Project Title

SignSpeak: American Sign Language Recognition Using Deep Learning

---

Abstract

SignSpeak is a deep learning-powered application designed to recognize static hand gestures representing American Sign Language (ASL) alphabet signs. The model classifies ASL signs from images with high accuracy, aiming to bridge communication barriers for the Deaf and Hard-of-Hearing community. It serves as a foundational tool for gesture-to-text systems and future real-time applications.

Project Objectives

This project aims to create a lightweight yet accurate AI model capable of classifying ASL alphabet gestures from image input. It seeks to demonstrate how computer vision and deep learning can contribute to inclusive technology, specifically focusing on accessibility for non-verbal users.

Methodology

The solution uses a custom Convolutional Neural Network (CNN) and a transfer learning-based MobileNetV2 architecture to classify hand signs into 29 ASL categories. The project is built and trained using TensorFlow and Keras in a Python environment. The dataset used is sourced from Kaggle, consisting of over 87,000 labeled images across the 29 ASL classes. After training, the model achieved a validation accuracy of 92.6%.

The model is structured for easy integration into real-time webcam systems (e.g., OpenCV or browser-based inference), and future plans include deploying the model using TensorFlow Lite or ONNX for edge device support.

Key Findings

- The final trained MobileNetV2-based model achieved a **92.6% validation accuracy**.
- The model is compact and well-suited for **real-time classification tasks** using a webcam.
- **Overfitting was minimized** using dropout layers, early stopping, and dynamic learning rate scheduling.
- The confusion matrix revealed common misclassifications between similar signs such as 'M', 'N', and 'T'.
- The training pipeline supports further fine-tuning and can be extended to dynamic gestures in the future.

Step-wise Solution Approach

Step 1: Load and analyze the ASL Alphabet Dataset (87,000+ images across 29 categories).
Step 2: Preprocess images using resizing, normalization, and augmentation.
Step 3: Build a custom CNN architecture and evaluate initial performance.
Step 4: Replace base model with **MobileNetV2** and apply transfer learning.
Step 5: Train model with early stopping and learning rate reduction on plateau.
Step 6: Evaluate the model using accuracy/loss curves and a confusion matrix.
Step 7: Save the model in `.keras` format and plan deployment for webcam integration.

Reference

1. Kaggle: ASL Alphabet Dataset – https://www.kaggle.com/datasets/grassknoted/asl-alphabet
2. Howard, A. G., et al. "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications." arXiv:1704.04861 (2017).
3. TensorFlow Documentation – https://www.tensorflow.org/
4. Simonyan, K. and Zisserman, A. "Very Deep Convolutional Networks for Large-Scale Image Recognition." ICLR, 2015.

Team

Suryansh Chaudhary