Chapter 5

Methods

5.0 METHODS

The study on biomechanical analysis of the selected five standing postures focused on extracting the key points such as shoulder, wrist, elbow, knee, etc. from the images or videos and comparing the postures in the final position from the database by analyzing the joint angles for the five yoga postures: Ardhacandrāsana (Half-moon pose), Tāḍāsana (Mountain pose), Trikoṇāsana (Triangular pose), Vīrabhadrāsana (Warrior pose II) and Vṛkṣāsana (Tree pose) considered for the study.

This study involved two main parts as follows: Part I – Human pose estimation and correction, Part II – Validating the tools in healthy individuals for performance accuracy

5.1 PART I – HUMAN POSE ESTIMATION AND POSE CORRECTION:

The first part included selection of yoga postures, creating the authentic database, training the model with the data and programming for estimating and correcting selected yoga postures in real time. This included:

a. Human pose estimation by four different architectures and make a comparative study to choose the best architecture for pose estimation.

b. Human pose correction by comparing the joint angles to the reference database and providing corrective measures to attain the proper final position.

5.1.1 Selection of Yoga Postures:

The yoga postures selected for this study are

1. *Ardhacandrāsana* (Half-moon pose)
2. *Tāḍāsana* (Mountain pose)

3. *Trikoṇāsana (*Triangular pose)

4. *Vīrabhadrāsana* (Warrior pose II)

5. *Vṛkṣāsana (*Tree pose)

In this work single camera was used to capture the images, hence the person performing yoga needs to be facing the camera. Instead, if the person turns to either side or faces his back to the camera, though the program will be able to estimate the posture using the localized focal points, the reliability drops substantially. Only yoga postures that show all the limbs and the torso to the camera can be implemented in this method.

5.1.2 Procedure of data collection:

Collection of huge data was done due to the unavailability of datasets related to chosen yoga postures. For dataset creation the participants were asked to perform the following yoga postures: Ardhacandrāsana (Half-moon pose), Tāḍāsana (Mountain pose), Trikoṇāsana *(*Triangular pose), Vīrabhadrāsana (Warrior pose II) and Vṛkṣāsana *(*Tree pose).

- A total of 6000 images per posture were captured and developed into a database. As the data set had 6000 postures a majority of the images were taken from a high-definition (HD) Panasonic camera, web camera or by mobile camera of participants in their convenient place and time. Yoga pose videos and images were captured at 4m to 5 m distance in front of the camera.
- The yoga pose dataset was created comprising of both males and females performing at different locations at their convent place and time. To make the data realistic and to train the model for real-life environments the images and videos were captured in the living room, garden area, terrace and in studios. Deliberately few images without

proper illusion were also considered to enhance the ability of the model during training.

5.1.3 Creation of Authentic Database

- A large dataset of yoga postures, approx. 6000 images have been created to train a model that recognizes the correct posture.
- The images were taken in a room with good lighting and the person performed the correct posture. The person was dressed in contrast clothing with respect to the wall behind (i.e., if the wall is white, black or dark colored clothes are preferred) and pant were preferred to make the legs individually visible.
- The images can be captured on any device but they must be transferrable into different formats (i.e., png or jpeg) to be processed further. A database of images captured by the camera will be created from volunteers who wish to provide us with the yoga postures in which the entire body is visible.
- The images were taken from healthy people who practiced yoga regularly to maintain the accuracy ina database of the reference image.
- The data of images were taken from volunteers of different heights and weights.
- The data were collected with their consent allowing us to use their data for the purpose of this research.

5.1.4 Preparing the set of Images required to create a skeleton of the position of the person.

Real time images and videos can be processed for recognition and detection purposes using OpenCV (**Open-Source Computer Vision)** which is a Library used for artificial intelligence in python. OpenCV has optimized computer vision and machine learning algorithms to detect facial features, tract humans in videos, stitch images to enhance their resolution and many more. OpenCV was originally written in C++ and now has Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS.

Detecting an object in an image involves computer vision, image processing and deep learning techniques which deal with detecting the parameters of the object in an image or video. Object Detection using OpenCV is done by importing the library in python and use functions to perform object detection on an input image file by estimating unique points on the human body known as key points from which the skeleton is drawn on each image by joining the body parts. The images acquired is used to train a model to recognize the yogic postures.

5.1.5 Procedure to train the model

To train a model in machine learning is to predict the outcome of an event. The prediction could be good enough to measure the accuracy by testing the model. The data set is divided into training set and testing set in the ratio of 80% to 20% respectively in this work. The input data to be trained is passed through an algorithm to correlate the output against the sample. TensorFlow, an open-source library for machine learning is used to train the model. Tensorflow architecture works by preprocessing the data, build the model and then train and estimate the model. A flow chart of operations could be performed on the input to get a matrix representation of the data. TensorFlow executes operations to import and parse the training dataset, download it, and Load the data on to a device.

5.1.6 Repeated Measures Cross-sectional study

Subjects will be assessed at five different time points with a gap of 10 minutes between each set of practices. Subjects will be asked to perform the defined 5 postures as per the standard procedure. After completing one sequence of postures, 10 minutes’ gap will be given before starting the next sequence.

The reference model is trained using TensorFlow and the images used to train the model are annotated images of people performing yoga postures. One rather large caveat of this implementation is that all the limbs of the body, the torso and the face should be in clear view of the camera.

5.1.7 Procedure for estimating and correcting the posture

The image captured in real-time is sent to pose estimation model to determine the closeness to the reference image.

The key points are extracted and a skeleton of the person is drawn by joining the body parts. Then it is passed to pose estimation algorithm which gives the confidence score which indicate the closeness of the Position being performed by the person to the reference Position. A flow chart of the whole process is as shown below.

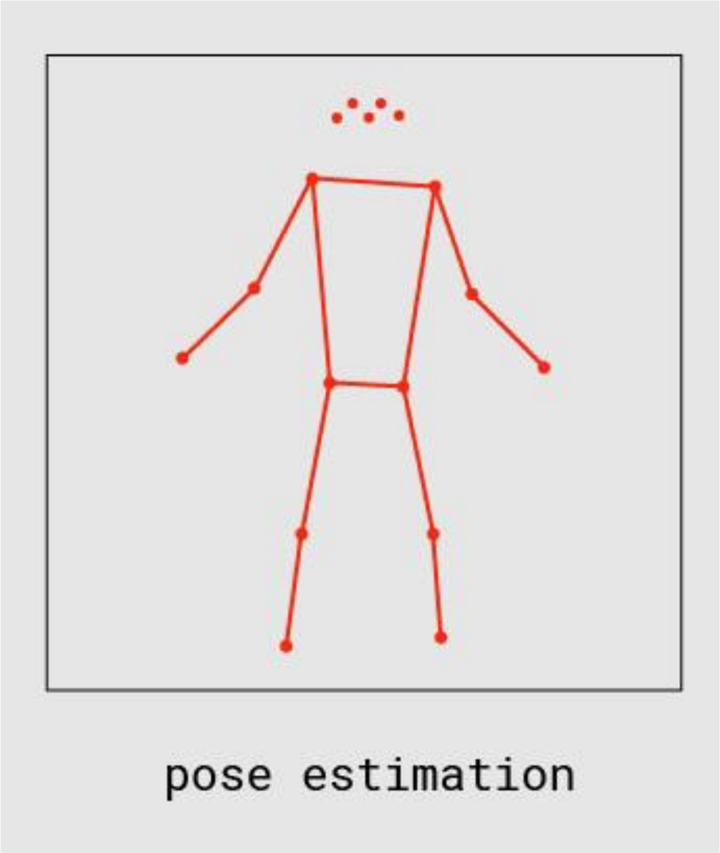


Figure 5.1: Key point extraction to form the skeletal view of the human body

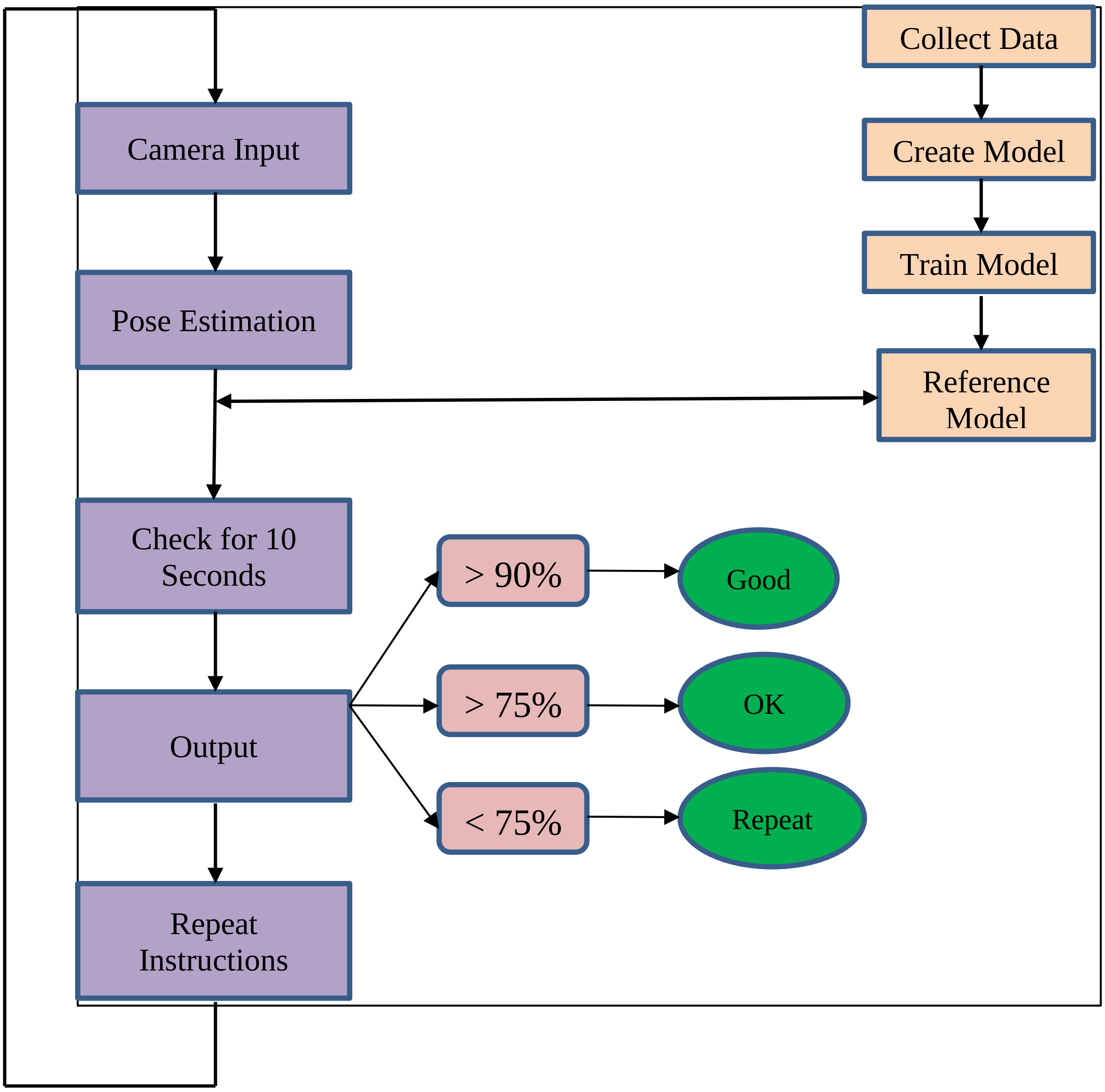


Figure 5.2: Flow Chart showing the pose estimation and correction Programming for pose estimation and correction

5.1.8 Software Tools used

- **Anaconda**

Anaconda (Nguyen et al., 2019) is a freely available open-source programming languages used for computing in machine learning applications. Open source packages can be individually installed from the Anaconda repository, Anaconda Cloud, or using the conda install command. Anaconda Navigator is a desktop graphical user interface

(GUI) that is used to launch applications and manage conda packages. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them.

- **Python**

  Python (Ghimire, 2020) is high-level programming language that help programmers to write clear, logical code for small and large-scale projects. python supports multiple programming models for Internet based applications which include creating a GUI and connect to databases.

  Coding is done by using Python language. The libraries such as Mediapipe, cv2, numpy, speech recognition, pyttsx3, math, datetime, pafy, vlc etc are imported to training and testing the deep learning model. cv2 is the openCV module used to read video by accessing the camera, numpy consists of multidimensional array and matrix data structures used for computing in Python. The speech recognition library is used for speech and audio interaction with the user and pyttsx3 is used for text-to-speech conversion library in Python. Math is a built in function for mathematical tasks to be performed in Python. Datetime library works with date and time, pafy library is used to download contents from you tube to retrieve metadata. vlc is a media player used for streaming media. sapi5 is a voice assistant used to interact with the user.

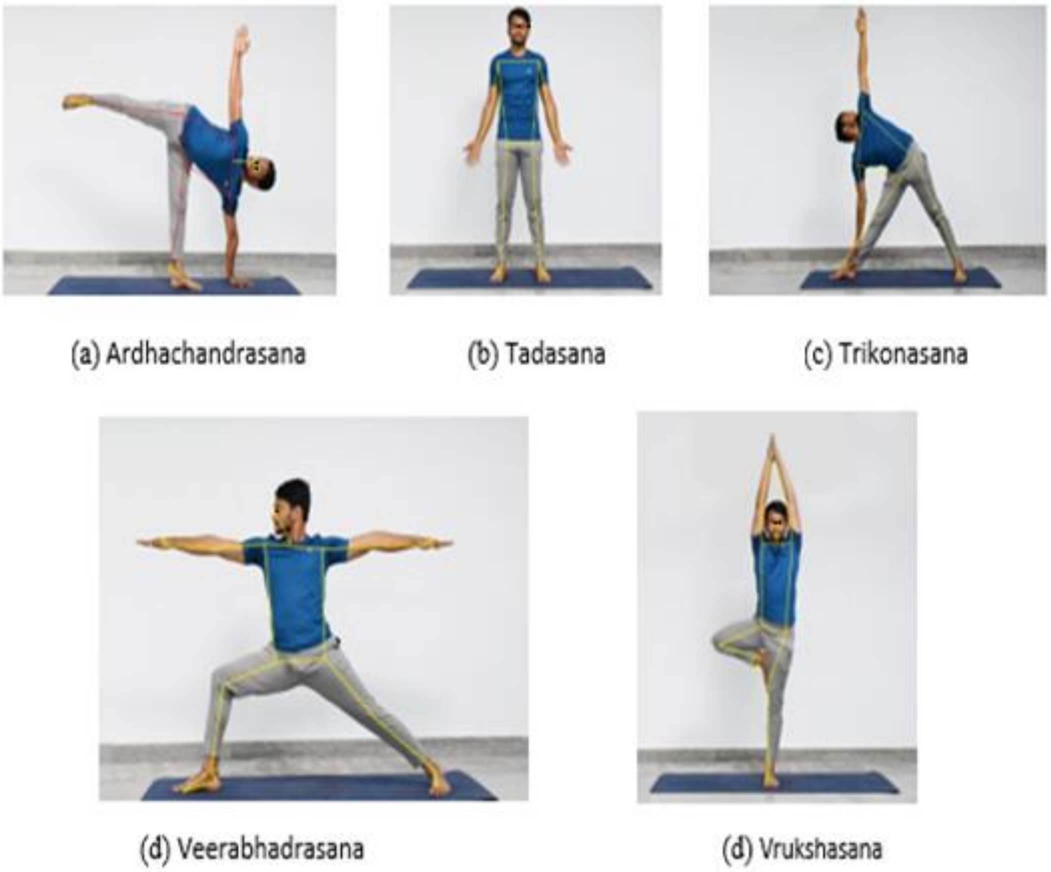
Testing the model

The programming was done to perform the following action.

- Display the required positions to be performed on the screen as well as give vocal instructions.
- Feed the data from the camera in a suitable format to the embedded device in real-time.

- Compare the data fed by the camera within the reference model and compare it with the trained network on it to check if the position is being formed properly.
- Give a confirmation of the position being properly formed and then move on to the next position.

- **Example of key point detection to estimate a pose**



(a) Ardhachandrasana

(b) Tadasana

(c) Trikonasana

(d) Veerabhadrasana

(d) Vrukshasana

Figure 5.3 (a-d) Showing pose estimation after Key point detection for the selected five asanas

5.1.9 Hardware Tools Used

- Camera is used to capture images and video to provide inputs for training and testing
- Microphone is used by user to give instructions and for interacting with the software.

- Display device is used for displaying demo video, output messages and other useful Information.
- The speaker is used to assist the user in the audio form (voice assistant) continuously during the course of performing the yoga posture.

5.2 PART II: Validating the tools in healthy individuals for performance accuracy

5.2.1 Participants (Sample size)

Twenty healthy practitioners of yoga belonging to both genders (10 female and 10 male) in the age group of 18 to 25 years [mean age (SD) 19.20 (4.2) years] were recruited based on the inclusion and exclusion criteria. All the participants were asked to perform five different postures, and in real-time, images were captured and fed separately to the trained model for pose estimation and correction. The sample size of 20 was based a previous study conducted in 10 subjects to assess Yoga based pose estimation using two-stage classifier and prior Bayesian network (Wu, Zhang, Chen, & Fu, 2019).

5.2.2 Selection and Source of Participants

All eligible participants were recruited from a Yoga University. These participants were the students of Bachelor of Naturopathy and Yogic Sciences (BNYS), Bachelor of Science (BSc), or Master of Science (MSc) in Yoga.

5.2.3 Inclusion Criteria

The participants should have been the residential students of the Yoga University with a minimum of two years of experience in practicing the selected yoga postures. Those who were healthy (mentally and physically) and did not have any serious health problem. Also, those who agreed to volunteer for the study by providing a signed informed consent were considered.

5.2.4 Exclusion Criteria

Participants consuming alcohol or smoking or having any other physical disability were excluded from the study. Participants with any significant injury to limbs and hands which interfered in their practice of asanas were also excluded. An online checklist was used to collect information including the practice of yoga.

5.2.5 Ethical Consideration

The study was approved by the Institutional ethics committee [IEC: RES/IEC-SVYASA/193/2021] and an informed consent was obtained from each participant after explaining the procedure involved in image acquisition. Their personal information was kept confidential and images acquired were used only for research purpose.

5.2.6 Selection of Yoga Postures

Five basic postures were considered in the present study which are commonly used not only for promoting positive health but also in yoga therapy. We have considered only standing postures as we will be using the input from one camera. The same program can be used to assess other categories also such as sitting postures, prone postures, and supine postures by adding additional cameras to capture the images from other angles.

The postures selected are:

The five yoga poses considered for posture estimation are

I. *Ardhacandrāsana* (Half-moon pose):

The subject has a yoga block handy at the front right-hand corner of the mat. Start in Warrior 2 with your right foot at the front of the mat, the front knee in line with your toes. Place left hand on the hip and reach out and then down with right arm, place fingertips in front of right toes. Step backfoot a bit forward, and shift weight into the right leg. As the subject press the right foot down, begin to extend the standing leg, as the left leg floats up in line with the hips. Place right hand on the block directly under the shoulder, towards the little-toe side of the right foot. To find stability in this pose, bring left leg slightly more forward rather than backward, as it will have the tendency to float in the space behind.

II. *Tāḍāsana* (Mountain pose): Tada means a mountain. Sarna means upright, straight, unmoved. *Sthitiis* standing still, steadiness. *Tāḍāsana*, therefore, implies a pose where one stands firm and erect as a mountain.

In *Tāḍāsana*, the arms are stretched out over the head, but for the sake of convenience, the subject placed them by the side of the thighs.

III. *Trikoṇāsana* (Triangular pose):

Stand straight. Separate your feet comfortably wide apart. Turn right foot out 90 degrees and left foot in by 15 degrees. The center of the right heel with the center of the arch of the left foot. Ensure that the feet are pressing the ground and the weight of the body is equally balanced on both feet. Inhale deeply and as exhale, bend your body to the right, downward from the hips, keeping the waist straight, allowing your left hand to come up in the air while your right hand comes down towards the floor. Keep both arms in a straight line.

IV. *Vīrabhadrāsana* (Warrior pose II): Stand in *Tāḍāsana*. Raise both arms above the head; stretch up and join the palms. Take a deep inhalation and with a jump spread the legs apart sideways4 to 4 1/5 feet. Exhale, and turn to the right. Simultaneously turn the right foot 90 degrees to the right and the left foot slightly to the right. Flex the right knee till the right thigh is parallel to the floor and the right shin perpendicular to the floor, forming a right angle between the right thigh and the right calf. The bent knee should not extend beyond the ankle but should be in line with the heel. Stretch out the left leg and tighten at the knee. The face, chest, and right knee should face the same way as the right foot, as illustrated. Throw the head up, stretch the spine from the coccyx and gaze at the joined palms.

V. *Vṛkṣāsana* (Tree pose): In this posture, the subject bend the right leg at the knee and place the right heel at the root of the left thigh. Rest the foot on the left thigh, toes pointing downwards. Balance on the left leg, join the palms, and raise the arms straight over the head. The same was repeated with the right leg.

PLATES 1-5: Line drawings of the selected postures used in the present study.

PLATE 1.

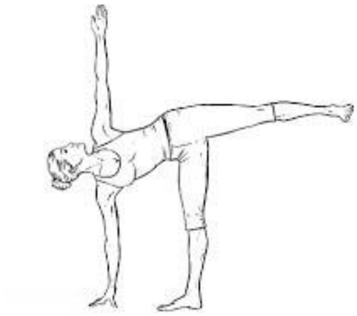


PLATE 2.

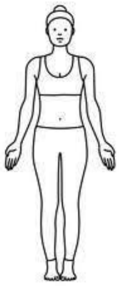


PLATE 3.

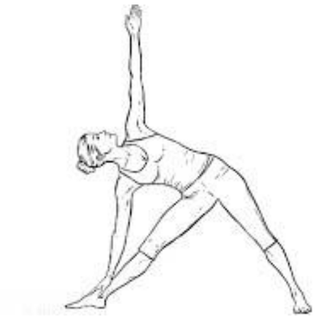


PLATE 4.

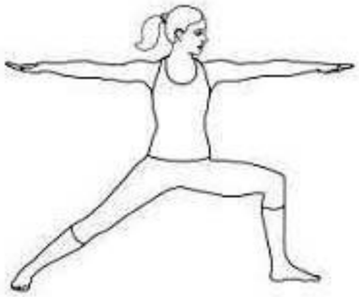
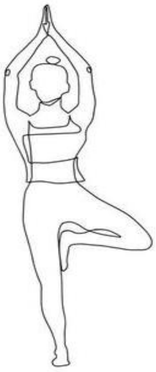


PLATE 5.



5.3 Data Extraction

5.3.1 First Part: Image Processing is a field where a whole set of techniques are used to process and analyze imagery data to extract valuable information used for a wide range of applications, such as identifying posture in digital images. Preprocessing the image is a prior step in computer vision, where the goal is to convert an image into a form suitable for further analysis. Examples of operations such as exposure correction, color balancing, image noise reduction, or increasing image sharpness are highly important and very care-demanding to achieve acceptable results in most computer vision applications like computational photography or even face recognition.

Some of the commonly used image processing techniques leveraging a very popular Computer Vision library are the OpenCV. Images in Computer Vision are defined as matrices of numbers representing the discrete color or intensity values present in every image pixel. Each image is considered as input data displayable in numerous ways, whether as arrays of pixel values or either multidimensional plots representing the distribution of pixel intensities. Images can be rendered in color, Grayscale and binary portraying black or white values only.

In OpenCV, images are converted into multi-dimensional arrays, which greatly simplify their manipulation. For instance, a grayscale image is interpreted as a 2D array with pixels varying from 0 to 255.Colored images deal with 3D arrays where each pixel is rendered in three different color channels. Pixel intensity distribution is done to identify and correct darkest areas in the image. A transformation function is applied to spread out the most frequent intensity values uniformly across the image. In Image processing, convolution is implemented which help in removing noise, blurring images, etc.

5.3.2 Second Part: Image processing of the recruited subjects for the pilot experimental work was completed using the software developed as mentioned above. Since the subjects were assessed while performing five different postures at five different time points, comparisons between each set of postures were done to ensure reproducibility.

5.4 DATA ANALYSIS

Data Analysis is a process of inspecting the data model that can be used for making decisions. Data analysis in this work has been by comparing the pose estimation by different architectures and choose the best one to apply pose correction techniques.

5.4.1 Data analysis of Pose Estimation

Pose estimation in this work has been done by four different architectures and compared for best estimation accuracy.

5.4.1a EPIPOLAR POSE

The Epipolar Pose is a self-supervised learning method used to construct a 3D structure from a 2D image of a human pose. 2D poses from multi-view images are used to generate a 3D pose using epipolar geometry that is used to train a 3D pose estimator (Wandt et al., 2021). It is used for 3D human pose estimation and tracking which doesn't need any 3D ground-reality data or camera intrinsic. It inputs an RGB image to give a 3D pose result. 3D Pose estimation is a difficult and a complex task as it involves gathering huge amount of 3D ground-truth data (L. Chen et al., 2021). To tackle this problem of 3D pose estimation, several self-supervised pose estimation methods have been suggested, but such methods along with 2D data require additional supervision of different kinds such as unpaired 3D ground truth data and a small subset of labels or the Multiview settings in the camera parameters. It is easier to deal with these kinds of problems by using Epipolar pose, which doesn't need any 3D ground-truth data for 3D human pose estimation. The set of images are applied to the epipolar architecture and the key point detection is as shown in Figure 5.4.

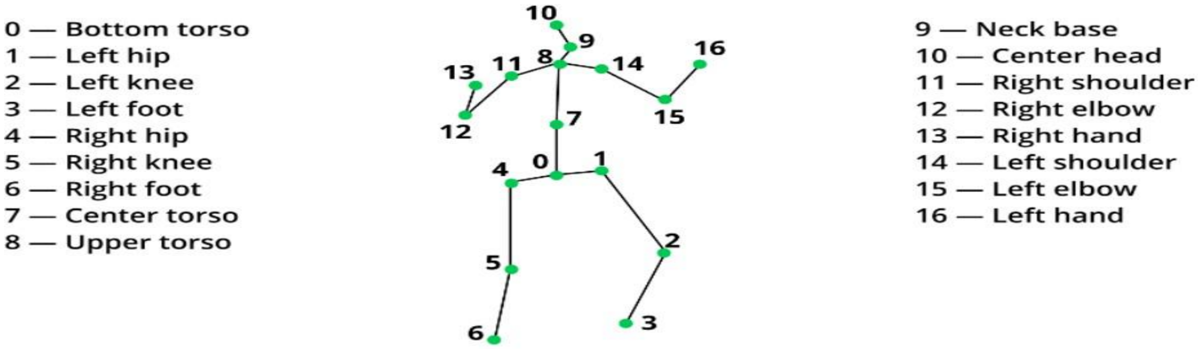


Figure 5.4: Key point detection using Epipolar pose.

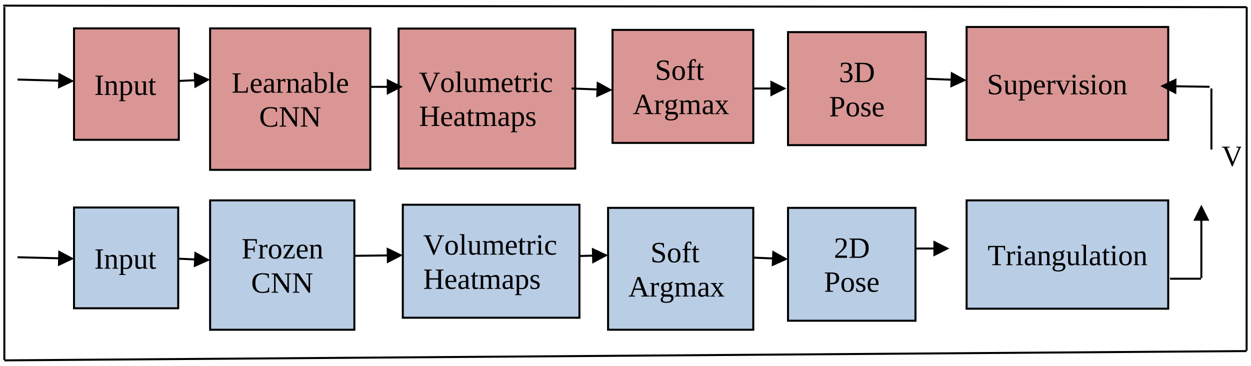


Figure 5.5: Architecture of Epipolar Pose involved during training.

Figure 5.4 shows the output skeleton view after joining all the key points given by Epipolar pose. The architecture of Epipolar pose is shown in Figure 5.5. The input block consists of the images captured from 2 or more cameras. These images are then fed to an CNN pose estimator. The same set of images are then fed to the training pipeline and after triangulation, the 3D human pose is obtained (V) is fed back to the upper branch. Hence this architecture is self-supervised.

The main advantage of this architecture is that it does not require any ground truth data (Muhammed, 2019). A 2D image of the human pose is first captured, then an epipolar geometry is utilized to train a 3D pose estimator (Kocabas et al., 2019). Its main disadvantage is requiring at least 2 cameras. The sequence of steps for training is shown in Figure 5.5 The upper row of the figure (orange) depicts the inference pipeline and the bottom row (blue) shows the training pipeline.

5.4.1b OPEN POSE

Open pose is the first real-time multi-frame architecture having hand, facial, and foot key-points in a single image (Mithsara, 2022). Open pose is one of the most popular procedures used to detect key points as shown in Figure 5.6. In the first few layers, the Open pose architecture extracts functions from an image and then fed to the parallel branches of convolutional layers. In the first branch a set of set of maps are predicted which points to a specific part of the human pose skeleton. In the second branch a set of affinity fields related to the credentials of association among other parts are predicted.

Open Pose is another 2D approach for pose estimation (C. H. Chen et al., 2019). The Open pose architecture is shown in Figures 5.7a, 5.7b, and 5.7c.

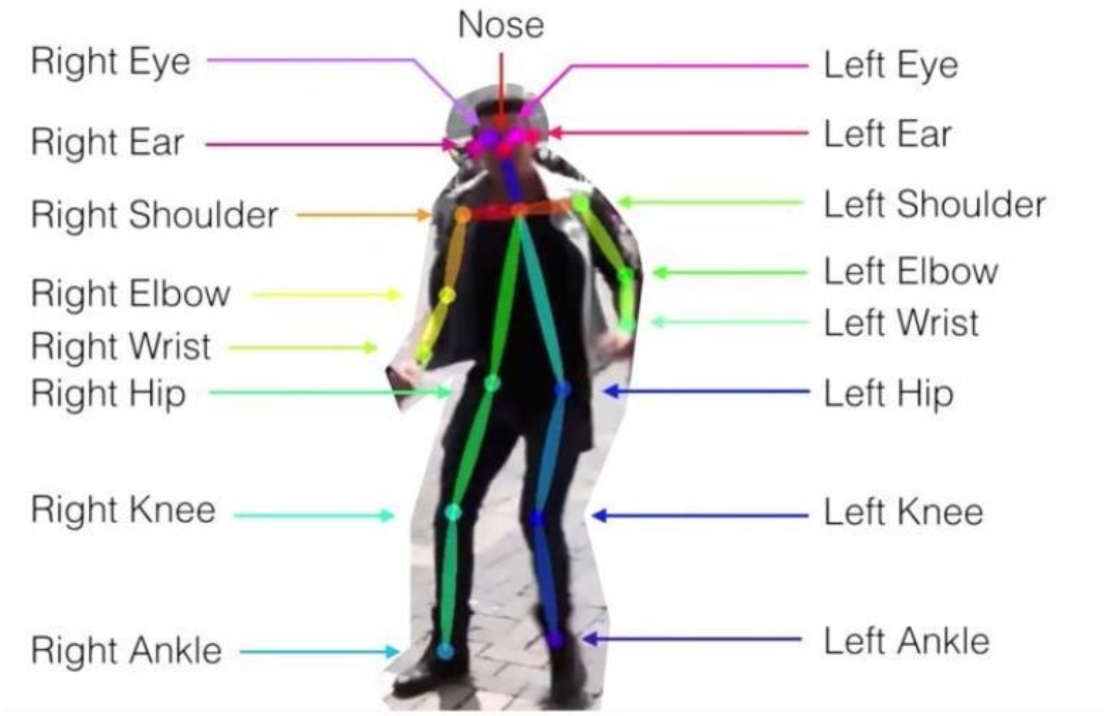


Figure 5.6: Key point detection using Open pose

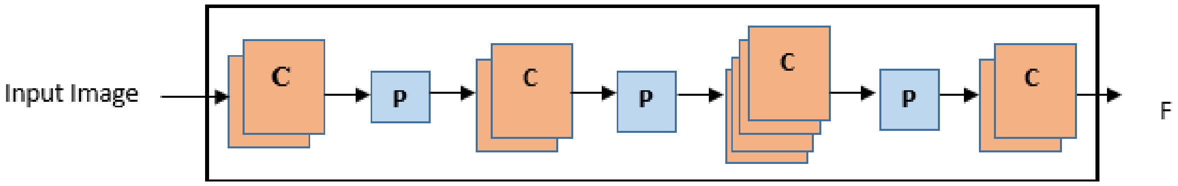


Figure 5.7a. VGG 19 Convolution Neural Network (C-Convolution, P-Pooling)

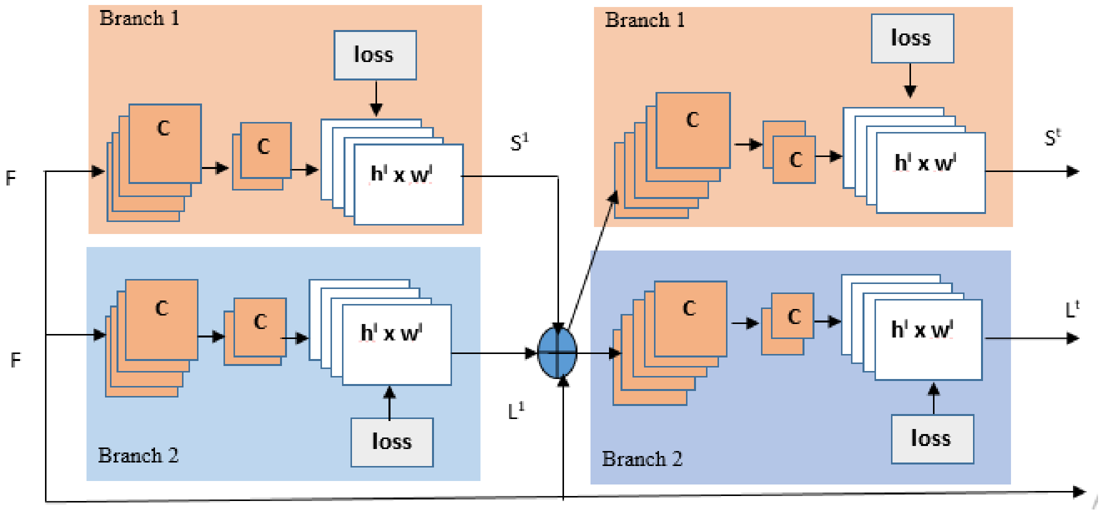


Figure 5.7b. Convolution Layer Branches

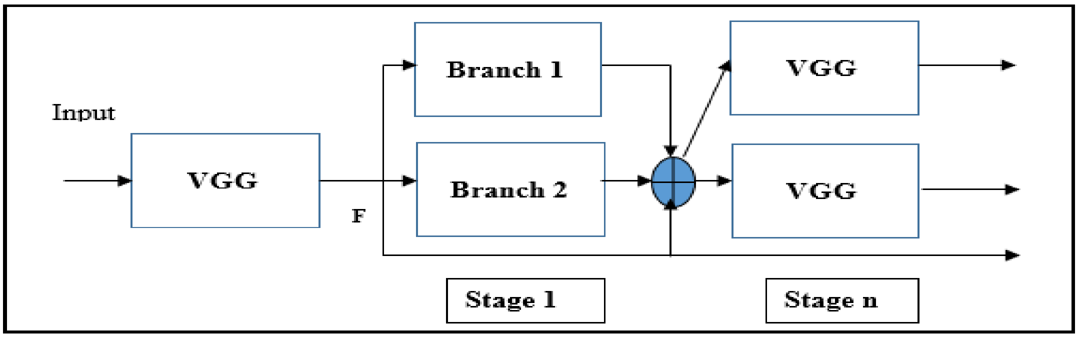


Figure 5.7 c. Open pose architecture

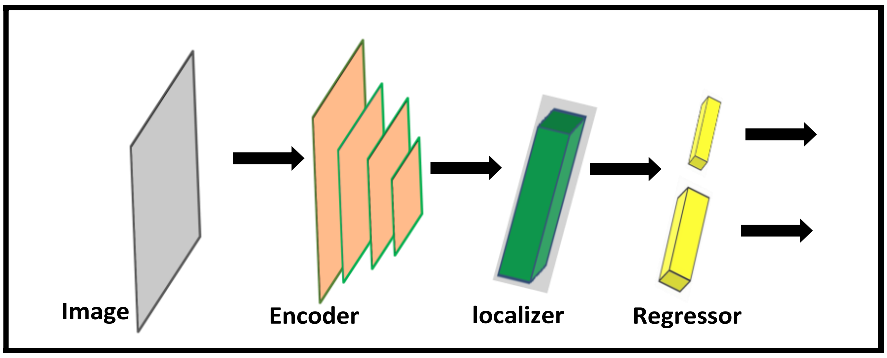


Figure 5.8. Pose net architecture

Input images can also be sourced from a webcam or CCTV footage. The advantage of Open Pose is simultaneous detection of body, facial and limb key point (Martinez et al., 2019). Figure 5.7a shows VGG-19, a trained CNN architecture from Visual Geometry Group. It is used to classify images using deep learning. It has 16 convolutional layers along with 3 fully connected layers, altogether making 19 layers and so-called VGG-19. The image extract of VGG-19 is fed to a “two-branch multi-stage CNN” as shown in Figure 5.7b. The top part of Figure 5.7c predicts the position of the body parts, the bottom part represents the prediction of affinity fields, i.e. the degree of association between different body parts. By these means, the human skeletons are evaluated in the image. The disadvantage of Open pose is that in crowded pictures having overlapping human beings, the annotation of one image may merge with the other failing in multi-individual key point construction.

5.4.1c POSENET

Posenet is an architecture that can estimate single or multiple poses by using different versions of algorithm. It is used for estimation of one individual or multiple people in an image (Clark et al., 2019a). However, there could be some variations in single and multi-pose estimation. Single pose estimation is faster and fits only one individual in the frame, and is less complicated compared to multi person estimation. PoseNet version can also take video inputs for pose estimation; it is invariant to image size, hence it gives a correct estimation even if the image is expanded or contracted (Luvizon et al., 2021). The posenet version can also take a processed image by digital camera as the input and gives the output as a set of key-points.

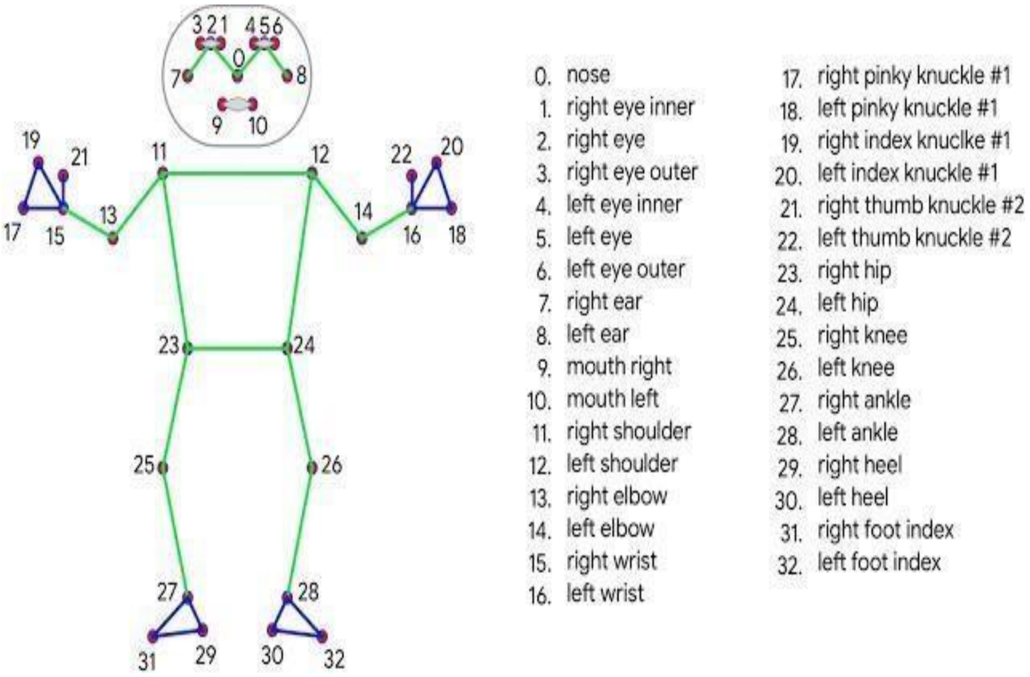
The architecture of posenet is shown in Figure 5.8, it has several layers with each layer having multiple units.

The first layer includes input images to be analysed, the architecture consists of Encoders that generate visual vectors from the image. These are then mapped onto a localization feature vector. Finally, two separated regressor layers give the estimated pose. The key points extracted using posenet are listed with part identity having a confidence score ranging from 0.0 to 1.0 which determines the overall unique points needed for pose estimation. The probability of the key points situated nearby is given by these confidence scores. The drawback is that the processing time taken to perform the activity is more compared to other architectures studies in this work. For this reason, it could only be used for qualitative analysis and maybe for action recognition but not as a scoring module. Example if some task has time involved then posenet is not appropriate.

5.4.1d MEDIAPIPE

MediaPipe is an open-source framework developed by Google used for machine learning solutions. It uses built-in packages for customizable solutions and also has tools for users to build their own solutions (Rinalduzzi et al., 2021). The development of algorithms and techniques work on mocap (mobile capture) data, an interpretation in the 3D global world used for validation and control.

MediaPipe has the best library for pose estimation (Clark et al., 2019b) and extracts body key points including foot, joining bones, and face using a single camera. It plays an important role in deploying neural networks to mobile devices and on to the desktop. MediaPipe can extract 33 key point from the human body (Singh, D., Panthri, S., & Venkateshwari, 2022) as shown in Figure 5.9 Extraction of all features becomes important based on the application. For example, in real-time applications pose detection and tracking must be very fast.



0. nose
1. right eye inner
2. right eye
3. right eye outer
4. left eye inner
5. left eye
6. left eye outer
7. right ear
8. left ear
9. mouth right
10. mouth left
11. right shoulder
12. left shoulder
13. right elbow
14. left elbow
15. right wrist
16. left wrist
17. right pinky knuckle #1
18. left pinky knuckle #1
19. right index knuclke #1
20. left index knuckle #1
21. right thumb knuckle #2
22. left thumb knuckle #2
23. right hip
24. left hip
25. right knee
26. left knee
27. right ankle
28. left ankle
29. right heel
30. left heel
31. right foot index
32. left foot index

Figure 5.9: Showing 33 key points detected by MediaPipe.

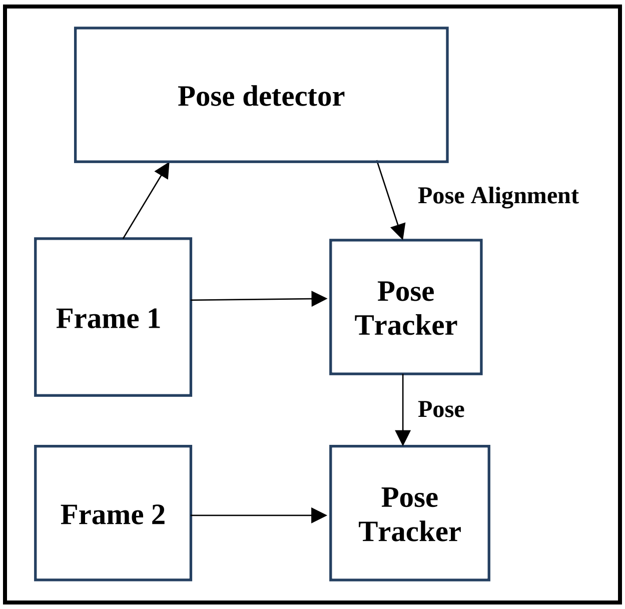


Figure 5.10: Human pose estimation pipeline overview

MediaPipe is used in smart recognition, dance and yoga that has several degree of freedom for appearances or outfits (Agarwal, V., Sharma, K., & Rajpoot, 2022) (Amrutha et al., 2021). In this word MediaPipe architecture is used for pose estimation as it can extract key points accurately from images taken from single camera. The processing speed is faster as it uses GPU acceleration.

We evaluated the Human Body Pose Estimation systems and to summarise our findings about MediaPipe on the following parameters. MediaPipe uses multi-platform TensorFlow Lite, multi- platform renderers, and already-made neural networks with any platform deployment. In media pipe it is also possible to mask out the area where they do not want the camera to see. The limitation of MediaPipe is that it cannot detect key point of the neck. This is an architecture for reliable pose estimation. It takes a colour image and pinpoints 33 key points on the image. The architecture is shown in Figure 5.10.

MediaPipe is a two-step detector-tracker ML pipeline used for pose estimation (Saini, 2021). Using a detector, this pipeline first locates the pose region-of-interest (ROI) within the frame. The tracker subsequently predicts all 33 pose key points from this ROI (Pauzi et al., 2021).

5.4.2 Data analysis of Pose Correction

The method of the complete posture correction system is as shown in Figure 5.11. 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. A rule-based algorithm 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 was shown.

A step-by-step description of data analysis and its implementation of the proposed Artificial Intelligent System

- Obtaining the dataset of 6000 images and dividing them into testing (20%) and training (80%) datasets.
- Classifying the images for 5 yoga poses and labeling them
- Pre-training a deep learning model using a Google teachable machine with 100 epochs, 32 batch sizes, and a learning rate of 0.001.
- Media pipe was used to extract key points and, in this work, the image is divided into four quadrants and the key points lying within the divided quadrants were compared with standard key points extracted from the reference images.
- Test image is captured using a camera and then given as input to the pre-trained model to detect all the key points.
- The key points detected from the pre-trained model give a skeletal view of the pose.
- In the correction model, the slope formula and tan formula is used to find the angles between the key points.
- The angles from the key points of the test image are compared with the reference image
- 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.

- Pose correction is performed by voice and text communication. This method of key extraction along with Google text-to-speech and speech-to-text was used for assistance.

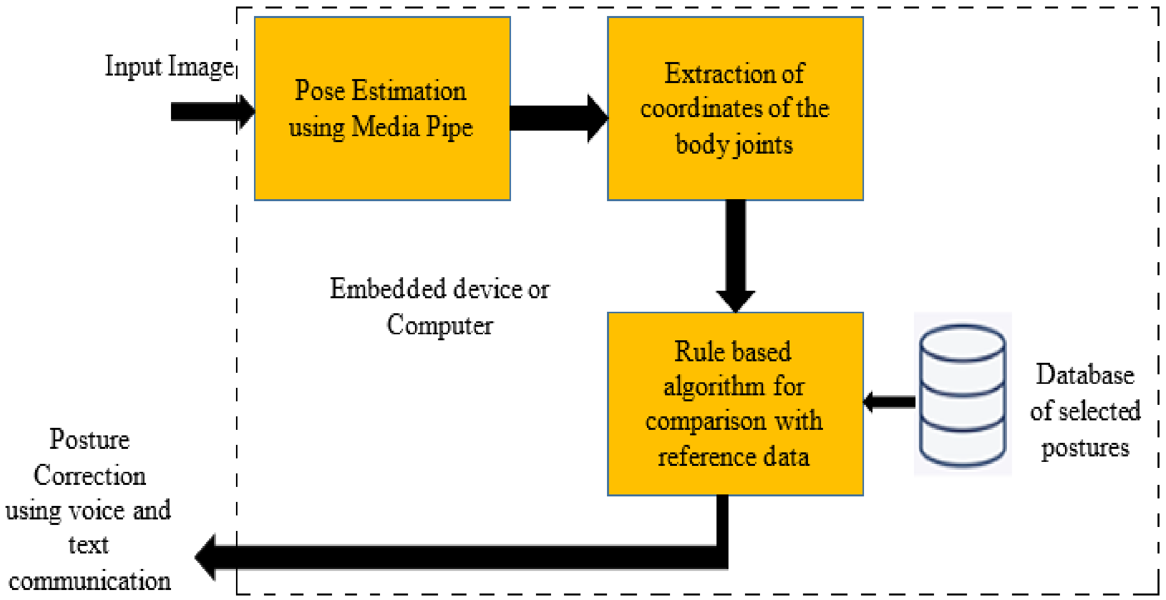

Input Image
Pose Estimation using Media Pipe
Extraction of coordinates of the body joints
Embedded device or Computer
Rule based algorithm for comparison with reference data
Database of selected postures
Posture Correction using voice and text communication

Figure 5.11: Block Diagram of Complete posture correction system