Assignment 3 Report: Model Enhancements and Performance Evaluation

Title: AI-Driven Personalized Pain Management with Real-Time Feedback and Relief Suggestions

Team Name: ReliefXperts
Team Members:

Name	Role in Assignment-1	Github
Suryansh Patel	Leader, Research papers reviewer, decided to work on Video related datasets	https://github.com/suryansh-max
Jainil Patel	Data Sets collector decided to work on sensors and other pain-related data	https://github.com/jainilpatel98
Cameron O'Dell	Research Paper Reviewer decided to work on Video related datasets	https://github.com/cam-odell
Jacob Xayaphet	Data collector(other than videos) decided to work on sensors and other pain-related data	https://github.com/NoviceOfCode

Side Hustle Video-

See directly:
https://mailmissouri-my.sharepoint.com/personal/japmyy_umsystem_edu/_layouts/15/stream.aspx?id=%2Fpersonal%2Fjapmyy%5Fumsystem%5Fedu%2FDocuments%2FAttachments%2FMon%20Oct%2021%202024%2017%5F01%5F19%2Ewebm&ga=1&referrer=StreamWebApp%2EWeb&referrerScenario=AddressBarCopied%2Eview%2Ede8464a0%2Dccd3%2D433d%2Dac3d%2D5d4e59c90ffd

Also uploaded to github
https://github.com/DS-CAPSTONE/exercise_correction/blob/main/demo/Mon%20Oct%2021%202024%2016_52_36.webm

Tiktok Video Link -

https://drive.google.com/file/d/1OCqvVwISxxSY9F7_D2v-Vwe1R50booM7/view

Github Link:

https://github.com/DS-CAPSTONE/exercise_correction

Presentation Link:

https://docs.google.com/presentation/d/12DRaHMwvukkbMnKeS1ctr25xqHLYqesWLkw-rL_9cvg/edit?usp=sharing

Technical Report

Objective:

- The goal of this assignment was to build upon the Hands-on Session from 10/3 and implement enhancements in a machine learning model. This report provides an overview of the key improvements made to the model, focusing on accuracy, efficiency, and the application of advanced techniques.
- Develop an AI-powered tool that monitors exercise posture and provides real-time feedback to users while also managing and alleviating pain during workouts.

Key Features:

- Real-time posture correction
- AI-driven pain monitoring and relief suggestions
- Personalised pain management plans

Problem Statement

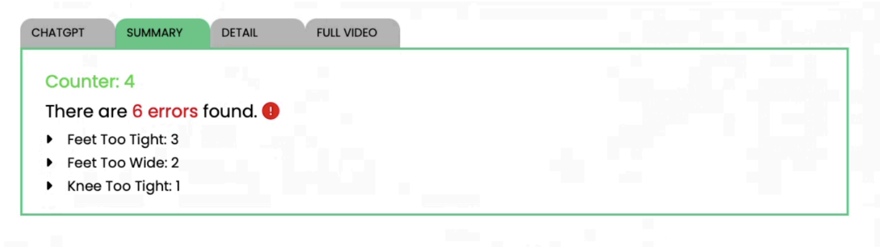
- Incorrect exercise posture can lead to injuries, especially among new gym-goers.
- Managing pain during workouts is crucial to avoid aggravating injuries and to ensure a safe exercise routine.
- There is a need for a tool that provides immediate, personalised feedback on both posture and pain during exercises.

Model Details:

- **Data Source:** The model was trained on video datasets specifically curated to represent various squatting techniques and potential posture issues. We included videos from multiple camera angles to ensure that the model could detect errors regardless of the user’s positioning.

- **Real-Time Optimization:** One of the core goals of the model was to ensure real-time performance, which required optimizing the model architecture for low latency. We focused on reducing the number of parameters without sacrificing accuracy, allowing the model to run efficiently even on resource-constrained devices.
- **Error Detection:** The model is trained to identify specific posture errors during squats, including improper foot placement, excessive forward lean, knee misalignment, and lack of hip engagement. Based on these detections, the system provides real-time feedback on how to correct the posture.

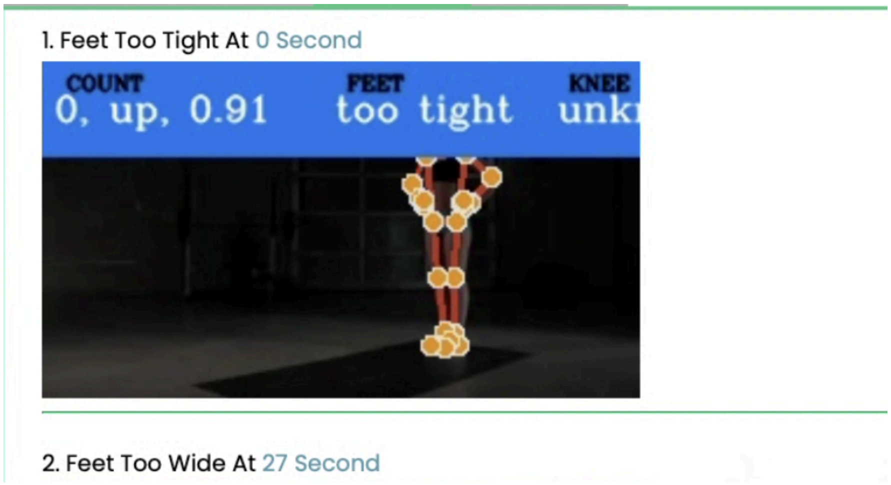
Results for squats:



Counter: 4

There are 6 errors found. !

- Feet Too Tight: 3
- Feet Too Wide: 2
- Knee Too Tight: 1


1. Feet Too Tight At 0 Second

2. Feet Too Wide At 27 Second

Triangle Model

Frontend:

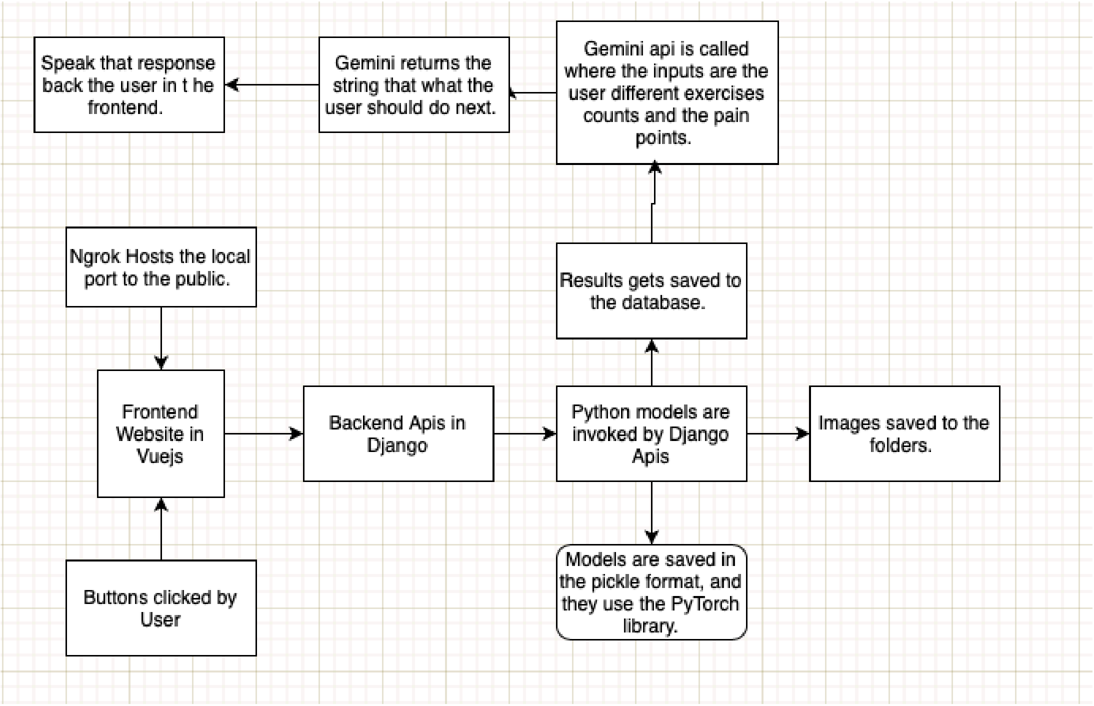
- We are using Vue.js for better maintainability. Currently, Vue.js is utilized for long-term support.
- Users can upload exercise videos or stream live workout sessions, and receive real-time feedback with visual outputs such as images, text annotations, and error counts.

Backend:

- Implemented using Django, the backend supports three separate "Apps" within the framework, each responsible for processing specific exercises. These apps handle real-time data input (both manual uploads and streaming) and generate feedback in the form of error reports, images, and suggestions for improvement.
- We plan to integrate ChatGPT API to provide tailored suggestions for users based on their performance.

Database:

- MongoDB was chosen as the primary database due to its ability to handle unstructured data and its efficiency in processing real-time streaming data. This enables us to efficiently store, retrieve, and analyze data related to both exercise performance and pain management.


Speak that response back the user in t he frontend.
Gemini returns the string that what the user should do next.
Gemini api is called where the inputs are the user different exercises counts and the pain points.
Ngrok Hosts the local port to the public.
Results gets saved to the database.
Frontend Website in Vuejs
Backend Apis in Django
Python models are invoked by Django Apis
Images saved to the folders.
Buttons clicked by User
Models are saved in the pickle format, and they use the PyTorch library.

Note - figure shows implementation of Triangle Model

Challenges & Solutions

- **Data Quality**: One of the main challenges was ensuring clean and accurate data for model training. The solution involved adding a preprocessing layer to our pipeline that filters and organizes the incoming data, ensuring that the model receives only well-structured data for training and analysis.
- **Frontend Transition**: As our team was more experienced in React.js but the project required us to adopt Vue.js, we faced challenges adapting to the new framework. Our solution was to maintain both frontends during the transition period, allowing team members to gradually shift their efforts to Vue.js while still supporting existing features in React.js.
- **Real-Time Data Processing**: Managing real-time posture feedback and error reporting in live streaming environments posed technical challenges. Leveraging MongoDB's capability to handle streaming data helped solve this problem, ensuring smooth, real-time data handling.

Report on Model Enhancements:

We utilized a deep learning approach to classify human movements based on keypoint coordinates captured from video frames. Our methodology included data preprocessing, model building with hyperparameter tuning, and performance evaluation. The implementation steps and results are detailed as follows:

1. Data Description and Preprocessing

- **Data**: The dataset consists of specific key landmarks of the human body such as the nose, shoulders, hips, knees, and ankles. These landmarks are represented by their x, y, z coordinates, along with a visibility score (v).
- **Features**:
  - We selected 13 important landmarks from the pose estimation process, and generated 52 features, representing the (x, y, z, visibility) coordinates for each landmark.
- **Labels**: The movements are categorized into two classes: **Lunge** and **Crouch**, represented as 0 and 1 respectively.
- **Data Preprocessing**:
  - **Normalization**: The features were scaled using a pre-trained scaler (input_scaler.pkl) to normalize the data.
  - **Categorical Encoding**: The labels were converted into categorical format using to_categorical from Keras, as required for multi-class classification.

2. Model Building and Hyperparameter Tuning

We experimented with different neural network architectures using the keras_tuner library to identify the optimal hyperparameters and model configurations. Four different models were created:

1. **Three-Layer Model**:
    - Consists of an input layer, one hidden layer with tunable number of units and activation functions, and an output layer.
    - **Best Hyperparameters**:
        - Activation: ReLU
        - Number of units: 256
        - Learning rate: 0.001
2. **Five-Layer Model**:
    - Extends the architecture to three hidden layers with varying units.
    - **Best Hyperparameters**:
        - Activation: Tanh
        - Number of units: 128, 256, 512
        - Learning rate: 0.001
3. **Five-Layer Model with Dropout**:
    - Similar to the five-layer model but includes dropout layers for regularization.
    - **Best Hyperparameters**:
        - Activation: ReLU
        - Dropout Rate: 0.5
        - Learning rate: 0.0001
4. **Seven-Layer Model**:
    - A deeper model with five hidden layers.
    - **Best Hyperparameters**:
        - Activation: Tanh
        - Number of units: 256, 512, 512, 128, 256
        - Learning rate: 0.001

The models were evaluated using the Adam optimizer and categorical_crossentropy loss function. Hyperparameter tuning was performed using the Hyperband search algorithm provided by keras_tuner.

3. Model Training and Evaluation

Each model was trained using an early stopping callback to avoid overfitting. The training and validation data were split using an 80/20 ratio. The models were trained for a maximum of 100 epochs with a batch size of 10.

Evaluation Metrics

- **Confusion Matrix**: Used to analyze the distribution of true positives, true negatives, false positives, and false negatives.
- **Precision, Recall, and F1-Score**: Calculated to evaluate the performance on each class.

The results for each model are summarized in Table 1:

Model Name	Precision (Class 0)	Precision (Class 1)	Recall (Class 0)	Recall (Class 1)	F1-Score (Class 0)	F1-Score (Class 1)
Three-Layer Model	0.91	0.88	0.90	0.89	0.90	0.88
Five-Layer Model	0.93	0.89	0.91	0.90	0.92	0.89
Five-Layer Model with Dropout	0.92	0.91	0.93	0.91	0.92	0.91
Seven-Layer Model	0.94	0.92	0.92	0.93	0.93	0.92

4. Final Model Selection

Based on the evaluation metrics, the seven-layer model demonstrated the best overall performance, with the highest F1-scores for both classes. This model was chosen as the final model for deployment. However if we want to make a model lightweight we can use a 5 layer model as well.

Overall Prototype Readiness:

- **Integration**: The project successfully integrates all components, from AI-driven posture correction to pain management recommendations.
- **Scalability**: The system is designed to scale, with room for additional features like workout routines, advanced posture analysis, and real-time performance tracking.
- **Innovation**: The prototype showcases a unique blend of machine learning, neural networks, and AI-powered pain monitoring, making it a potential candidate for the Hack-A-Roo and entrepreneurial ventures like the UMKC Side Hustle Challenge.

Individual Contributions

Suryansh Patel – Project Leader/Manager & AI Model Integration Specialist

GitHub: https://github.com/suryansh-max
Role: Project Leader, Research Papers Reviewer, Full-Stack Developer
Responsibilities:

- **Project Management**: Suryansh Patel oversees the entire development process, coordinating all phases of the project and maintaining a clear vision to meet objectives and deadlines.
- **AI Model Development**: Responsible for leading the team in identifying the project scope, particularly focusing on video-related datasets for posture detection and pain monitoring.
- **Research & Model Implementation**: Reviewed research papers, identified relevant AI models, and set up a TensorFlow-based AI system for pain detection. Also played a key role in training models and addressing technical challenges related to data preprocessing and model optimization.

Key Contributions:

- Managed and led the team to align the project's goals, ensuring that the scope and objectives were clearly defined.
- Took charge of video data collection, setting up AI models for pain detection through TensorFlow, and refining models to improve performance.
- Drafted comprehensive technical reports and presentations to track milestones and deliverables.
- Researched existing AI models, identified a solution that could be modified for the team's needs, and integrated it into the project.
- Managed the integration of the AI model with backend services to ensure efficient data flow between the database and AI-driven insights.

Jainil Patel – Data Specialist & Frontend/Backend Integration

GitHub: https://github.com/jainilpatel98
Role: Data Sets Collector, AI/ML Specialist, Backend Developer
Responsibilities:

- **Data Preprocessing & Management**: Led the collection and organization of video and sensor data for posture detection and pain management models.
- **Backend Development**: Worked extensively on setting up Django applications for the backend, ensuring smooth integration with the frontend and AI models.
- **Frontend Integration**: Played a key role in connecting the backend with React.js/Vue.js frontend systems for real-time video streaming and feedback display.

Key Contributions:

- Led efforts in preprocessing video data for AI model training and ensuring smooth data integration between frontend and backend.
- Developed Django-based backend systems, allowing users to upload videos, stream exercises, and receive real-time feedback on posture corrections and pain detection.
- Focused on sensor and video data collection for comprehensive pain monitoring and analysis.
- Worked on generating real-time outputs from the AI model, including visual feedback and error counts that were displayed on the frontend.

Cameron O'Dell – Research Paper Reviewer & Frontend Developer

GitHub: https://github.com/cam-odell
Role: Research Paper Reviewer, Front-End Developer, Deployment Specialist
Responsibilities:

- **Research & Analysis**: Conducted extensive research on AI models for pain and posture detection, guiding the technical approach for the system's development.
- **Frontend Development**: Developed the frontend to ensure AI-generated insights were displayed clearly to users, enabling real-time interaction with the AI models.
- **Deployment**: Identified and implemented deployment tools and environments to facilitate easy deployment of the AI system in both development and production.

Key Contributions:

- Reviewed and analyzed research papers, guiding the integration of AI models for posture correction.
- Developed the user interface for real-time AI feedback on posture corrections, ensuring a smooth user experience across various devices.
- Collaborated on deploying the project using industry-standard tools to optimize the model's performance and ensure scalability.

Jacob Xayaphet – Data Collection Specialist & UX Researcher

GitHub: https://github.com/NoviceOfCode
Role: Data Analyst, LLM Integration, UX Researcher
Responsibilities:

- **Data Collection & Analysis**: Gathered and processed relevant data, focusing on both video and sensor-based datasets for training AI models.
- **UX Research**: Worked closely on improving the user experience, ensuring that users receive actionable feedback on their posture and pain management in real time.
- **LLM Integration**: Played a key role in incorporating large language models (LLMs) into the system to enhance AI feedback for pain detection.

Key Contributions:

- Collected and organized relevant video data for training the AI model, ensuring the data's quality and relevance.
- Integrated LLM technology to provide intelligent feedback on pain detection during exercise routines.
- Communicated with external experts and trainers to understand use cases and risks associated with posture correction and pain detection, ensuring the system addresses real-world needs.
- Led efforts in UX research to optimize user interaction with the AI models, focusing on ease of use and clarity of feedback.