**Project Initialization and Planning Phase**

|  |  |
| --- | --- |
| Date | 15 June 2024 |
| Team ID | SWTID1750006853 |
| Project Title | ASL- Alphabet Image Recognition |
| Maximum Marks | 3 Marks |

**Project Proposal (Proposed Solution) template**

This project proposal outlines a solution to address a specific problem. With a clear objective, defined scope, and a concise problem statement, the proposed solution details the approach, key features, and resource requirements, including hardware, software, and personnel.

|  |  |
| --- | --- |
| **Project Overview** | |
| Objective | To provide a means for users to leverage machine learning to classify hand sign images into alphabets |
| Scope | This project focuses on the development of a machine learning-based image recognition system designed to classify American Sign Language (ASL) alphabet hand gestures. It will involve collecting or utilizing an existing dataset of ASL alphabet images, preprocessing the data, training a classification model, and developing a user-friendly interface to allow users to upload hand gesture images and receive alphabet predictions. The project is limited to static image input (not video) and covers only the ASL alphabet (A–Z), excluding dynamic gestures or full word recognition. |
| **Problem Statement** | |
| Description | Communication barriers exist for individuals who rely on American Sign Language (ASL), especially when interacting with those unfamiliar with sign language. Despite advancements in technology, there remains a lack of accessible tools that can translate ASL hand gestures particularly alphabet signs into readable text in real-time using simple image inputs. This gap makes everyday communication and integration more difficult for ASL users. |
| Impact | Solving this problem would greatly enhance accessibility and inclusivity for the deaf and hard-of-hearing community. It would enable better communication in educational, social, and professional environments by providing a bridge between ASL users and non-ASL users. It can also serve as a foundational tool for further development into full ASL sentence recognition systems and educational platforms. |
| **Proposed Solution** | |
| Approach | The project leverages Convolutional Neural Networks (CNNs), specifically a fine-tuned VGG16 model pretrained on ImageNet, to classify ASL alphabet hand gestures from static images. The pipeline includes:   * **Data Collection/Preparation:** Use of an existing ASL alphabet image dataset or creation of a custom one. * **Preprocessing:** Image normalization, resizing, and augmentation to improve model generalization. * **Model Training:** Design and train a CNN using frameworks such as TensorFlow or PyTorch. * **Evaluation:** Measure accuracy and performance using validation datasets and confusion matrices. * **Deployment:** Develop a simple interface (web or desktop) that allows users to upload an image and receive a predicted letter. |
| Key Features | * **Transfer Learning with VGG16:** for high-accuracy classification. * **Data Augmentation:** to improve generalization. * **t-SNE Visualization:** to explore learned features in 2D. * **Image-based Prediction Interface:** with real-time output. * **Extensible Codebase:** suitable for adapting to full ASL word recognition in future iterations. |

**Resource Requirements**

|  |  |  |
| --- | --- | --- |
| **Resource Type** | **Description** | **Specification/Allocation** |
| **Hardware** | | |
| Computing Resources | CPU/GPU specifications, number of cores | 1 x NVIDIA RTX 3060 GPU, 8-core CPU |
| Memory | RAM specifications | 16 GB DDR4 RAM |
| Storage | Disk space for data, models, and logs | 500 GB SSD |
| **Software** | | |
| Frameworks | Python frameworks | Flask (for web interface), TensorFlow/Keras (for model training) |
| Libraries | Additional libraries | NumPy, OpenCV, scikit-learn, Matplotlib |
| Development Environment | IDE, version control | Visual Studio Code,  Jupyter Notebook,  Git for version control |
| **Data** | | |
| Data | Source, size, format | Kaggle ASL Alphabet dataset, ~87,000 labeled JPG images |