SEG3904 Project Progress

Project Title: Hand Gesture Calculator

Overview:

The purpose of this project is to develop a Hand Gesture Calculator that interprets numerical hand gestures (0-9) using a webcam and performs basic arithmetic operations (addition, subtraction, multiplication, and division). The project will involve implementing computer vision techniques and machine learning algorithms to detect and classify hand gestures in real-time. The application will be developed using Python, leveraging libraries such as OpenCV and TensorFlow. This project will help in understanding how to develop and deploy real-time machine learning models for practical use cases.

Project Progress:

Week 1:

- Project Planning: Defined the project goals and requirements for the Hand Gesture Calculator.
- Environment Setup: Installed necessary tools, including Python, OpenCV, TensorFlow, and Keras.
- Initial File Structure: Set up the basic project structure and Git version control.

Week 2:

- Data Collection (Part 1): Started collecting hand gesture images (0-9) using the webcam.
- image_capture.py: Created the script to capture and save images in gesture-specific directories.
- Data Organization: Set up directories for each gesture and began the collection process.

Week 3:

- Data Preprocessing: Converted the collected images to grayscale and resized them to a uniform size (128x128).
- preprocess_images.py: Developed a script to preprocess images by converting them to grayscale and resizing.

Week 4:

- Data Preparation: Loaded the preprocessed images, labeled them, and split the dataset into training and testing sets.
- prepare_data.py: Created a script to organize, label, and split the data into training and testing sets (90/10 split).

Week 5:

- Model Design: Built a Convolutional Neural Network (CNN) model using TensorFlow and Keras for gesture recognition.
- Model Training: Trained the model using the collected and preprocessed gesture data.
- model.py: Created the script to define, train, and save the trained model.

Week 6:

- Real-Time Gesture Recognition: Implemented a real-time gesture recognition system using the webcam.
- real_time_recognition.py: Developed a script to capture gestures in real-time and predict hand gestures using the trained model.

Week 7:

- UI Enhancement: Added on-screen buttons for arithmetic operations (+, -, *, /), allowing users to perform calculations with recognized gestures.
- Operation Implementation: Enhanced the system to allow users to select operations via clickable buttons, and perform calculations using recognized hand gestures.

Week 8:

- Process Rework: Reworked the process modeling by integrating the preprocessing and loading of images directly into model.py.

Week 9:

- Data Collection (Part 2): Focused on improving model accuracy by collecting and adding more diverse images of hand gestures.

Week 10:

- Model Optimization: Continued working on improving accuracy by implementing data augmentation, a learning rate scheduler, and other advanced techniques to enhance the model's training process.

Week 11:

- Integration of MediaPipe: Integrated MediaPipe for hand landmark detection, enabling the extraction of 3D coordinates (x, y, z) of 21 key points for each hand gesture. This replaced the previous image-based gesture recognition approach.
- Landmark Dataset Creation: Captured and processed hand gestures to extract landmarks using MediaPipe.
- Model Redesign: Updated the model to use landmarks instead of raw images, improving accuracy and training efficiency.
- UI Implementation (Version 1): Created the first version of the UI using OpenCV, integrating:
    - Real-time gesture recognition with live webcam feed.
    - On-screen buttons for arithmetic operations (+, -, *, /).

- Display of gestures, selected operation, and calculated results on the webcam feed.

- Directory Reorganization: Reorganized the project directory to separate scripts and data for better clarity and maintainability.