# Form Correctness Detection Using Pose Estimation

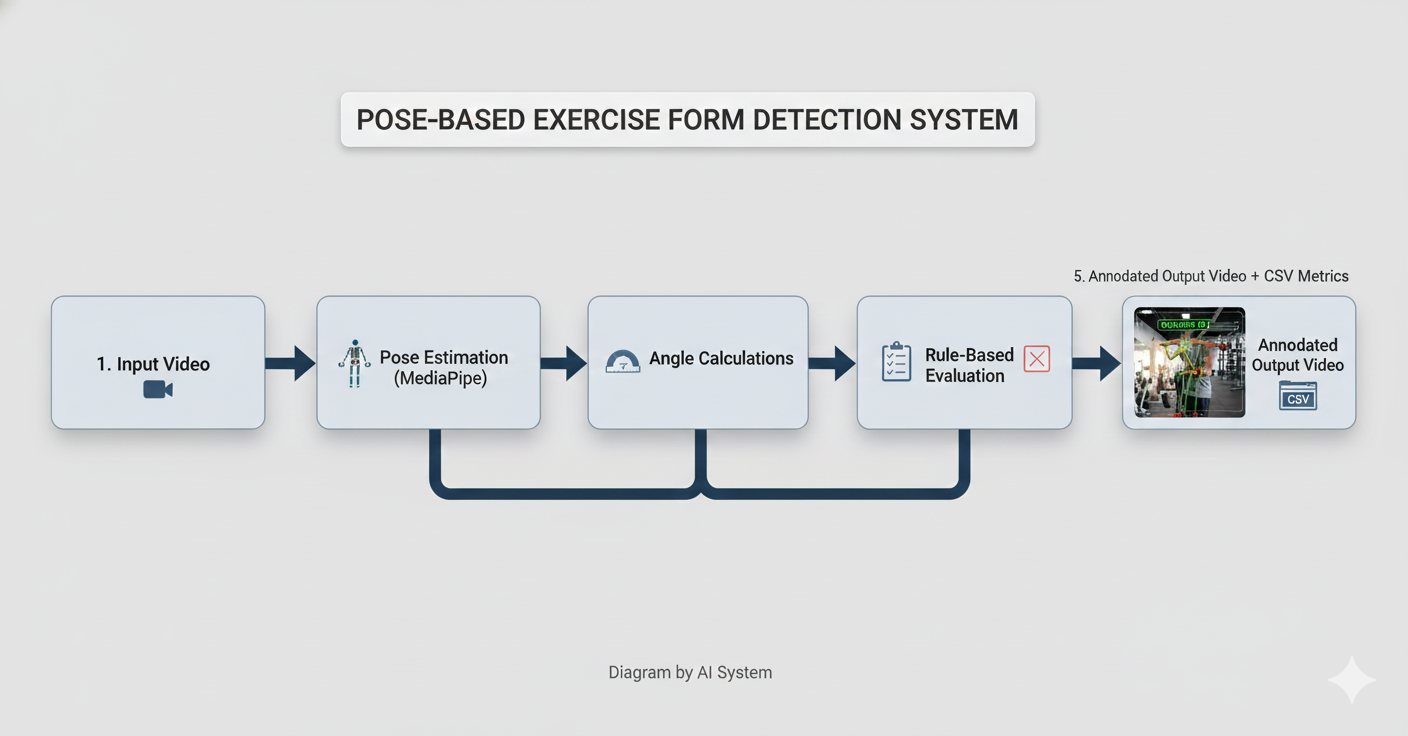
[Github Link:](https://github.com/Yogeshkharb111/Form-Correctness-Detection-Using-Pose-Estimation) <https://github.com/Yogeshkharb111/Form-Correctness-Detection-Using-Pose-Estimation>

## 1. Introduction

This report presents an exercise form correctness detection system developed using MediaPipe Pose Estimation and rule-based geometric analysis. The system evaluates three exercises: Bicep Curl, Lateral Raise, and Squat. It detects body keypoints from video, computes joint angles, applies form rules, and generates annotated videos and CSV metrics. The project was executed in Google Colab due to local system incompatibilities with MediaPipe and Python versions.

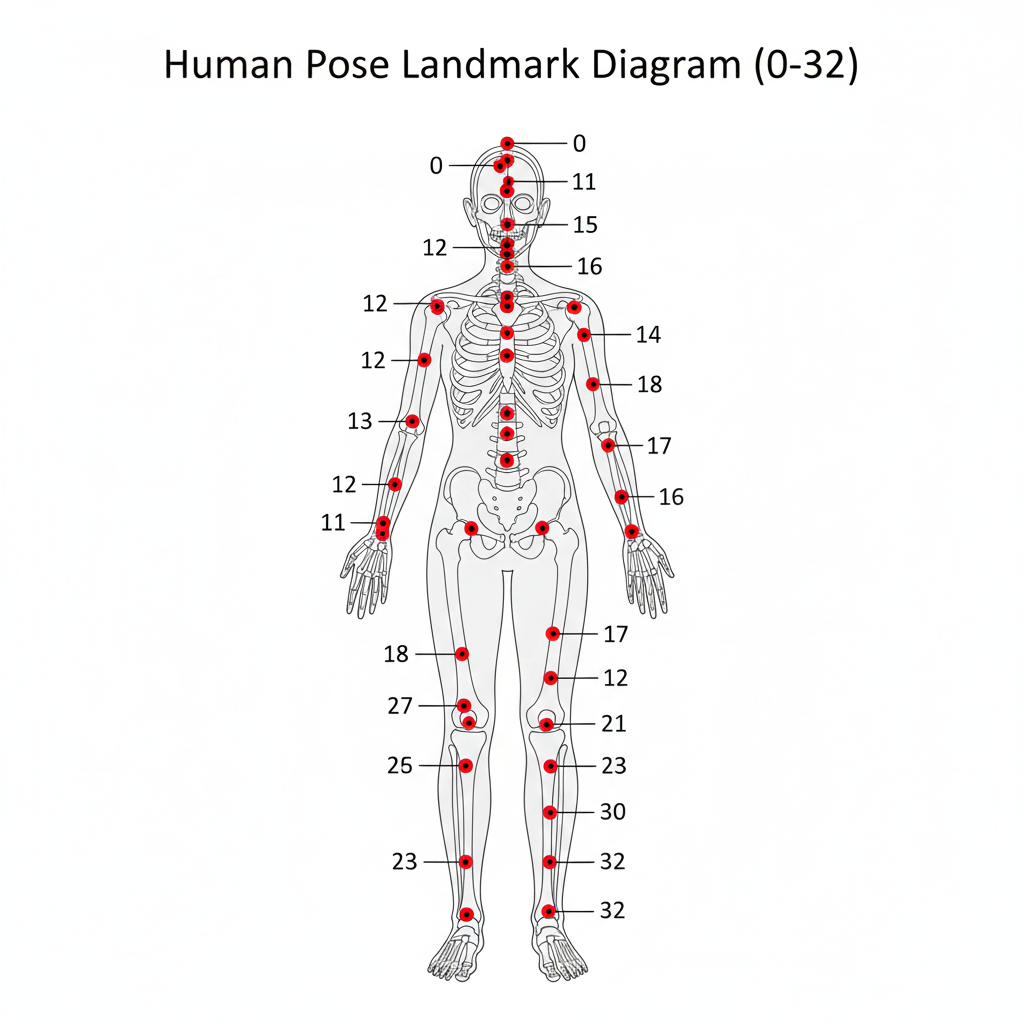
## 2. System Pipeline

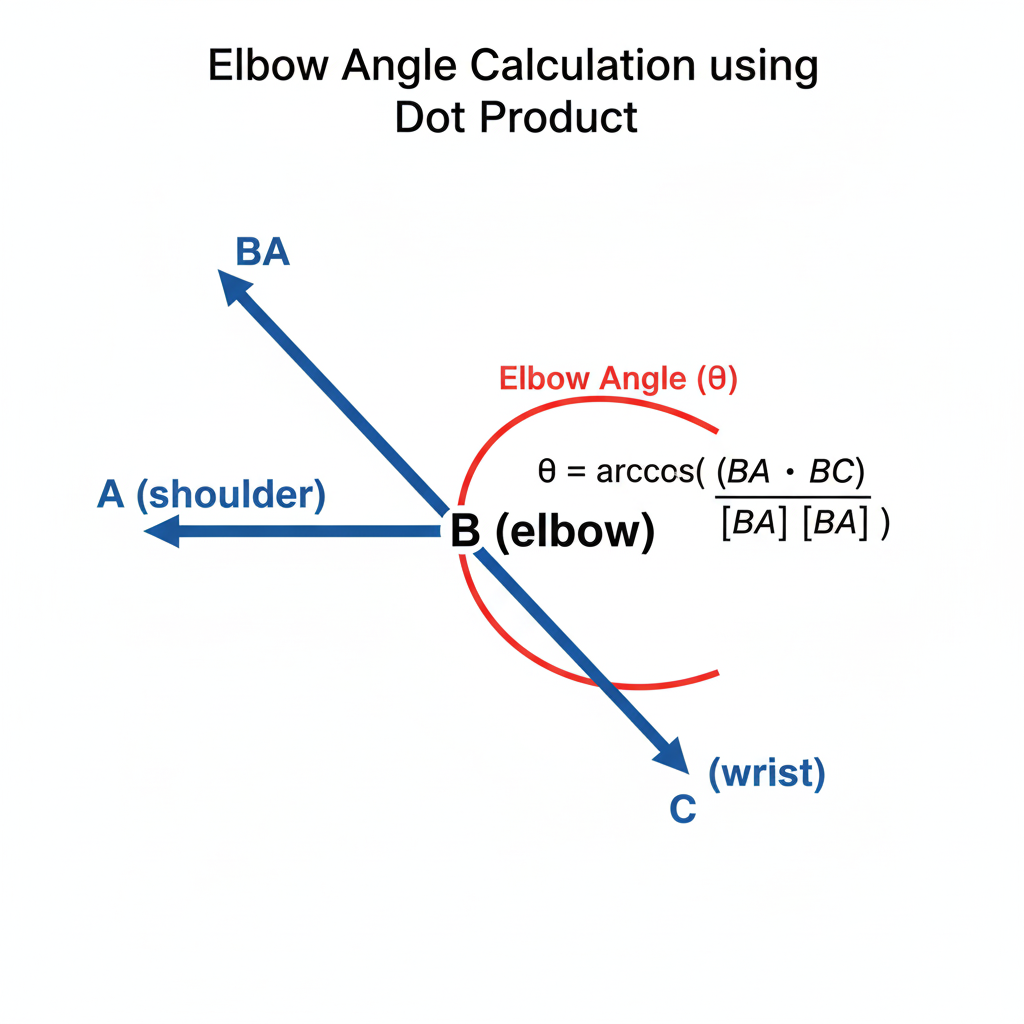
1. Input video frames are processed using MediaPipe Pose.  
2. Landmarks are extracted for all 33 body keypoints.  
3. Angles are computed using the dot-product-based vector angle formula.  
4. A moving-average smoothing filter removes jitter.  
5. Rule-based logic evaluates exercise correctness.  
6. Annotated frames and CSV metrics are generated as output.



## 3. Mathematical Foundations

Angle Calculation using Dot Product:  
cos(θ) = ((A−B) • (C−B)) / (||A−B|| × ||C−B||)  
θ = arccos(clamp(cos(θ), −1, 1))  
This method is used for elbow angle, knee angle, and shoulder–elbow–wrist alignment.

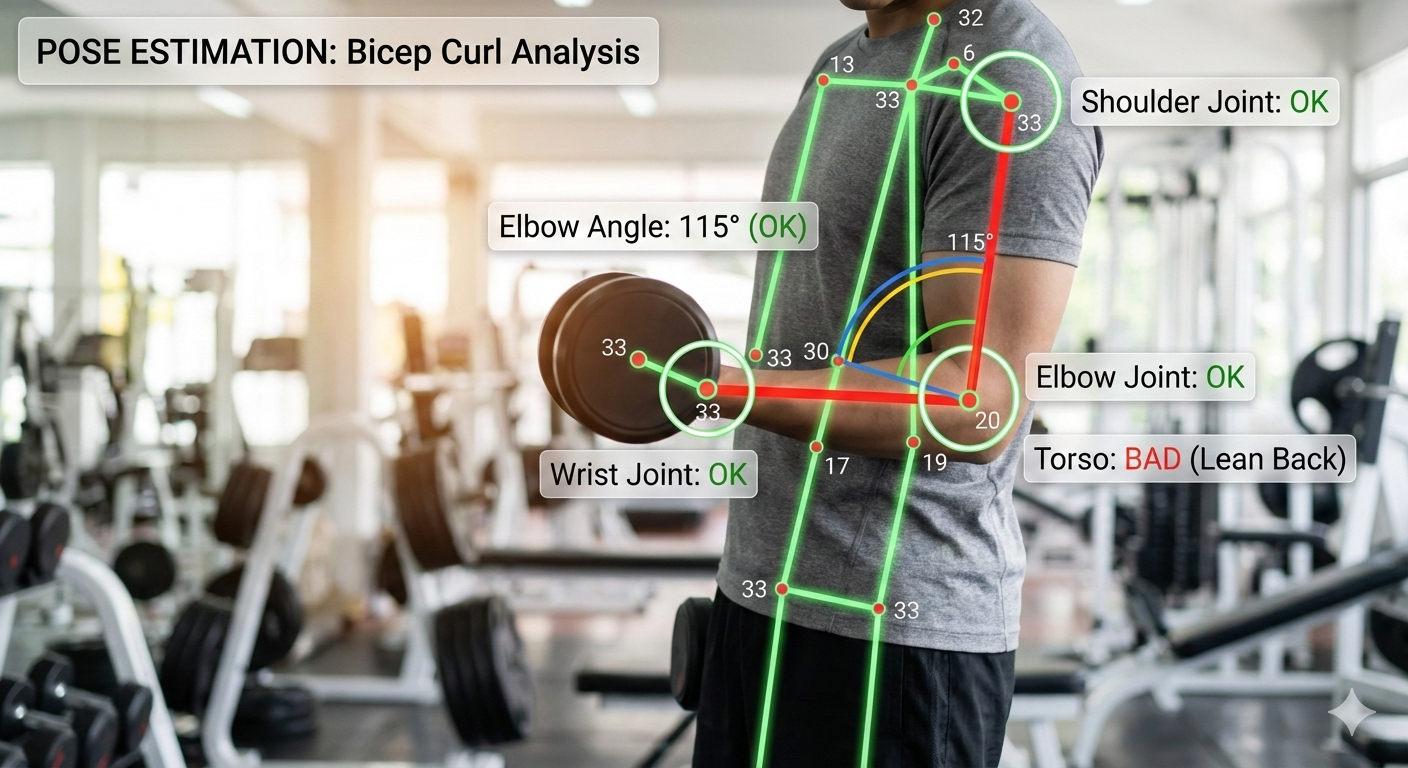




## 4. Posture Rules Implemented

### Bicep Curl

• Elbow angle must remain between 30° and 170° during motion.  
• Shoulder remains stable.  
• Wrist aligned below the elbow.



### Lateral Raise

• Wrist-to-shoulder vertical distance must be small.  
• Elbow angle ≥ 150° for proper raise height.  
• Symmetry required between both arms.



### Squat

• Knee angle ≤ 100° for adequate depth.  
• Torso tilt ≤ 25° to maintain upright posture.  
• Knee-over-ankle displacement within ±0.12 normalized units.



## 5. Handling Multiple Persons

The system selects the person with the largest bounding box and evaluates only that individual to ensure consistent single-subject analysis.

## 6. MLflow Experiment Tracking

As an additional enhancement, **MLflow** was integrated to track experiment metadata and results.  
For selected exercise runs, MLflow logs:

* Input parameters such as video name, FPS, and frame count
* Statistical joint-angle metrics (mean, minimum, and maximum values)
* Output artifacts including CSV metric files and annotated sample frames

Experiment tracking is performed using a **local SQLite backend**, which is suitable for execution in Google Colab environments. Due to Colab networking restrictions, the MLflow UI was not externally exposed; however, all experiment data is logged locally to ensure reproducibility and structured evaluation.

## 6. Challenges Faced

• MediaPipe jitter and inconsistent landmark detection.  
• Angle fluctuations requiring smoothing.  
• Multi-person interference in videos.  
• Local system incompatibility with MediaPipe, requiring Google Colab.  
• Video writing and frame annotation complexities.

## 7. Conclusion

The developed system successfully detects body posture, computes key exercise metrics, and evaluates form correctness for bicep curls, lateral raises, and squats. Future improvements may include rep counting, ML-based scoring, support for additional exercises, and deployment on mobile or web applications.

Prepared by: Yogesh Kharb, Reg: 12219441, Email: yogeshkharb111@gmail.com