1. Project Idea

- **Sign Language Numbers to Text**
- **Why This Project?**
    1. There is a need to bridge the communication gap between deaf individuals and the rest of society. Approximately **5% of the global population (430 million people)** suffer from disabling hearing loss (source: WHO). This project addresses this gap by making numbers in sign language understandable to everyone.
    2. This project can be extended in the future to translate full sign language sentences, contributing to accessibility tools for the deaf community.

---

2. Dataset

- **Dataset Name**: Sign Language Digits Dataset https://github.com/ardamavi/Sign-Language-Digits-Dataset
- **Details**:
    - **Data**:
        - **Features (Images)**: Contains the image data for hand gestures.
        - **Labels (Numbers)**: Contains the labels for each hand gesture (0–9).
    - **Image Format**: Colored (RGB)
    - **Resolution**: 100x100 pixels per image.
    - **Number of Classes**: 10 (digits 0 through 9).

---

3. Technical Features from the Code

The code includes several techniques and algorithms:

1. **Data Handling and Preprocessing**:
    - Normalizes pixel values (scales them to the range [0, 1]).
    - Reshapes data for use with both traditional ML classifiers and neural networks.
2. **Traditional Machine Learning Models**:
    - **K-Nearest Neighbors (KNN)**: For classification using Euclidean distance.
    - **Logistic Regression**: A linear model for multi-class classification.
    - **Support Vector Machine (SVM)**: With radial basis function (RBF) kernel.

3. **Deep Learning (Neural Networks)**:
    - **Convolutional Neural Network (CNN)** with:
        - Three convolutional layers.
        - Max-pooling layers to reduce dimensionality.
        - Fully connected layers for classification.
        - Dropout for regularization.
4. **Performance Metrics**:
    - Accuracy, Precision, Recall, and F1-score.
    - Confusion matrix visualization for error analysis.
5. **Visualization**:
    - Displays sample images.
    - Plots learning curves for the neural network.

---

4. Project Scope and Goals

- **Primary Objective**: Translate sign language digits (0–9) into corresponding numbers using both traditional ML models and a CNN.
- **Expected Outcome**: The program will take hand gesture images as input and output the correct digit label.

---

5. Tools and Libraries

- **Development Platform**: Google Colab.
- **Key Libraries**:
    - **Numpy**: Data manipulation.
    - **Pandas**: Data analysis.
    - **Matplotlib/Seaborn**: Visualization.
    - **scikit-learn**: Traditional ML classifiers.
    - **TensorFlow/Keras**: Deep learning (CNN).
    - **OpenCV**: For reading images.

---

6. Timeline

Phase 1: Setup and Data Preparation (16–18 Dec)

- Search for a suitable dataset.
- Load and explore the dataset.
- Visualize sample images to ensure proper loading.

Phase 2: Feature Extraction and Model Training (19–21 Dec)

- Normalize data and prepare it for traditional ML and CNN models.
- Implement and test the KNN, Logistic Regression, and SVM models.
- Evaluate models using accuracy, precision, recall, and F1-score.

Phase 3: Deep Learning (22–23 Dec)

- Build and train the CNN model.
- Validate results and tune hyperparameters.
- Plot training and validation curves for the CNN.

Phase 4: Visualization and Finalization (24 Dec)

- Generate confusion matrices for all models.
- Compare the performance of KNN, Logistic Regression, SVM, and CNN.
- Write the final report and prepare for presentation.
