

Yoga Pose Detection and Realtime Feedback System

Subject: Foundations of Data Science (CS 711)

Instructor: Dr. Timothy D. Oleskiw

Presented By:

Maulik Raval (Student ID: 200531711)

Vinit Rathod (Student ID: 200528537)

Wai Phu Paing (Student ID: 200530777)

2025, April 10



University
of Regina

Go far, together.

Motivation



Executive Summary



Image Preprocessing

Pose Estimation

Key point Extraction

Random Forest Classifier

Pose Identification

Angle Calculation

Deviation Analysis

Image Preprocessing

Image Resize

Image
Renaming

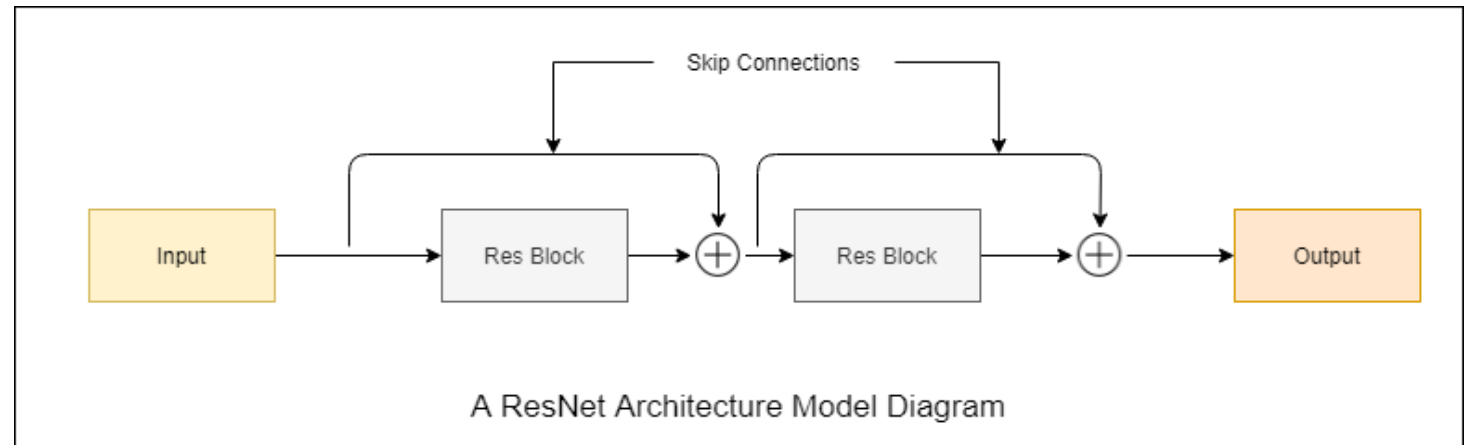
Image
Augmentation

Data
Exploration

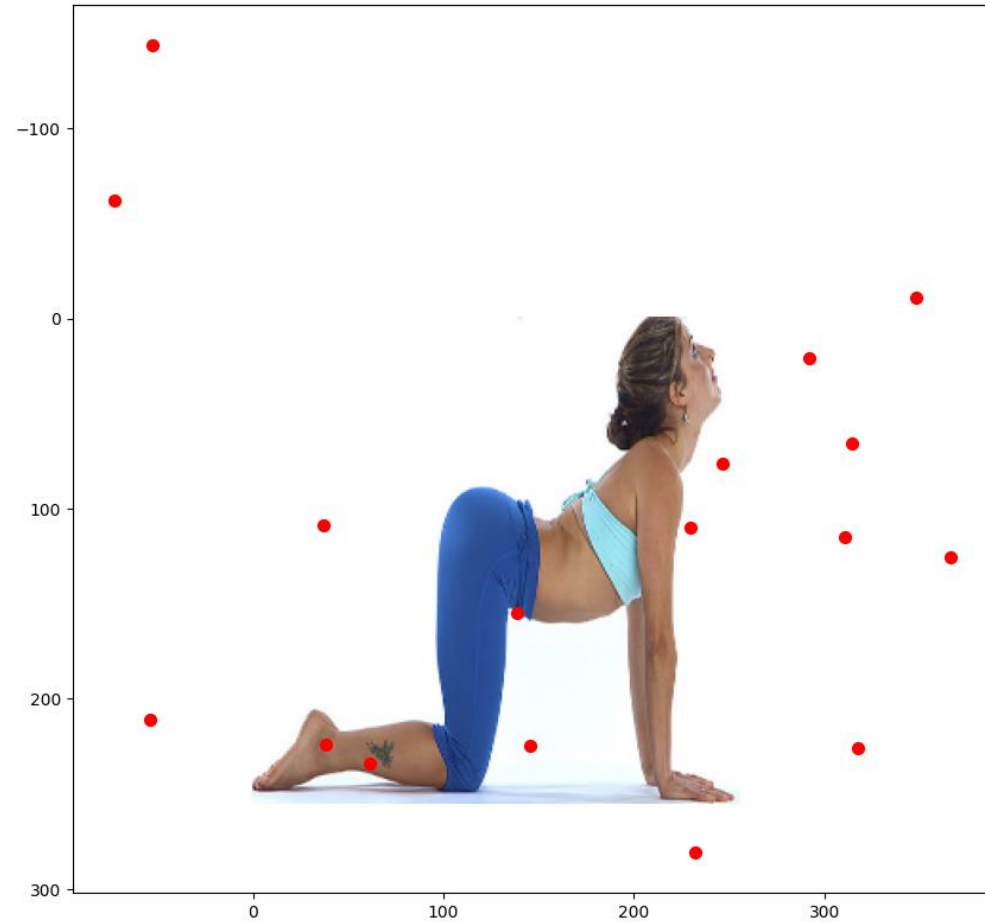


ResNet

- **age Classification** (e.g., ImageNet)
- **Object Detection** (e.g., Faster R-CNN, RetinaNet)
- **Semantic Segmentation**
- **Feature Extraction** (as a backbone network)



Key Points via ResNet



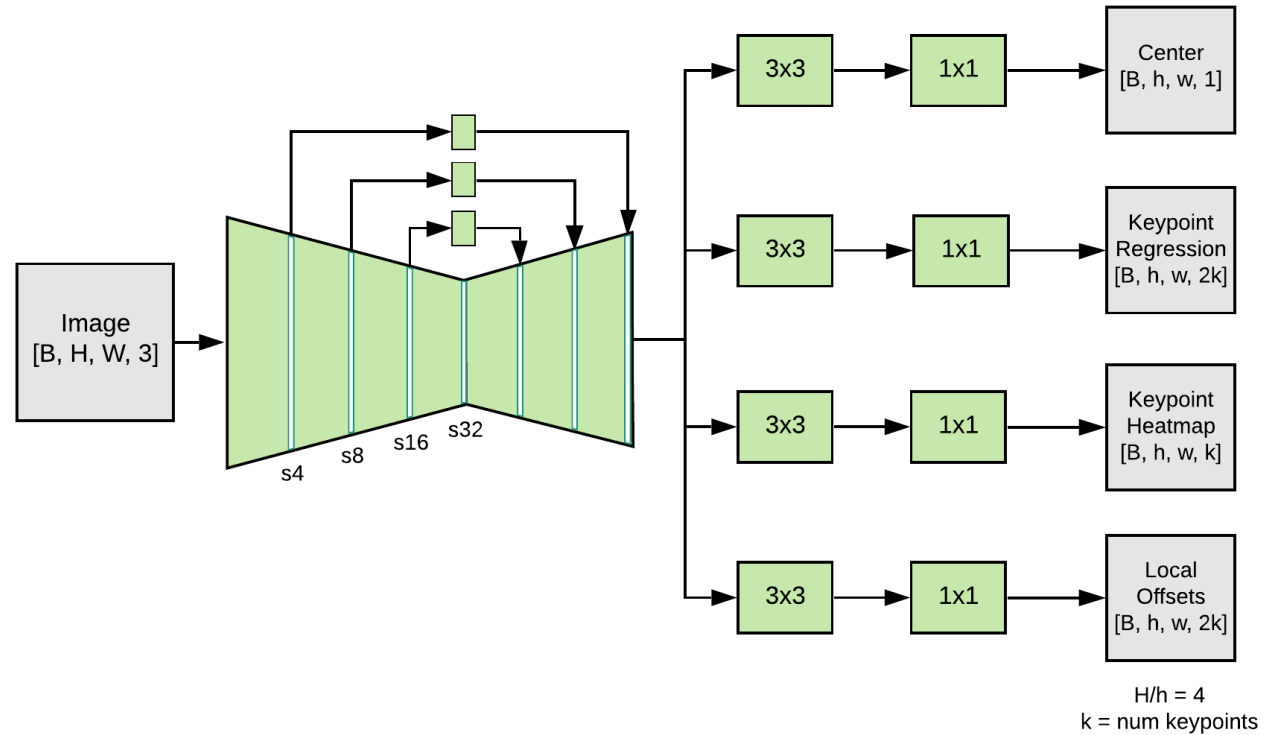
ResNet Accuracy : 19%

Why not suitable for Yoga Pose Detection ?

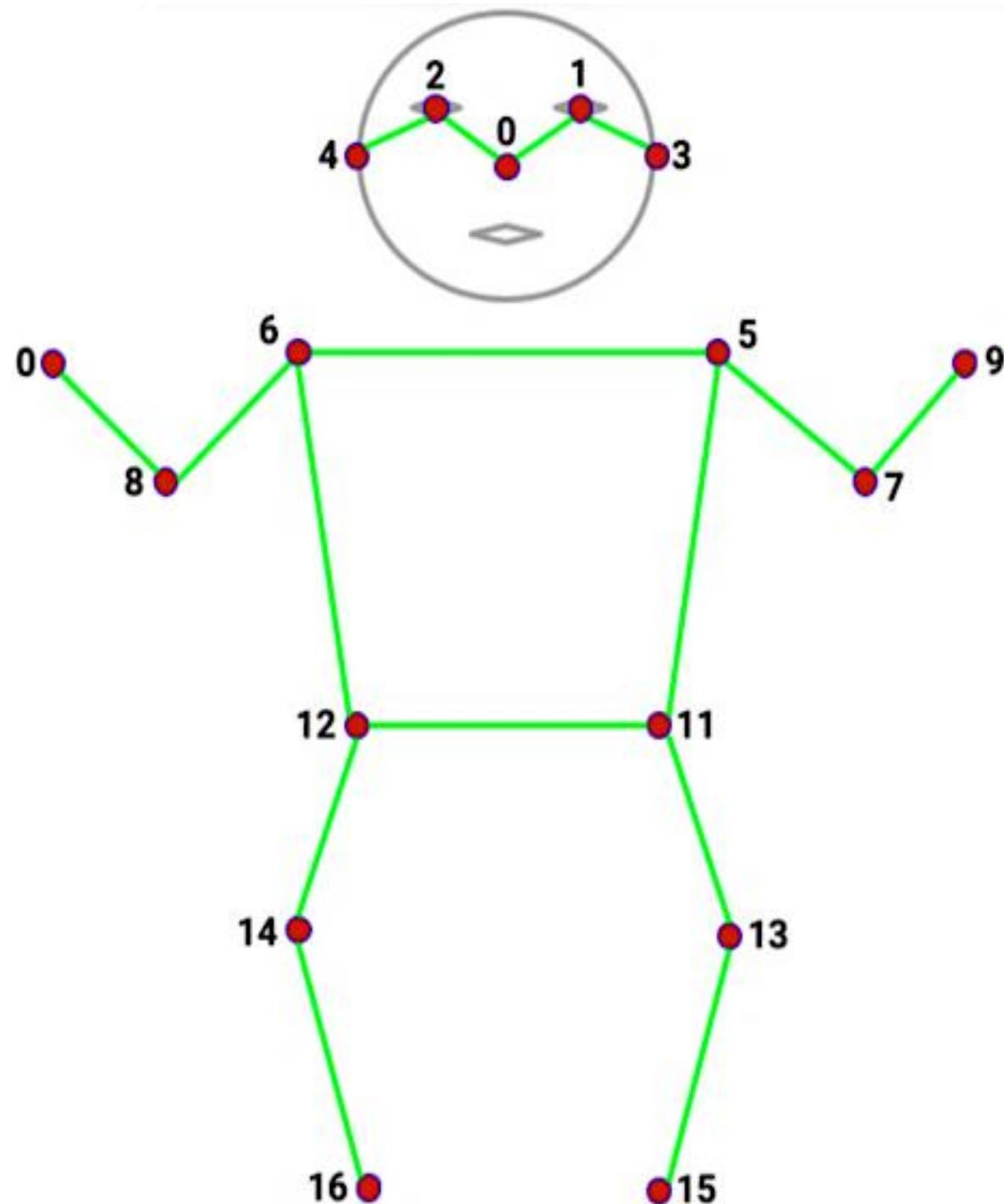
- **Lack of Spatial Awareness**
- **Not Optimized for Pose Estimation**
- **Limited Handling of Occlusions & Complex Poses**

MoveNet

- High Speed & Real-Time Performance
- Lightweight Yet Accurate
- Optimized for Full-Body Keypoints
- Robust to Challenging Poses



- MoveNet focuses on just 17 key body points
 - ❖ ears, eyes, and nose
 - ❖ shoulders, elbows, wrists
 - ❖ ankles, knees, hips
- These key points are associated with (x, y) coordinates.



Model Training

15 Yoga Poses

Cat-Cow Stretch Pose

Cow Pose

Crane Pose

Eight Limbed Pose

Frog Pose

Half Frog Pose

Half Lord of the Fishes Pose

Half Moon Pose

Handstand Pose

Happy Baby Pose

Low Lunge Pose

Shoulder Pressing Pose

Sleeping Vishnu Pose

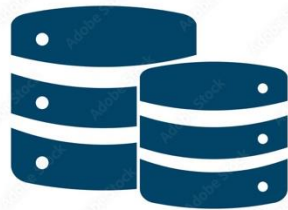
Upward Facing Dog

Wild Thing Pose



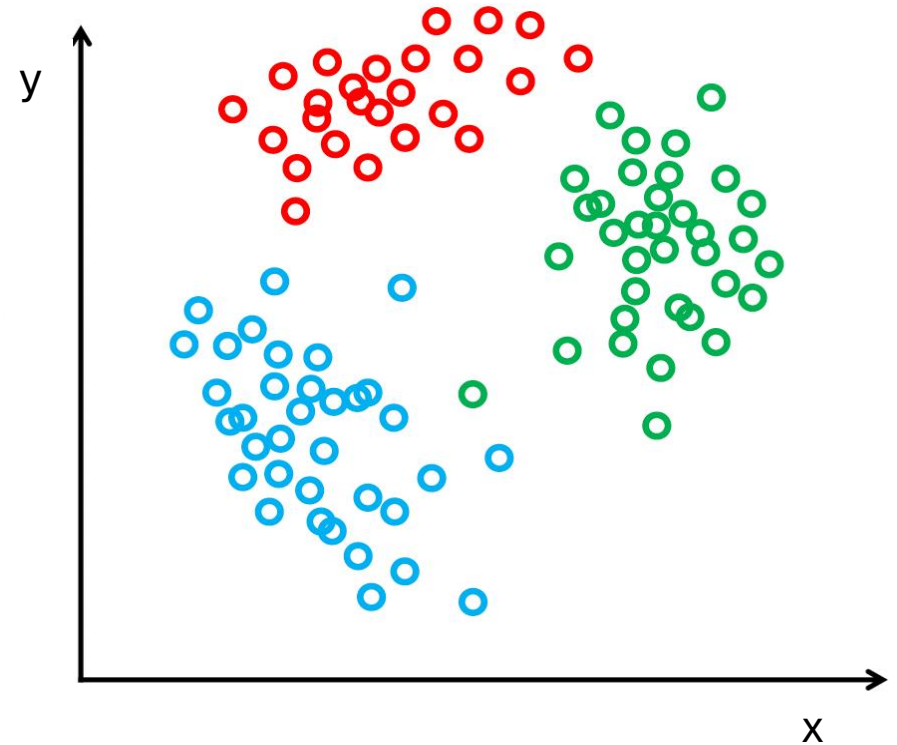
839 Images in Total

MoveNet



DATASET

839 * 17 Keypoint NumPy Array



Pic Ref: Dr. Timothy D. Oleskiw 's Decision trees, random forests Lecture slide

Apply Random Forest Classifier

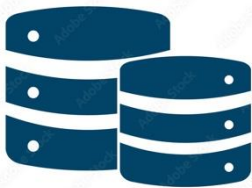
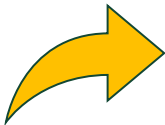
Getting Key Point Data of Image

Image size: 192x192



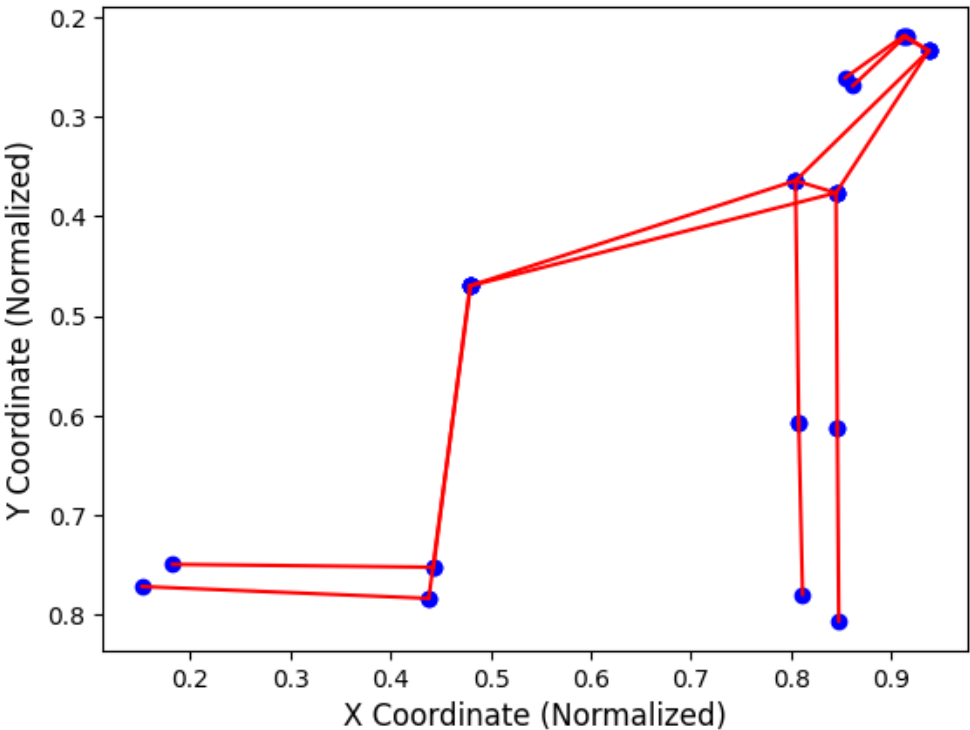
Keypoint NumPy Array

```
[[[0.23341215 0.9376577 0.71097064]
 [0.21885228 0.91211146 0.6244072 ]
 [0.21878207 0.91467065 0.6367774 ]
 [0.26059568 0.854275 0.8506904 ]
 [0.26864886 0.8618912 0.8434478 ]
 [0.36407575 0.80455256 0.64511234]
 [0.3763264 0.8450555 0.93105304]
 [0.60764235 0.8078069 0.6253908 ]
 [0.6120035 0.8463502 0.73567337]
 [0.7798307 0.811634 0.7194369 ]
 [0.8067042 0.8478359 0.69438255]
 [0.46964628 0.47938144 0.5592968 ]
 [0.46973348 0.47990435 0.8267148 ]
 [0.75257874 0.4428897 0.77903104]
 [0.7839204 0.43813977 0.60901004]
 [0.7496841 0.18239877 0.6904856 ]
 [0.7719881 0.15183108 0.76919746]]]]
```

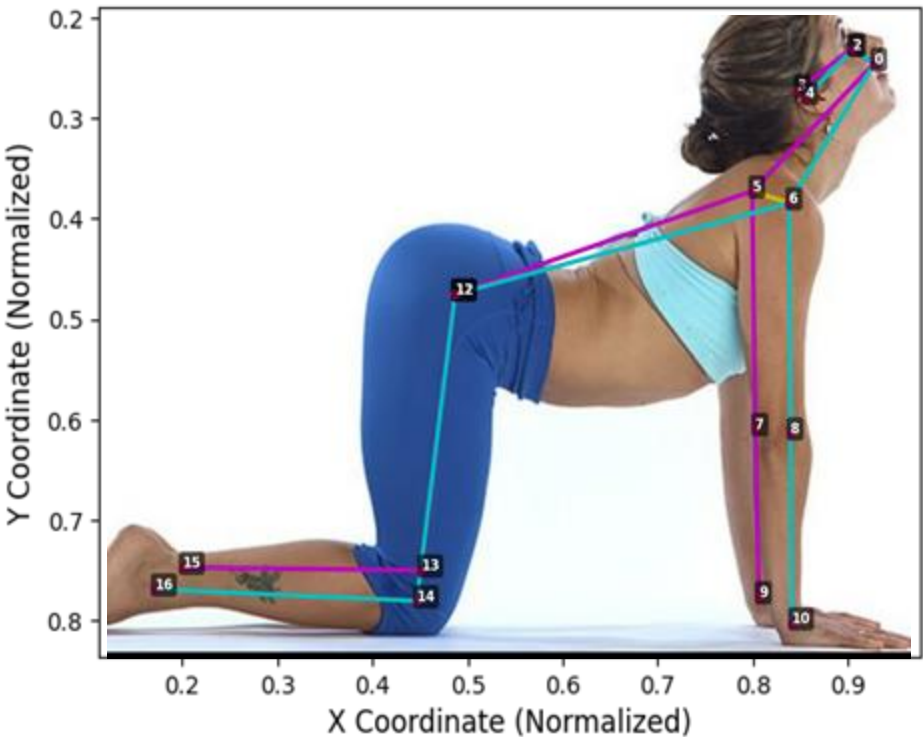


DATASET

Pose Skeleton



Pose Skeleton

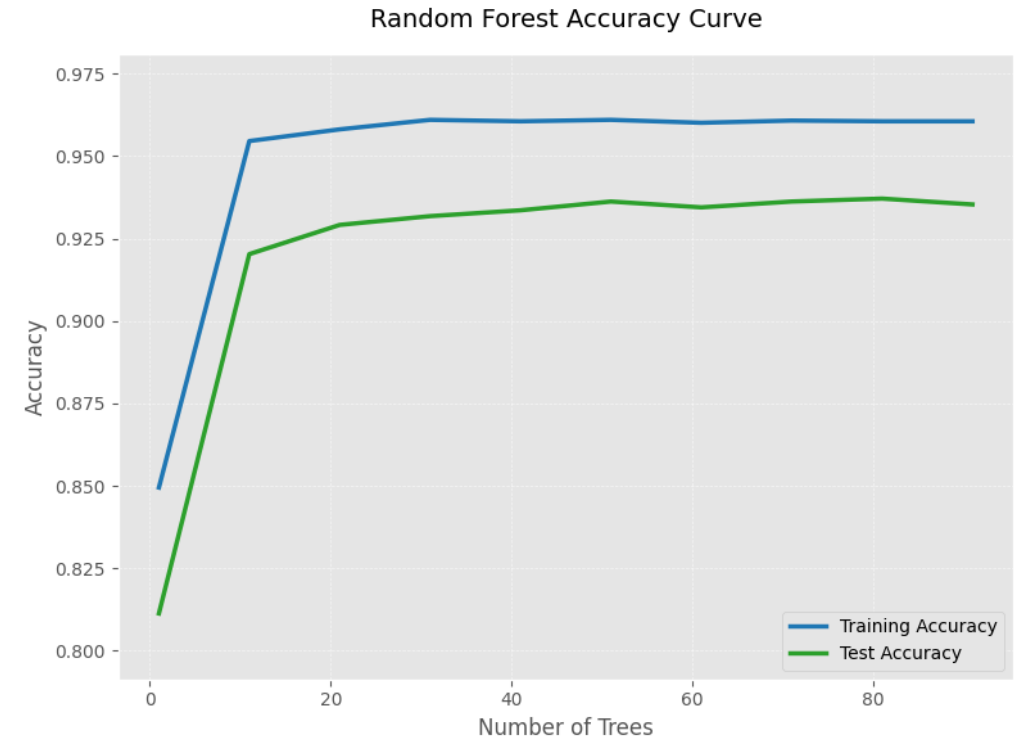
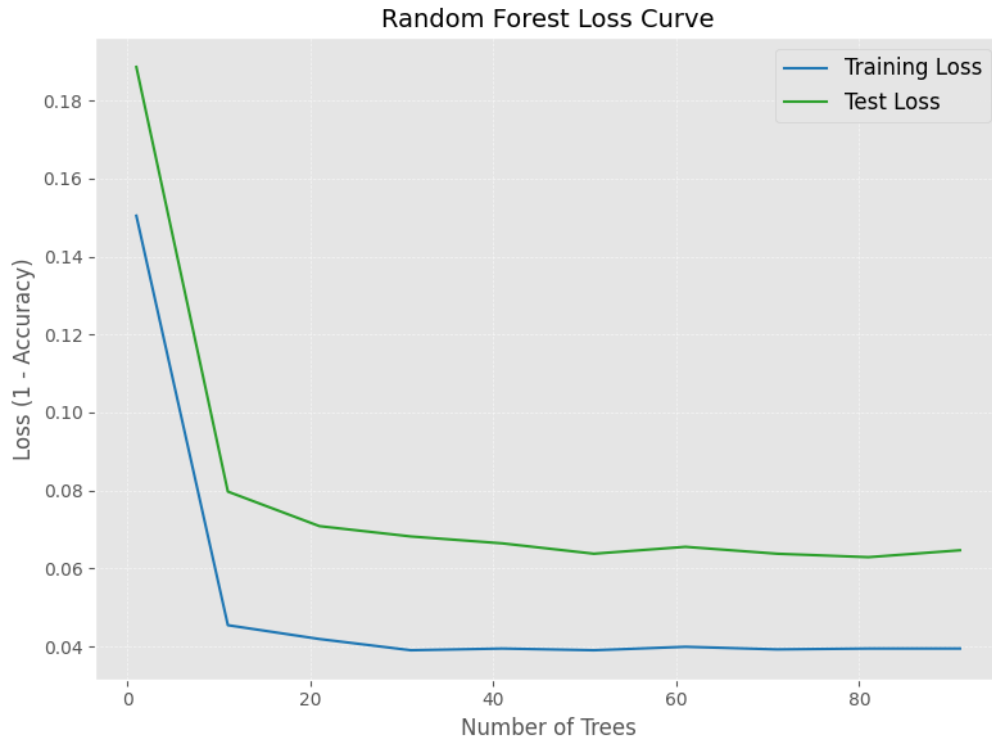


Key Point
Estimation
Accuracy : 75.1%

Model Accuracy (Random Forest)

Train Accuracy: 0.9605

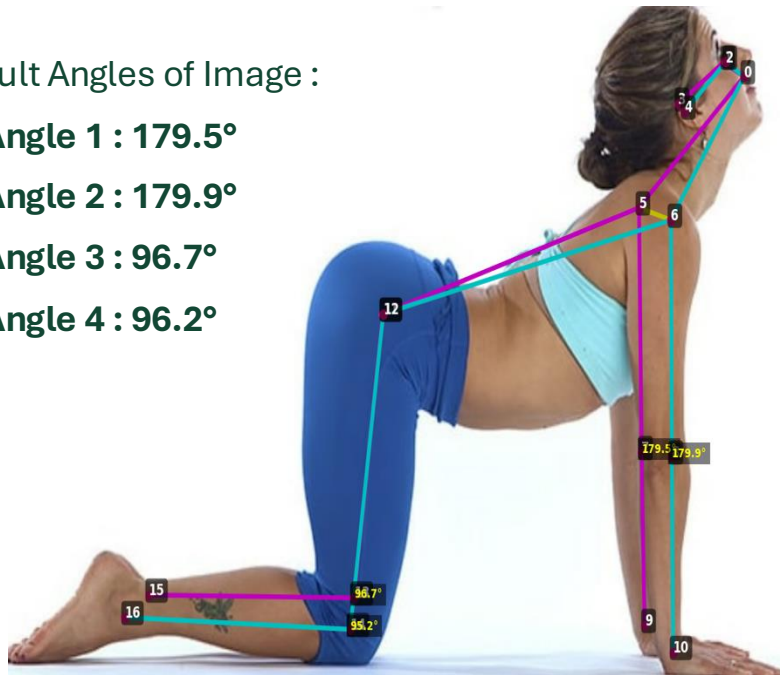
Test Accuracy: 0.9353



Calculation Angles

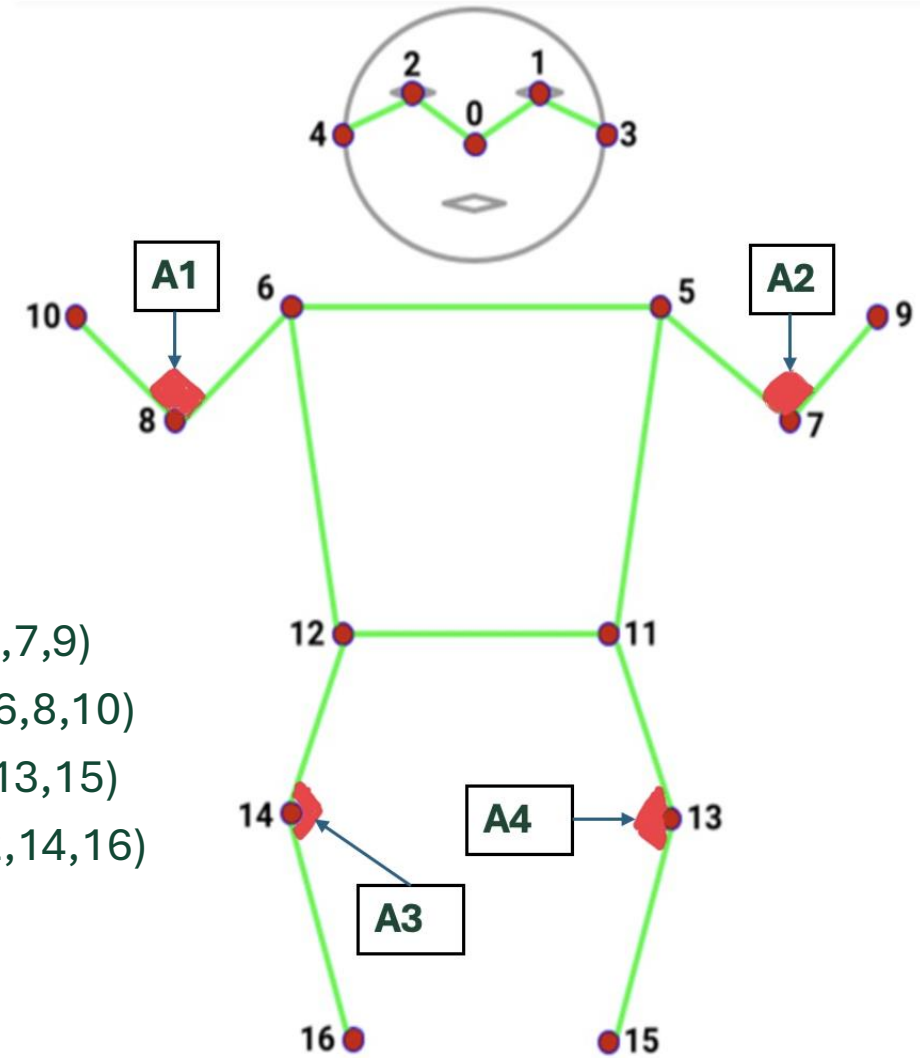
Result Angles of Image :

- Angle 1 : 179.5°
- Angle 2 : 179.9°
- Angle 3 : 96.7°
- Angle 4 : 96.2°

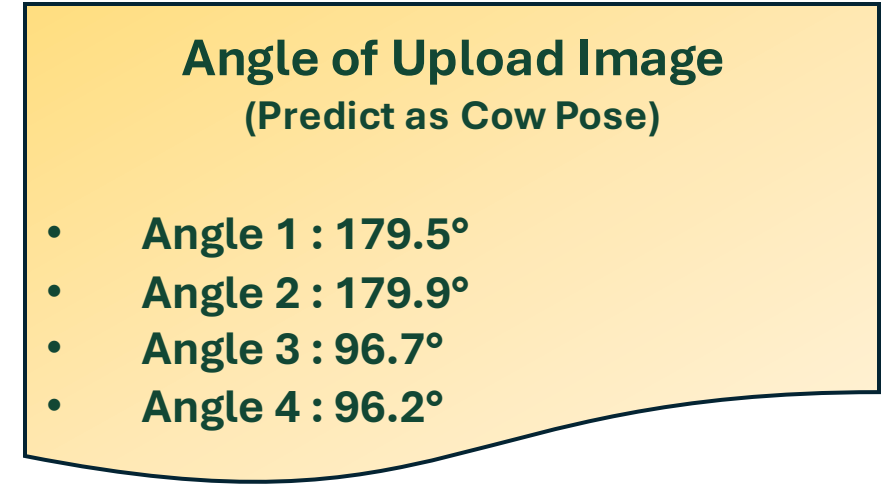
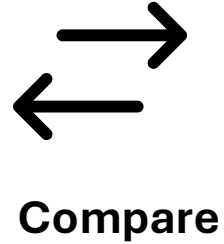
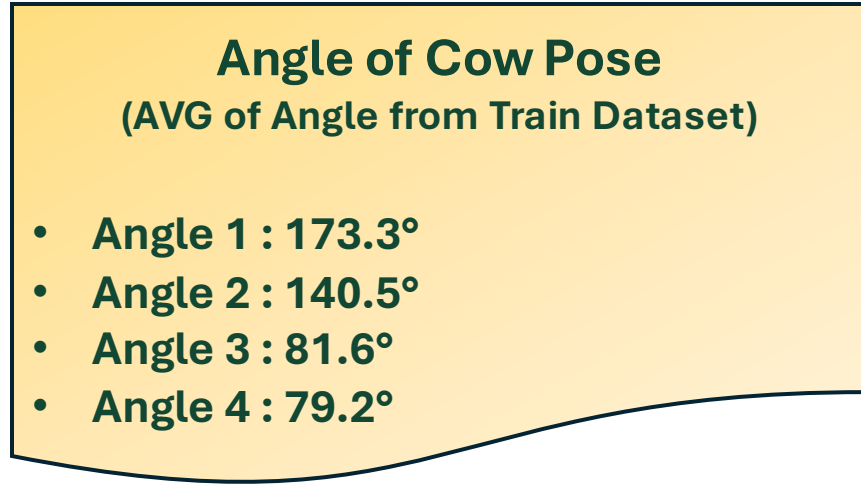


Critical Points

- Angle 1 : Left Hand (5,7,9)
- Angle 2 : Right Hand (6,8,10)
- Angle 3 : Left Leg (11,13,15)
- Angle 4 : Right Leg (12,14,16)



Calculation Angles (Example : Cow Pose)



- **Calculation :**
Result tolerance = Abs (User's Angle – Ideal Angle)
- **Validation :**
Check if '**Result tolerance**' is within the tolerance level.

Tolerance Levels

Beginner : More forgiving ($\pm 50^\circ$)

Intermediate : Moderate strictness ($\pm 15^\circ$)

Advanced : Strictest ($\pm 10^\circ$)

Web Application Overview



- Purpose of the Web App
 - To provide an easy interface to user for uploading of yoga images.
 - Calculate angles, give instant feedback and Detect the pose.
 - Visualize Results, highlight deviations, and suggest improvements.

- Technology Stack

Layer	Tech Used
Frontend	HTML, CSS, Bootstrap 5.3
Backend	Flask (Python)
Model	Movenet + Randon Forest
Integration	Javascript (Fetch API)
File Handling	NumPy, OpenCV

Web Application Overview

- User Workflow
 - User Uploads an Image + selects difficulty Level
 - Flask Sends the image to the Movenet Model
 - Keypoints are extracted and angles are calculated
 - Pose is classified and results are displayed
 - Detected Pose
 - Key Angles & their deviations
 - Pass/fail feedback based on tolerance





THANK YOU
FOR YOUR
ATTENTION!



ANY
QUESTION