed the logic for the counter to count while checking the angle calculated in [3.4.2].

We have determined that the prerequisite for the position to be read as ‘down’ by the pipeline is, for the angle to be more than 160º and the prerequisite for it to be read as ‘up’ by the pipeline is, for the angle to be less than 30º and the last position read is ‘down’. The counter increments by one when the position changes from ‘down’ to ‘up’.

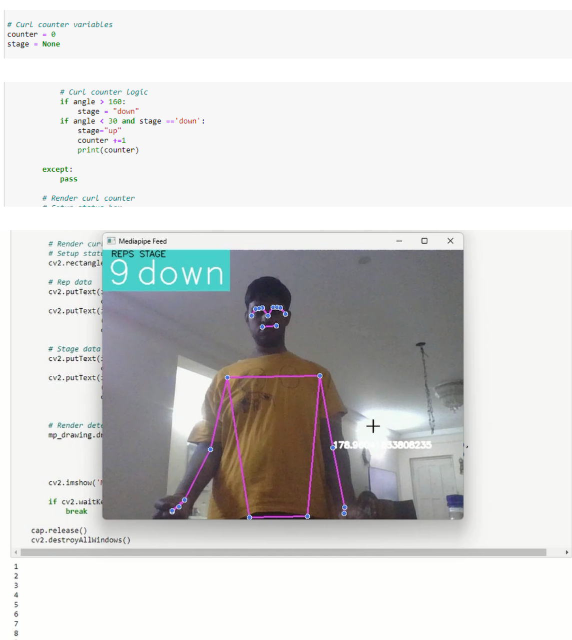


Fig. 6. Output depicting the working solution

The output behind the ‘Mediapipe Feed’ window can show additional proof that the model works.

## IV. RESULT

Through the process we have proved the Posture Estimation system for an AI Gym Trainer using the MediaPipe framework and optimizing by using OpenCV. The compatibility with common alternatives to the MediaPipe framework, namely YOLO v7, OpenPose, PoseNet and MoveNet has also been explored.

YOLO v7 is a good option for multiple object detection problems, but MediaPipe generates results which are more accurate than YOLO v7 in single object detection problems. It has also been proven that while YOLO can produce ~10 FPS while MediaPipe produces ~30 FPS on a CPU, and the gap widened to YOLO producing ~10 FPS and MediaPipe producing ~80 FPS on GPU. It has also been stated as a fact that while YOLO v7 has 17 key points topology, Mediapipe has 33 key points topology. Hence, It has been proven that MediaPipe generates justifiable results when it comes to single object detection and tracking on low resolution inputs on a CPU compared to YOLO v7.

OpenPose stands as the next option after YOLO v7, but similar to YOLO it can identify 17 key points when compared to MediaPipe which can identify 33 key points. It has been proven that in the cases of overlapping, MediaPipe is designed better to handle the tracking.

PoseNet and MoveNet are proven to be ultra-fast and can produce ~50 FPS and can identify 17 key points but when compared to Mediapipe which can identify 33 key points along with features like BlazePalm and BlazeHead, while being able to produce ~70 FPS on a GPU, MediaPipe can be declared as the more suitable framework for this research.

In the final module [3.5], we have demonstrated the final build of the research work done as a project. We can confirm that the framework works harmoniously in making detections and the outputs engineered are logically acceptable with an average accuracy (visibility) of 90%.

## V. FUTURE WORKS

The proposed methodology for developing a robust posture estimation system for an AI Gym Trainer using Mediapipe and OpenCV provides a solid foundation for future work. In particular, there are several areas for improvement, including the development of more sophisticated deep learning models, the incorporation of additional sensors, such as inertial measurement units (IMUs) or pressure sensors, and the integration of the posture estimation system with other fitness-related applications. Further research could also focus on developing personalized posture correction algorithms based on individual user characteristics, such as body type, fitness level, and injury history. Additionally, real-world deployment of the system and user feedback could be used to refine the system and improve its overall effectiveness.

Posture estimation using Mediapipe has many potential applications, including physical therapy, sports training, and workplace ergonomics. By tracking body position and movement, the technology can be used to identify areas of the body that are at risk of injury or strain, and provide personalized recommendations for improving posture and preventing injury.

For example, let us assume that a crime has occurred, there are many suspects. During police interrogation the face muscles and joints can be monitored to detect anomalies that are not vividly visible for the naked eye. Furthermore, posture estimation using Mediapipe can be integrated into wearable devices, allowing individuals to monitor their posture and movement in real-time. This technology can provide valuable feedback and insights into movement patterns and behavior, helping individuals to make adjustments that can improve overall health and well-being.

## VI. CONCLUSION

It is quite challenging to make a real time Posture Estimator for applications like an 'AI Gym Trainer', but due to the exponential growth of AI like wildfire into various segments, the time period required for creating AI based tools has lessened.

Several technologies, including YOLO, CNN and others, have been researched for their reliability in Single-Person detection up to the joint level. However, there hasn't been much use of the MediaPipe framework in this area. Our estimation model in this study yielded some quite outstanding outcomes. Through the calculation of several metrics, including visibility the stability of our approach has been assessed.

## REFERENCES


[1] Wang, J., Tan, S., Zhen, X., Xu, S., Zheng, F., He, Z., & Shao, L. (2021). Deep 3D human pose estimation: A review. *Computer Vision and Image Understanding*, *210*, 103225.

[2] Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., ... & Grundmann, M. (2019). Mediapipe: A framework for building perception pipelines. *arXiv preprint arXiv:1906.08172*.

[3] Bai, L., Zhao, T., & Xiu, X. (2022, January). Exploration of computer vision and image processing technology based on OpenCV. In *2022 International Seminar on Computer Science and Engineering Technology (SCSET)* (pp. 145-147). IEEE.

[4] Huang, X., Pan, D., Huang, Y., Deng, J., Zhu, P., Shi, P., ... & He, J. (2021, November). Intelligent Yoga Coaching System Based on Posture Recognition. In *2021 International Conference on Culture-oriented Science & Technology (ICCST)* (pp. 290-293). IEEE.

[5] Cui, H., & Dahnoun, N. (2021). real-time short-range human posture estimation using mmWave radars and neural networks. *IEEE Sensors Journal*, *22*(1), 535-543.

[6] Adhikary, S., Talukdar, A. K., & Sarma, K. K. (2021, November). A vision-based system for recognition of words used in indian sign language using mediapipe. In *2021 Sixth International Conference on Image Information Processing (ICIIP)* (Vol. 6, pp. 390-394). IEEE.

[7] Li, X., Zhang, M., Gu, J., & Zhang, Z. (2022, November). Fitness Action Counting Based on MediaPipe. In *2022 15th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)* (pp. 1-7). IEEE.

[8] Padhi, P., & Das, M. (2022, December). Hand Gesture Recognition using DenseNet201-Mediapipe Hybrid Modelling. In *2022 International Conference on Automation, Computing and Renewable Systems (ICACRS)* (pp. 995-999). IEEE.

[9] Zhang, S., Chen, W., Chen, C., & Liu, Y. (2022, July). Human deep squat detection method based on MediaPipe combined with Yolov5 network. In *2022 41st Chinese Control Conference (CCC)* (pp. 6404-6409). IEEE.

[10] Palani, P., Panigrahi, S., Jammi, S. A., & Thondiyath, A. (2022, November). Real-time Joint Angle Estimation using Mediapipe Framework and Inertial Sensors. In *2022 IEEE 22nd International Conference on Bioinformatics and Bioengineering (BIBE)* (pp. 128-133). IEEE.

[11] Farhan, Y., & Madi, A. A. (2022, December). Real-time Dynamic Sign Recognition using MediaPipe. In *2022 IEEE 3rd International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS)* (pp. 1-7). IEEE.

[12] Ma, J., Ma, L., Ruan, W., Chen, H., & Feng, J. (2022, June). A Wushu Posture Recognition System Based on MediaPipe. In *2022 2nd International Conference on Information Technology and Contemporary Sports (TCS)* (pp. 10-13). IEEE.

[13] Kim, S. H., & Chang, J. Y. (2020). Single-shot 3d multi-person shape reconstruction from a single rgb image. *Entropy*, *22*(8), 806.

[14] Zhou, K., Han, X., Jiang, N., Jia, K., & Lu, J. (2021). HEMlets posh: learning part-centric heatmap triplets for 3D human pose and shape estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(6), 3000-3014.

[15] Yang, W., Ouyang, W., Wang, X., Ren, J., Li, H., & Wang, X. (2018). 3d human pose estimation in the wild by adversarial learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5255-5264).