Chapter 6

Results

6.0 RESULTS

6.1 POSE ESTIMATION

Pose estimation is to detect human body poses from frames captured using a camera. The estimation models detect the key points of different body parts or joints such as eyes, ears, nose, shoulders, hips, elbows, knees, wrists, and ankles and collectively form a pose. A general example of key point extraction and pose estimation is shown in Figure 6.1.

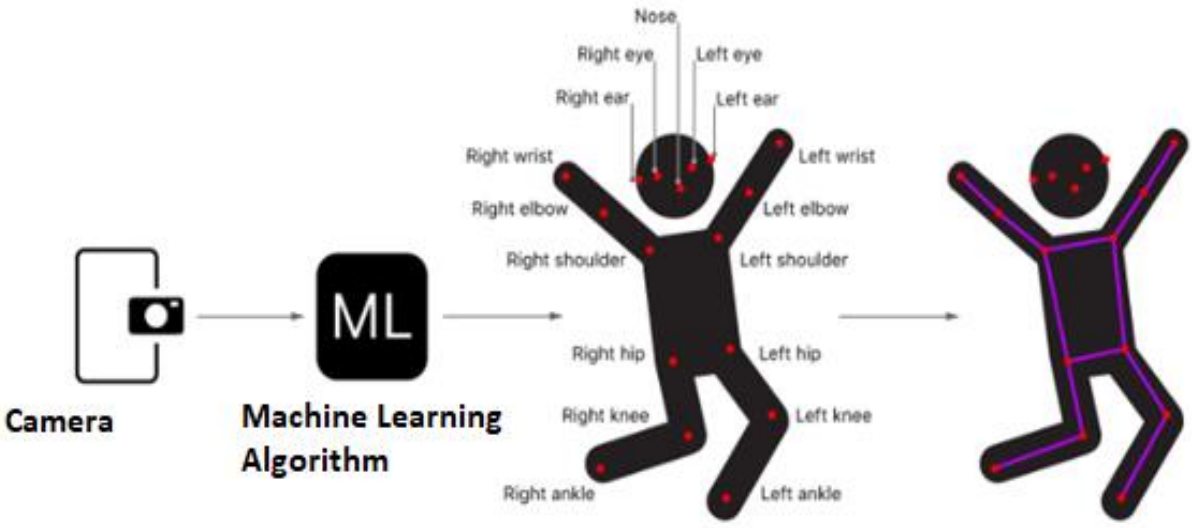


Figure 6.1: Human body Pose estimation

Pose estimation for five yoga postures has been done using different proposed techniques. The estimation images are shown for each of the five asanas for all the four architectures used. For simplicity, the images of the same individual are shown (after taking consent) for all estimations and comparisons. The five yoga poses considered for posture estimation are

i. *Ardhacandrāsana* (Half-moon pose)

ii. *Tāḍāsana* (Mountain pose)

ii. *Trikoṇāsana* (Triangular pose)

iii. *Vīrabhadrāsana* (Warrior pose-II)

iv. *Vṛkṣāsana* (Tree pose)

6.1.1 RESULTS OF POSE ESTIMATION USING EPIPOLAR POSE

The pose estimation results obtained for five yoga postures using Epipolar pose are as shown in Figure 6.2 (a-e)

Figure 6.2 (a-e). Key point detection by Epipolar Pose

6.1.2 Results of Pose estimation using Open Pose

The pose estimation results obtained for five yoga postures using Open pose is as shown in Figure 6.3 (a-e).

Figure 6.3 (a-e). Key point detection by Open Pose

6.1.3 Results of Pose estimation using PoseNet

The pose estimation results obtained for five yoga postures using posenet is as shown in Figure: 6.4(a-e)

Figure: 6.4 (a-e). Key point detection by PoseNet

6.1.4 Results of pose estimation using MediaPipe

The pose estimation results obtained for five yoga postures using MediaPipe is as shown in Figure 6.5 (a-e).

Figure 6.5 (a-e). Key point detection by MediaPipe

Pose estimation of the five yoga postures has been done for different methods as shown in Figure 6.2 (a-e) through Figure 6.5 (a-e). After validation of the model, 20 sample images were captured in real time, fed individually to the model, and estimated the posture accuracy. The average value of accuracy is tabulated as shown in Table:1. Here, the method used for calculating the accuracy is the classification score which is the ratio of the number of correct predictions (CP) made to the total number of predictions (TP) (i.e total number of predictions = the sum of CP and the number of wrong predictions (WP))

$$\mathrm{TP} = \frac{\mathrm{CP}}{\mathrm{CP} + \mathrm{WP}}$$

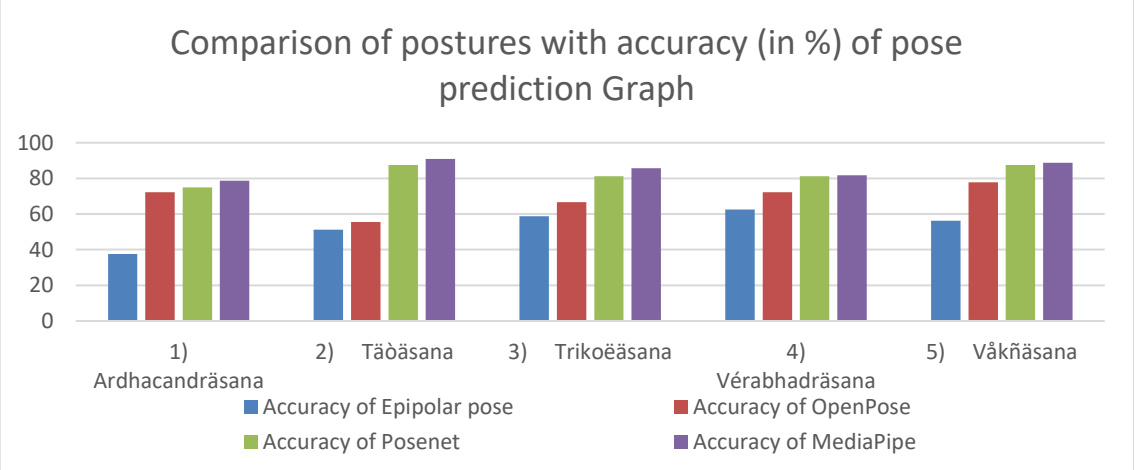
The accuracy of each of these could be increased further with an increase in the training

dataset but nevertheless, it clearly illustrates the comparative study between different pose estimation methods.

All the values in the table, are in terms of percentage (%).

Table 1. Comparison of postures with accuracy (in %) of pose prediction

Postures	**Accuracy of Epipolar pose**	**Accuracy of OpenPose**	**Accuracy of Posenet**	**Accuracy of MediaPipe**
1) Ardhacandrāsana	37.5	72.22	75	78.78
2) Tāḍāsana	51.25	55.55	87.5	90.9
3) Trikoṇāsana	58.75	66.66	81.25	85.75
4) Vīrabhadrāsana	62.5	72.22	81.25	81.81
5) Vṛkṣāsana	56.25	77.77	87.5	88.81



By observing the comparison in Table:1 keeping in mind the background light and contrast values, it is clear that MediaPipe provides better results and can estimate postures more accurately than other methods and hence it is the most suitable technique for pose classification.

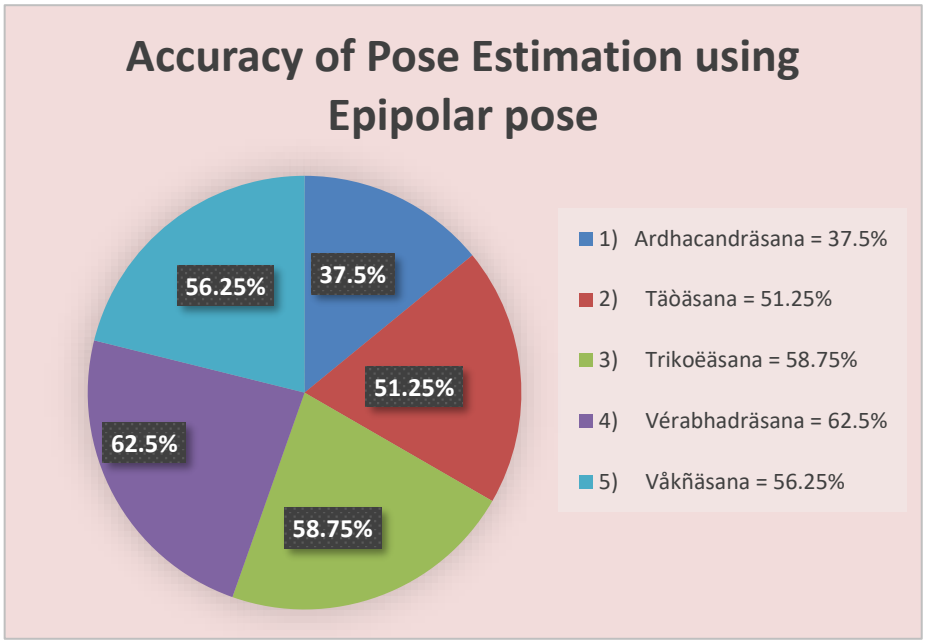


Figure 6.6: Comparative accuracy analysis for selected 5 postures using Epipolar Pose

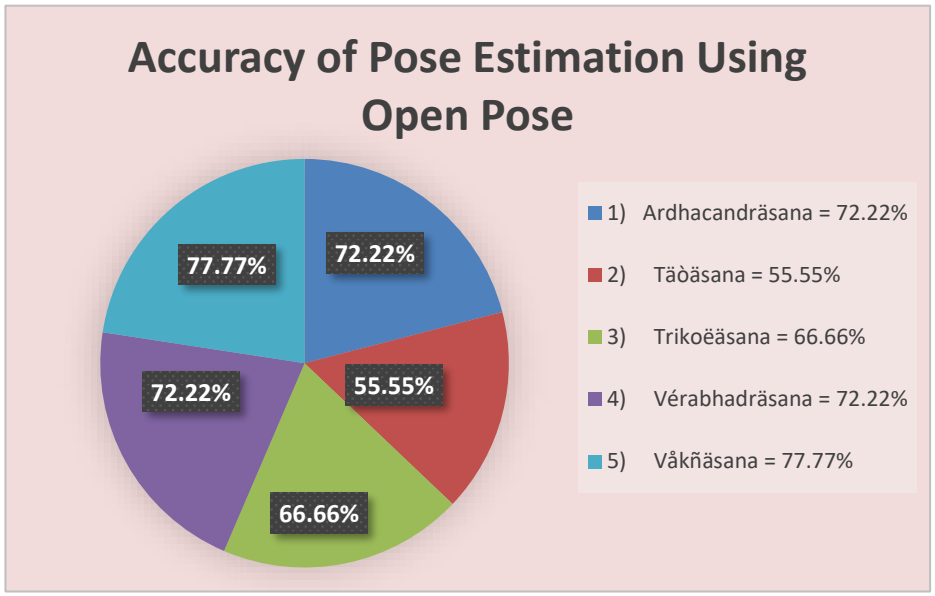


Figure 6.7: Comparative accuracy analysis for selected 5 postures using Open Pose

It is observed from Figure 6.6 that the accuracy of prediction using Epipolar Pose is around 50%. This is because the Epipolar pose is generally suited for describing and analyzing multi-camera vision systems dealing with two viewpoints of the same points in a pair of images (Chai et al., 2017). As this work involves capturing images from only one camera

the accuracy of pose is less and also may be observed that the number of key points detected is less as shown in Figure 6.2.

It is observed from Figure 6.7 that the accuracy of prediction using Open Pose is around 70%. Open pose is preferred for 2D pose detection for multi-person systems which includes body, facial, foot and hand key points (*Https://Www.Geeksforgeeks.Org.*, n.d.). It is reported to have been used for vehicle detection as well. This method of pose estimation suffers in estimating the poses when the ground truth has non-typical postures and also in estimating poses in crowded images leading to the overlapping of key points. The number of key points detected is more than Epipolar pose, yet during computation, the accuracy is compromised.

It has been reported that after using the fully connecting layer to detect the features using posenet the results have not improved as the network was likely to over fit the training data. In our work, pose net methods have given accuracy of about 80%. Figure 7.8 shows the key point detection by posenet. (Kim et al., 2021).

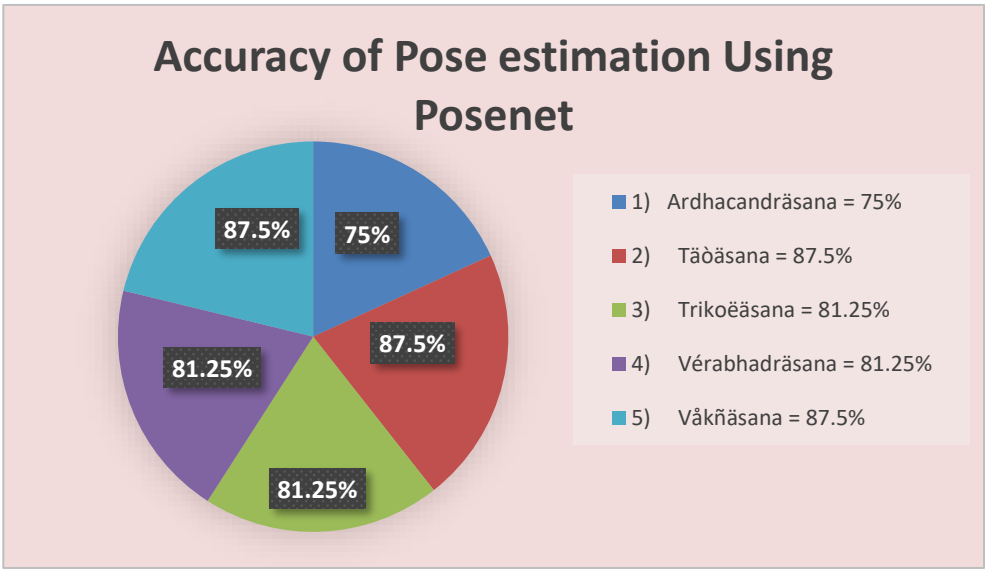


Figure 6.8: Comparative accuracy analysis for selected 5 postures using Posenet

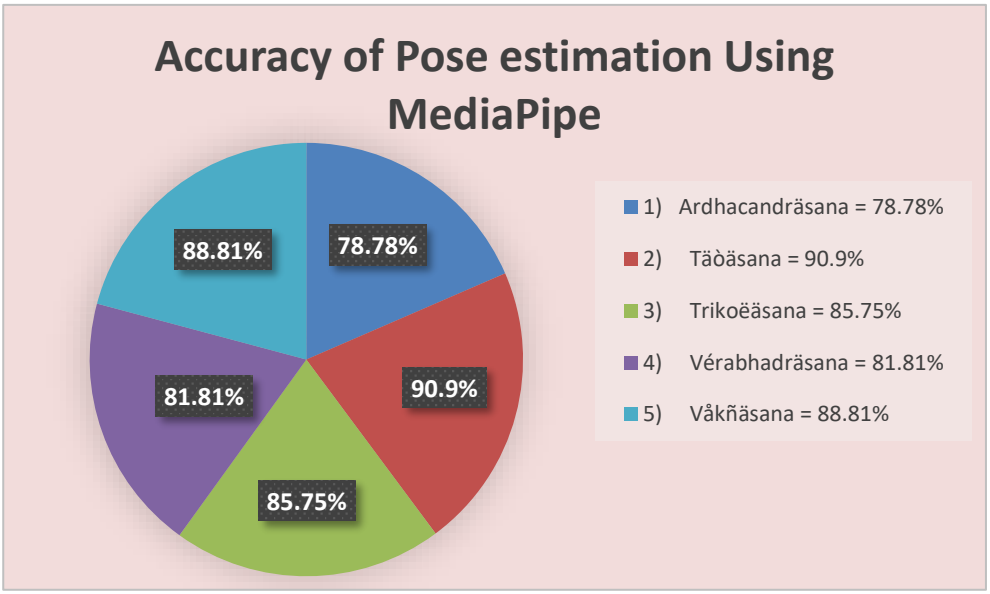


Figure 6.9: Comparative accuracy analysis for selected 5 postures using MediaPipe

Referring to Figure 6.9 mediapipe has better accuracy as compared to epipolar pose, open pose, and posenet. It may be observed from Table1 that the reason for less accuracy for other methods could also be due to pose estimation by using a single camera.

After comparison of all pose estimation methods it is concluded that Media Pipe provides better results and can estimate postures more accurately than other methods. The accuracy of of pose estimation could be increased by increase in the training dataset.

The present study used four different deep learning architectures such as Epipolarpose, Openpose, Posenet and Mediapipe for pose estimation to evaluate yoga postures. The results support the fact that media pipe has better accuracy compared to the other methods in spite of using a single camera. Further research would be needed to expand this technique for other advanced postures for pose estimation and correction using the same methodology which involves simple tools with better accuracy to assist individuals practicing Yoga postures as a self-evaluation as well as a bio-feedback mechanism

6.2 Pose Correction

Pose correction is to provide advice on the correct sitting/standing or posture related to specific activity. The pose correction is all about extracting the angles made between the body joints and comparing them with the reference angles for giving advice on the corrective measures. The database of images is represented by matrix of angles made by the different joints. When the user starts performing the pose in front of the camera, geometric analysis is done and compared with the reference angle in the database. If the calculated angles are beyond the threshold corrective action can be suggested to the user for pose correction.

Pose correction in this work has been done by first extracting the 14 coordinate values using media pipe as shown in Figure 6.10. The angle at each joint, for example between shoulder, elbow and wrist can be calculated. Let P (x1, y1), Q (x2, y2) and R (x3, y3) represent the three joints (points) where point Q is the common point. Let the lines PQ and QR intersect at Q, then the angle between PQ and QR can be calculated by using the basic slope formula.

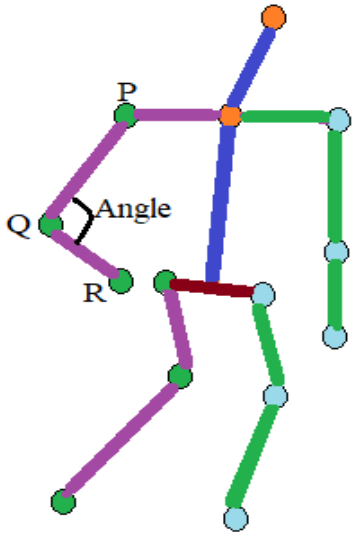


Figure 6.10: Sketch showing angle calculation at a key point

The slope of the line PQ is given as

$$m1 = \frac{y2-y1}{x2-x1} \qquad \text{Eqn.....1}$$

m1 is the slope for the line joining the point P and Q , where y1 and x1 are the coordinates of point P and x2 and y2 are the coordinates of point Q. similar concept is applied to the line QR.

Slope of the line QR is given as,

$$m2 = \frac{y3-y2}{x3-x2} \qquad \text{Eqn.....2}$$

m2 is the slope between the line joining the point Q and R. y2 and x2 are the coordinates of the point Q and x3 and y3 are the coordinates of point R. The point Q is the common point between the three points P, Q and R and so the angle is formed at the joint Q and can be calculated as

$$\tan\theta = \frac{m1-m2}{1+m1*m2} \qquad \text{Eqn.....3}$$

The angle calculated can be either positive or negative, Inverse function of tan gives the angle made by PQ and QR.

Referring to Figure 6.10 PQ and QR can be considered as two bones or skeletal structures of the human body, assuming the line PQ as the elbow and line QR as the hand, the angle made between the elbow and hand can be calculated. On further applying this analysis to all the other joints we can calculate the angles made at each joint.

In this work, the image is divided into four quadrants and then compared to find the accuracy of the pose performed. The difference in the angle between the test and the reference image is used to correct the posture. if the difference is positive the correction direction is downwards and if negative it is upwards.

As an example various angles after attaining final posture in left and right direction for

Warrior II pose (shown in Figure 6.11) is as shown in Table2.

Figure 6.11: Final position for Warrior II Pose

Table 2. Key point values for warrior II pose in terms of angle

Key point values for warrior II pose in terms of angle			
Final position Seeing Left		**Final position Seeing right**	
Left Elbow	180^o	Left Elbow	180^o
Right Elbow	180^o	Right Elbow	180^o
Right leg	180^o	Right leg	90^o
Left leg	240^o	Left leg	180^o
Right Mid	90^o	Right Mid	90^o
Left Mid	90^o	Left Mid	90^o

Various angles after attaining final posture in left and right direction for Half Moon pose (shown in Figure 6.12) is as shown in Table 3.

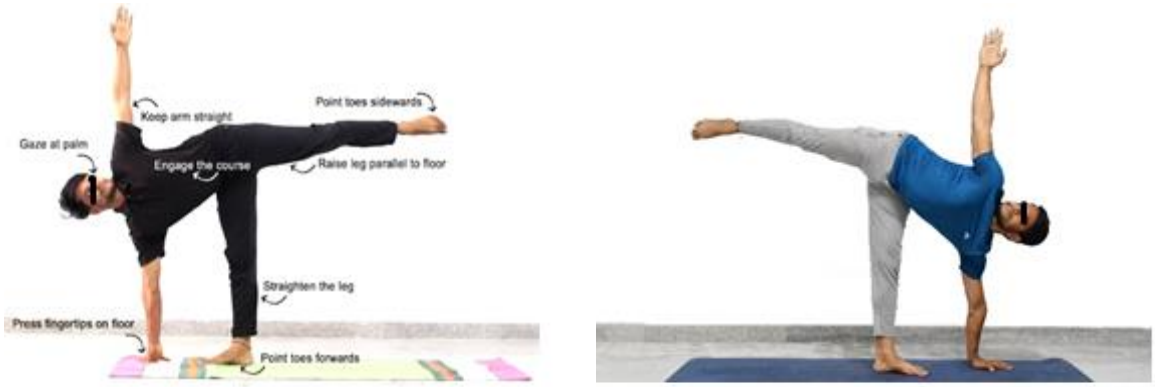


Figure 6.12: Final position for Half Moon Pose

Table 3. Key point values for Half Moon pose in terms of angle

Key point values for Half Moon pose in terms of angle			
Final position Seeing Left		**Final position Seeing right**	
Left Elbow	180º	Left Elbow	180º
Right Elbow	180º	Right Elbow	180º
Right leg	180º	Right leg	180º
Left leg	180º	Left leg	180º
Right Mid	90º	Right Mid	120º
Left Mid	120º	Left Mid	90º

Various angles after attaining final posture in left and right direction for Tree pose (shown in Figure 6.13) is as shown in Table 4.

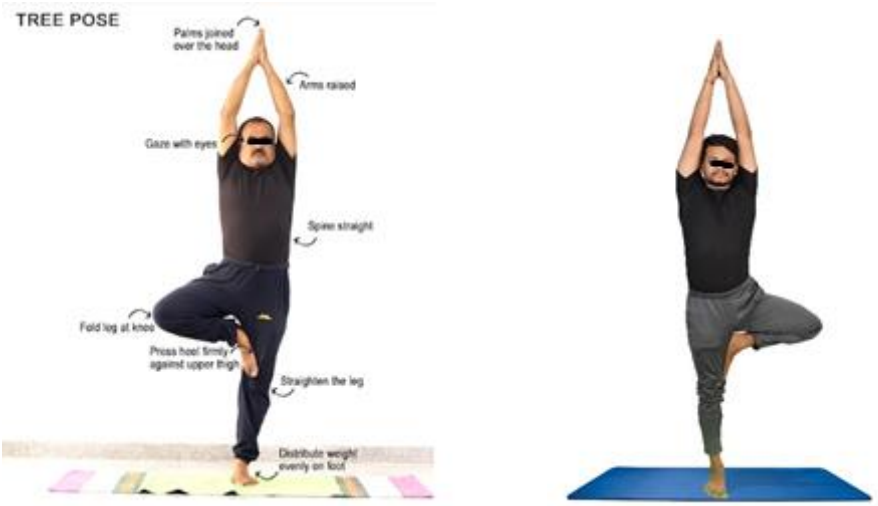


Figure 6.13: Final position for Tree Pose

Table 4. Key point values for Tree pose in terms of angle

Key point values for Tree pose in terms of angle			
Final position Seeing Left		**Final position Seeing right**	
Left Elbow	180^o	Left Elbow	180^o
Right Elbow	180^o	Right Elbow	180^o
Right leg	180^o	Right leg	50^o
Left leg	320	Left leg	180^o
Right Mid	180^o	Right Mid	180^o
Left Mid	180^o	Left Mid	180^o

Various angles after attaining final posture for Mountain pose (shown in Figure 6.14) is as shown in Table 5.

Figure 6.14: Final position for Mountain Pose

Table 5. Key point values for Mountain pose in terms of angle

Key point values for Mountain pose in terms of angle			
Final position angles			
Left Elbow	180^o	Left Leg	180^o
Right Elbow	180^o	Right Mid	30^o
Right leg	180^o	Leg Mid	30^o

Various angles after attaining final posture for Triangle pose (shown in Figure 6.15) is as shown in Table 6.

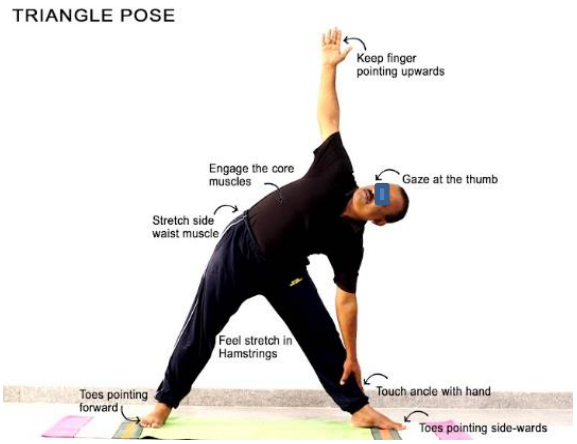


Figure 6.15: Final position for Triangle Pose

Table 6. Key point values for Triangle pose in terms of angle

Key point values for Triangle pose in terms of angle			
Final position Seeing Left		**Final position Seeing right**	
Left Elbow	180^o	Left Elbow	180^o
Right Elbow	180^o	Right Elbow	180^o
Right leg	180^o	Right leg	180^o
Left leg	320	Left leg	180^o
Right Mid	120^o	Right Mid	90^o
Left Mid	110^o	Left Mid	130^o

During the data preparation stage, the angles made for each yoga pose is calculated prior and stored in the database. This analysis is done to get all the angles for a particular yoga pose. These angles are calculated for all the five yoga poses prior to usage and are stored in the database for reference.

Giving feedback to the user of the performed pose is of utmost importance. This helps in

guiding the user to correct posture if wrongly done and thus learning to practice the yoga pose correctly. The feedback regarding the performance of the user is provided in real-time *via* the display and audio messages. When the user deviates beyond the threshold value the user is notified. Based on the varying levels of flexibility one can set the threshold as 10 degrees or 20 degrees in either direction. Users can observe the correction and make necessary adjustments to accurately practice the yoga routine. The feedback is given as a visual alert on the screen and also an audio message through a speaker so that the user need not turn their head in order to read the message on the screen. Also based on the yoga pose the user may be away from the screen and may not be able to correctly read the text shown on the screen. Thus, a Bluetooth-connected speaker could be a good alternative to send the feedback message to the user regarding the posture.

All the software and hardware is integrated as an Artificial Intelligent system to assist an individual to perform yoga postures accurately with proper guidance for pose correction.

The method of the complete posture correction system is shown as a flow chart in the Figure 7.16. Initially, the image of a yoga practitioner performing an Āsana is captured and fed to the media pipe, which is a pre-trained pose estimation model which detects human postures in images or videos by extracting the key points. All the key points are joined to form the skeletal structure. For posture correction a rule-based algorithm is used in which the input image is divided into 4 four quadrants and the key points lying within the divided quadrants were compared with standard key points. Using the trained dataset, real-time pose estimation and correction are implemented. If it does not match any of the selected yoga postures (Āsana) from the database, an error is shown and can start from the beginning.

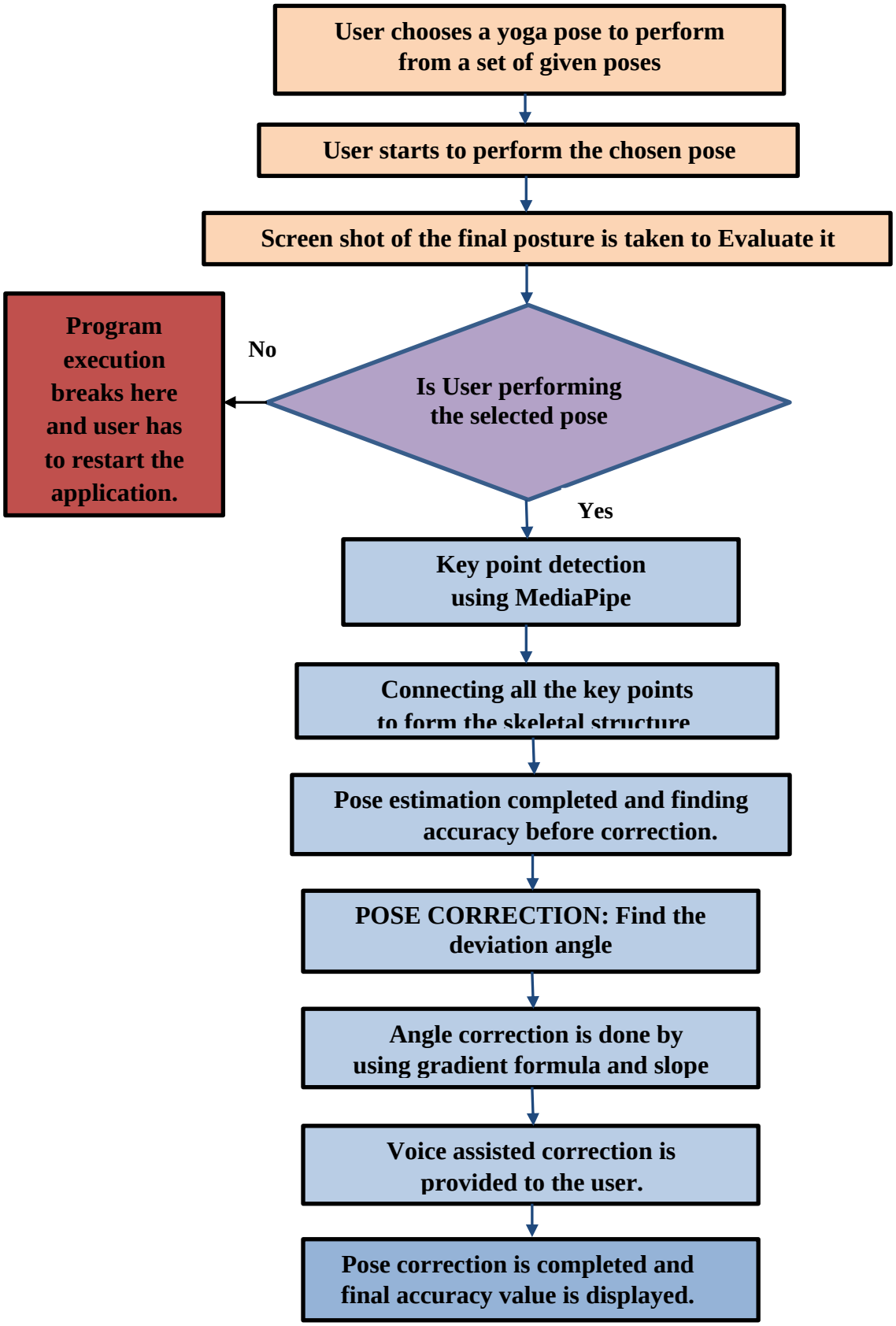


Figure 6.16: Flow Chart of the complete implementation of the system

Step by Step Description of the result from the viewpoint of the end-user.

The illustration of the output for warrior pose as an example is shown below. The flow of program shows the outputs through the voice assistance.

- The voice assistant greets you with a good morning/ good afternoon /good evening message and displays the yoga poses available for the user to perform.

- Voice input is obtained from the user regarding the pose her/she wishes to perform.

```
wishme()
query=takecommand().lower()
```

```
Good evening.
please select the yoga posture you want to perform

1 tree pose

2 mountain pose

3 half moon pose

4 triangle pose

5 warrior 2 pose
recognizing your speech
 you selected:Warrior 2 pose
```

- The voice assistant also gives the steps to be followed to perform the selected pose.

```
    print('Step 9 \n gaze or look at your front fingertips')
    speak('Step 9 \n gaze or look at your front fingertips')
    pilImage = Image.open("pics/warrior 2 instructions.jpg")
    fit_center(pilImage)
```

```
Here are the steps to  perform warrior 2 pose

Step 1
 Straighten one of your leg backside
Step 2
 back leg foot should be parallel to the ground
Step 3
 bend the other leg forwards such that front thigh is parallel to the floor
Step 4
 front knee should be directly over front ankle
Step 5
 Engage the core or stomach muscles
Step 6
 Hip must point sideways
Step 7
 Shoulder should be over your hips
Step 8
 Stretch both the arm at shoulder level
Step 9
 gaze or look at your front fingertips
```

- Once the input is obtained, a demo video of the selected *Āsana* is displayed to the user.

```
if "tree " in query:
    video('https://youtu.be/Dic293YNJI8')
elif "mountain " in query:
    video('https://youtu.be/p0XvVovdZsI')
elif "triangle " in query:
    video('https://youtu.be/S6gB0QHbWFE')
elif "half moon" in query:
    video('https://youtu.be/csErxVR7dpA')
elif "warrior " in query:
    video('https://youtu.be/Mn6RSIRCV3w')
```

```
Here is a demo video to perform warrior 2 pose
```

Figure 6.17: Demo video for selected pose

- In addition, an image of the final position of the selected pose is displayed for user convenience.

Figure 6.18 Demo of final position of the selected pose

- Now, the user has all the information about the Āsana he/she wishes to perform. The timing required to perform the Āsana (a trained person may require less time compared to others) is requested from the user. The time selected is displaced to the user using a down counter.

```
            # Display the clicked frame for 2
            # sec.You can increase time in
            # waitKey also
            cv2.imshow('a', img)

            # time for which image displayed
            cv2.waitKey(5000)

            # Save the frame
            cv2.imwrite('camera.jpg', img)
            break
            # HERE we can reset the Countdown timer
            # if we want more Capture without closing
            # the camera

# Press Esc to exit
    if k == 27:
        break

# close the camera
cap.release()

# close all the opened windows
cv2.destroyAllWindows()
```

```
performing pose detection and estimation
please perform the selected posture towards the left side
please select the time for perfoming selected posture
60
```

- When the user performs the pose, a screenshot at the 0^{th} second is captured and used for prediction.

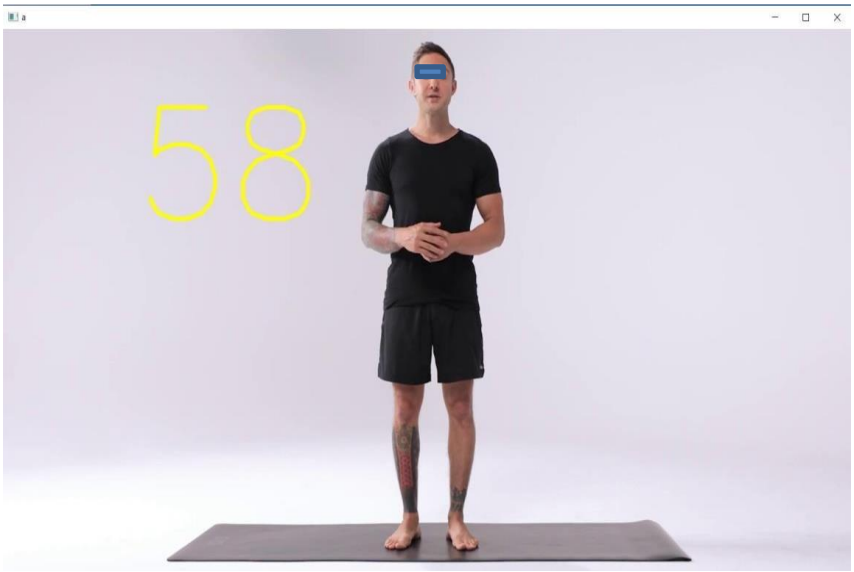


Figure 6.19: Count down for Final position of the selected pose

- The pose detection results

```
print(prediction)
print(f"According To Our prediction you are currently performing {prediction}")
speak(f"According To Our prediction you are currently performing {prediction}")
i=0
while (i==0):
    if prediction == query:
        pass
        i=i+1
    else:
        print("please perform the selected posture")
        speak("please perform the selected posture")
        break
```

```
[[0. 0. 0. 0. 1.]]
warrior 2 pose
According To Our prediction you are currently performing warrior 2 pose
```

- The key points detection using mediapipe on user pose are highlighted, and at the back end it is used for comparison with the reference image.

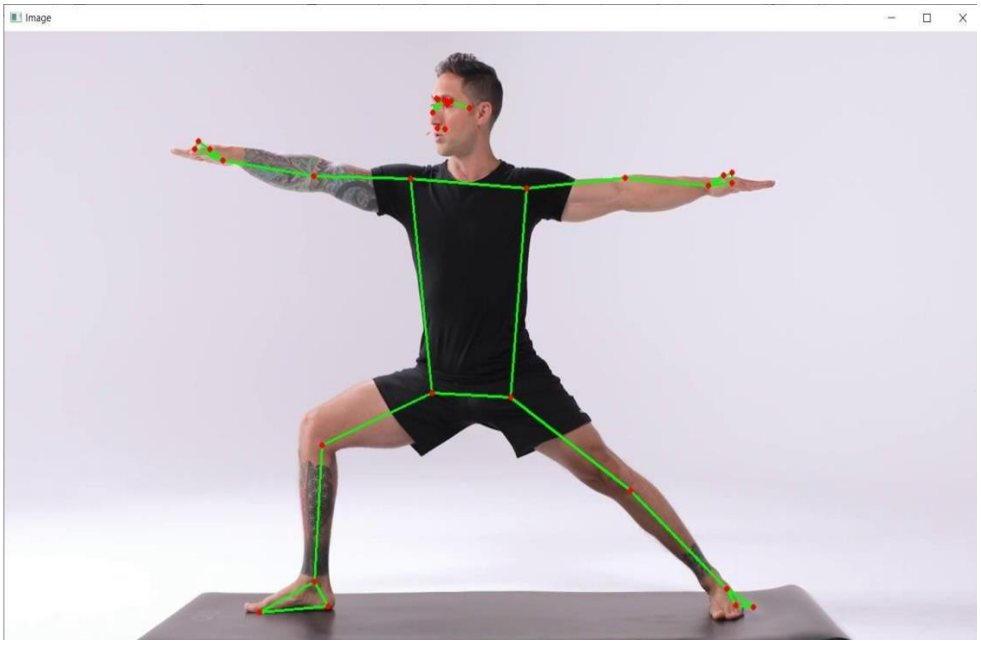


Figure 6.20: Calculation of angles for the final posture

- Calculation of the angles from the key points.

```
        cv2.waitKey(5000)
        break
        if cv2.waitKey(1) == 27:
            break
cv2.destroyAllWindows()
```

```
left arm
190
right arm
187
right leg
116
left leg
187
right mid
97
left mid
100
```

- The correctness of the pose is checked.

```
    elif "tree" in query:
        total=((p1+p2+p3+p4+p5+p6)/1220)
        total=int(total*100)
        print(f"your current posture accuracy without correction is {total} % ")
        speak(f"your current posture accuracy without correction is {total} % ")
    elif "mountain" in query:
        total=((p1+p2+p3+p4+p5+p6)/760)
        total=int(total*100)
        print(f"your current posture accuracy without correction is {total} % ")
        speak(f"your current posture accuracy without correction is {total} % ")
    elif "triangle" in query:
        total=((p1+p2+p3+p4+p5+p6)/940)
        total=int(total*100)
        print(f"your current posture accuracy without correction is {total} % ")
        speak(f"your current posture accuracy without correction is {total} % ")

beforeaccuracy(angleD1,angleD2,angleD3,angleD4,angleD5,angleD6)
print("To obtain Maximum accuracy for your posture please follow these instructions")
speak("To obtain Maximum accuracy for your posture please follow these instructions")
```

```
your current posture accuracy without correction is 91 %
To obtain Maximum accuracy for your posture please follow these instructions
```

- If there are any variations with respect to the reference image, then the correction in the form of voice and text is conveyed to the user. For example, if your accuracy is less, raise your hand by 10 degrees up/down.

```
        speak("right mid correct")
    else:
        print(f"strecth your right core by {abs(int(angleD5-110))} degrees")
        speak(f"strecth your right core by {abs(int(angleD5-110))} degrees")

    if int(angleD6)<=120 and int(angleD6)>=110 :
        print("left mid correct")
        speak("left mid correct")

    else:
        print(f"strecth your left core by {abs(int(angleD6-110))} degrees")
        speak(f"strecth your left core by {abs(int(angleD6-110))} degrees")
```

```
left arm correct
right arm correct
straighten your right leg by 63 degrees
bend your left leg by 52 degrees
right mid correct
left mid correct
```

- Based on the correction the user will be able to perform the yoga poses effectively. Hence pose correction is performed.

Summary

Human body pose estimation was done using four techniques for selected five yoga posture and concluded that mediapipe gives the best estimation accuracy compared to the other techniques. Pose correction was done in real time to give feedback on the pose performed.