Contributions

Hamza Naeem: Deep Learning Algorithms and Overfitting Mitigation

- Orchestrated the implementation of crucial components including data processing, cleaning, and training pipelines.
- Executed algorithm training on both cloud platforms (Google Colab) and local environments, employing random search for fine-tuning hyperparameters.
- Effectively addressed common challenges associated with overfitting during the training of AI algorithms, applying tailored approaches such as early stopping and architecture modifications to mitigate overfitting risks.
- Implemented the conversion process for model compatibility with TensorFlow.js, enabling efficient deployment in web-based applications.
- Formulated a versatile pipeline for testing on images and real-time applications.
- Developed an API to scrutinize algorithm behavior, conducting rigorous testing using Postman for comprehensive evaluation.

Ali Ullah: Dataset Exploration and Model Development

- Explored datasets for Facial Emotion Recognition (FER) to understand data characteristics and requirements.
- Conducted an extensive literature review to analyze diverse methodologies for facial emotion recognition, including point-based techniques and deep learning models.
- Applied image preprocessing techniques, notably utilizing the Chehra detector for extracting facial feature points.
- Created a novel dataset incorporating distance metrics between facial feature points, enhancing data richness for model training.
- Designed and implemented a dense Neural Network architecture tailored for training deep learning models on extracted facial features.
- Explored and implemented data augmentation strategies and feature cleaning methodologies to optimize model performance and generalization.
- Conducted comprehensive training of algorithms across both cloud and local environments, employing grid search for hyperparameter tuning.

Ghulam Mustafa: Front-End/Back-End Integration and Application Development

- Built the Front-end view using Vanilla JavaScript, HTML, CSS, ensuring an intuitive user interface.
- Developed a user-friendly web application, designing APIs on Flask to manage back-end functionalities effectively.
- Integrated developed AI algorithms into the real-time system, ensuring seamless communication between the front-end and back-end.
- Utilized Flask to create RESTful APIs interacting with Python code for real-time detection and processing, enhancing application robustness.
- Attached statistical analyses for model performance evaluation, enabling real-time feedback on accuracy and effectiveness.

- Resolved critical bugs post-integration, enhancing application stability and reliability.
- Implemented features such as emotion tracking over time and data saving to disk, enriching the application's functionality and user engagement.

Collective Efforts: Visualization and Reporting

- Collaborated on creating an interactive visualization tool to build the final prototype, enhancing data presentation and user interaction.
- Worked on reporting the project's outcomes, including accuracy, performance metrics, and model comparisons, ensuring detailed and comprehensive documentation.
- Integrated additional features into the web application, significantly enhancing the overall user experience.
- Ensured a comprehensive and cohesive summary of the project's goals, methodology, and outcomes was available for stakeholders and future reference.