Hand Gesture Recognition System

Project Description:

This project aims to build a hand gesture recognition system that uses a camera to detect and classify different hand gestures in real time. The system relies on deep learning and computer vision techniques to understand hand movements and turn them into digital commands. It can be used to control devices, interact with applications, or provide hands-free accessibility options.

Group Members & Roles:

Team Members:

- Ahmed Mohamed Ahmed Albadawy Abdel-Gawad
- Hazem Ahmed Mohamed Salem
- Youssef Fady Gamil Mahrous
- Abdelrahman Tarek Mansour El-Sherbiny
- Mohamed Hatem Salah Ibrahim Rashed

Below is a table explaining each team member's assigned tasks and the reason behind this work division.

Milestone	Task	Assigned Members	Reason
1- Data Collection, Preprocessing & Exploration	Data Collection	All Team members	To build a custom dataset with diverse hand shapes and gestures from all members, improving the model's ability to generalize.
	Data Preprocessing	Ahmed Mohamed Hazem Ahmed	----------
2- Model Development and Training	Model selection, training, evaluation, and optimization	Youssef Fady Abdelrahman Tarek Mohamed Hatem	Assigning multiple members increases model diversity and allows for comparing multiple approaches to choose the best-performing model.
3- Real-Time Gesture Recognition and Deployment	Real-time gesture recognition (OpenCV)	Ahmed Mohamed	----------
	Deployment	Hazem Ahmed	----------
	Application Integration:	Abdelrahman Tarek	----------
4- MLOps & Monitoring	MLOps Setup	Youssef Fady	----------
	Continuous Monitoring	Mohamed Hatem	----------
5- Final Documentation and Presentation	Final report, presentation, future improvements	All Team members	All members contribute to reflect the full development process, discuss challenges, and propose future system enhancements.

Team Leader:

Name: Ahmed Mohamed Ahmed Albadawy Abdel-Gawad

Objectives:

The main objective of this project is to develop a real-time hand gesture recognition system using deep learning and computer vision. The system aims to accurately detect and classify hand gestures, enabling smooth and intuitive interaction with digital interfaces. Additionally, it focuses on delivering a scalable and reliable solution that can be effectively applied to real-world use cases.

Tools & Technologies:

This project involves a variety of tools and technologies covering the complete development, from dataset creation to deployment, monitoring, and documentation.

- **Programming & Scripting:** Python.
- **Computer Vision & Image Processing:** OpenCV, MediaPipe.
- **Data Handling & Visualization:** NumPy, Pandas, Matplotlib, Seaborn.
- **Deep Learning & Model Development:** TensorFlow, Keras, scikit-learn, Transfer learning models like MobileNet, ResNet, etc.
- **User Interface (UI) Development:** Python libraries such as Streamlit, Gradio, and Tkinter.
- **Cloud & Deployment:** Azure Cognitive Services.
- **MLOps & Version Control:** MLflow, DVC, Git, GitHub, TensorBoard.
- **Documentation & Presentation:** Microsoft Word, Google Docs, PowerPoint, Canva.

Milestones & Deadlines:

Milestone 1: Data Collection, Preprocessing, and Exploration:	1 week
Milestone 2: Model Development and Training:	1 week
Milestone 3: Real-Time Gesture Recognition and Deployment:	1-2 weeks
Milestone 4: MLOps Implementation and Model Monitoring:	2 weeks
Milestone 5: Final Documentation and Presentation:	1 week

KPIs (Key Performance Indicators):

1. Data Quality

- Percentage of missing values handled: 100%
- Data accuracy after preprocessing: 97%
- Dataset diversity (representation of different categories): 90%

2. Model Performance

- Model accuracy (Accuracy/F1-Score): 93%
- Model prediction speed (Latency): 60-135 milliseconds
- Error rate (False Positive/False Negative Rate): 7%

3. Deployment & Scalability

- API uptime: 99%
- Response time per request: 130-300 milliseconds
- (If applicable) Real-time processing speed (e.g., FPS for video models): 20–25 FPS

4. Business Impact & Practical Use

- Reduction in manual effort: 65%
- Expected cost savings: 35%
- User satisfaction: 85%