Reflective Report: British Sign Language Recognition Model

Data Collection Approach

The dataset for British Sign Language (BSL) recognition was created through live image capture using OpenCV. The system captures grayscale images of hand signs at a resolution of 48x48 pixels. A webcam feed is utilized, and a region of interest (ROI) is defined to focus on the hand sign. A total of 250 images per label are collected, covering letters A-Z , except H and J as their signs are dynamic and cant be detected using image classification and a blank category.

The collected dataset is structured into separate directories for each class. A data augmentation strategy was not explicitly applied during collection, but preprocessing steps like resizing and grayscaling were performed to maintain consistency. The dataset is then split into training (80%) and validation (20%) sets using the splitfolders library.

Core Model Design and Rationale

The model follows a Convolutional Neural Network (CNN) architecture optimized for image classification. The structure consists of multiple convolutional layers with ReLU activation, followed by max pooling layers for feature extraction and dimensionality reduction. Dropout layers are strategically added to prevent overfitting.

Architecture Details:

- **First Layer:** A Conv2D layer with 128 filters of size 3x3, followed by MaxPooling and Dropout (40%)
- **Second Layer:** A Conv2D layer with 256 filters, followed by MaxPooling and Dropout (40%)
- **Third Layer:** A Conv2D layer with 512 filters, followed by MaxPooling and Dropout (40%)
- **Fourth Layer:** Another Conv2D layer with 512 filters, followed by MaxPooling and Dropout (40%)
- **Fully Connected Layers:**
    - Dense layer with 512 neurons (ReLU activation) and 40% dropout
    - Dense layer with 64 neurons and 20% dropout
    - Dense layer with 256 neurons and 30% dropout
    - Additional layers with similar configurations to enhance feature learning
    - Output layer with 25 neurons (Softmax activation) for multi-class classification

The model is compiled using the **Adam** optimizer with **categorical cross-entropy** as the loss function, ensuring stability during training. The training process involves 100 epochs with batch normalization, improving generalization.

Model Deployment and Testing

The trained model is saved in JSON and H5 formats for future use. For real-time inference, the model processes live video feed, extracts hand features, and classifies them into corresponding BSL letters. The final output displays the predicted sign along with confidence scores.

This approach ensures an efficient and scalable BSL recognition system with real-time performance.