Biceps Curl Correction

Description

This project is a computer vision–based application that determines whether a biceps curl exercise is performed correctly or incorrectly. Incorrect exercise posture can reduce training effectiveness and increase the risk of injury. This system addresses the problem of providing instant feedback during exercise, helping users maintain proper form and reduce the risk of musculoskeletal injuries. ## Objectives The project aims to provide automatic feedback on posture during biceps curl exercises and to count only the repetitions performed correctly. Technically, it focuses on detecting key body landmarks, extracting pose features, and classifying movements in real-time. ## Technologies Used - Python - OpenCV - Mediapipe - NumPy - Pandas - Scikit-Learn - SVM (Support Vector Machine) ## System Workflow 1. Dataset Collection Record videos of both correct and incorrect biceps curl postures, performed separately with the left and right hands. 2. Data Labeling The shoulder, elbow, and wrist landmark coordinates of both hands are recorded for each video frame. Each frame is labeled as correct or incorrect according to the video name. All labeled data are saved in a CSV file. 3. Preprocessing Convert landmark coordinates into angle-based and movement features. The dataset is then split into training (90%) and testing (10%) sets. Both datasets are saved in CSV format. 4. Model Training Train a random forest classifier using the training dataset and evaluate it on the testing dataset. The trained model is saved as a .joblib file. The model achieved an accuracy of 82%. 5. Inference Build an inference program for video-based or real-time predictions, with the workflow like Input video, Landmark extraction, Feature calculation, Pose classification, Repetition counting, Feedback. outputting Status (correct or incorrect), Total repetitions counted, Stage (up or down) ## Key Features - Pose classification for biceps curl (correct / incorrect) - Counting correct repetitions based on validated poses ## Challenges & Issues - Limited dataset - Labeling each frame is time-consuming - Determining the right hyperparameters - Imbalanced dataset - Low model accuracy ## Solutions & Technical Decisions - Downsample videos with incorrect labels to balance the dataset - Use models that perform well on small datasets (e.g., SVM) - Apply trial-and-error to optimize hyperparameters and improve accuracy ## Results & Evaluation The implemented solutions successfully addressed the main issues. However, accuracy could be further improved with a larger and more diverse dataset.

Lessons Learned

- The importance of having sufficient data
- Handling sequence models and training procedures
- Automating data labeling
- Calculating angles from joint coordinates