American Sign Language Alphabets Recognition Using Computer Vision

Pranjal Upadhyay

Indian Institute of Science Education and Research, Bhopal

Pranjalu22@iiserb.ac.in (22244)

Problem statement

This project aims to train an AI model capable of extracting the text from the visual representations of Sign Language. In this project phase, I aim to recognize individual alphabets from static hand gestures captured in image format in different lighting conditions and backgrounds. The goal is to train a robust sign recognition and classification system that translates visual inputs to corresponding alphabets efficiently and accurately.

Motivation

Thousands of individuals worldwide are hearing or speech impaired, relying on sign language as their primary means of communication. However, most normal people do not understand sign language, creating a significant communication barrier. In today's digital era, where meetings and interviews are increasingly conducted online, individuals with hearing impairments face challenges in fully participating. This project seeks to address this issue by developing a model that can efficiently and accurately translate sign language into text. This solution would remove the dependency on human translators, enabling seamless communication and integration into everyday life for the deaf community in online as well as offline channels.

Challenges

This project has the following challenges and limitations:

- Lighting variations of the training data. The training images are captured in different lighting conditions.
- Background cluttering. The training images have some set of images that are cluttered with the background.
- Object detection. We have to accurately detect the hand gesture from the image. •

Classification. We will classify the gesture as one of the A-Z alphabets of English.

Datasets

- My model is independent of the dataset so, according to the suggestions from Prof. Akshaya Agarwal, I have two datasets 1st is the American Sign Language dataset and 2nd is the Indian Sign Language dataset. I have trained my model on the first dataset but it can be replaced by 2 or any other similar dataset just by changing the number of classes.

The 1st dataset is from Kaggle, ASL Alphabet which is an open-source dataset. The data set is a collection of images of alphabets from American Sign Language, separated into 29 folders that represent the various classes. The training data set contains 87,000 images which are 200x200 pixels. There are 29 classes, of which 26 are for the letters A-Z and 3 classes for SPACE, DELETE, and NOTHING. These 3 classes are very helpful in real-time applications, and classification. The test data set contains a mere 29 images, to encourage the use of real-world test images.

DATASET LINK: ASL Alphabet

The 2n dataset is a collection of images representing Indian Sign Language (ISL) alphabets, organized into folders for each letter, except H, J, and Y, which are absent. Each folder contains 20 to 50 images, standardized to 126x126 pixels. The images were captured under varied conditions, featuring gestures performed by the dataset creator and collaborators. Controlled noise was added, including messy, blurry, and colorful backgrounds, to mimic real-world scenarios. This dataset is designed to train models for effective ISL recognition in dynamic environments, offering a unique resource for research and development.

DATASET LINK: Indian Sign Language Dataset

Algorithm/Model

To address the task of American Sign Language (ASL) alphabet recognition, a hybrid deep learning model combining VGG-style convolutional layers and residual blocks was developed. The model architecture is designed to extract robust features from images while effectively mitigating vanishing gradient issues, ensuring optimal performance on a diverse dataset with dynamic environmental conditions. Below is a detailed description of the methodology:

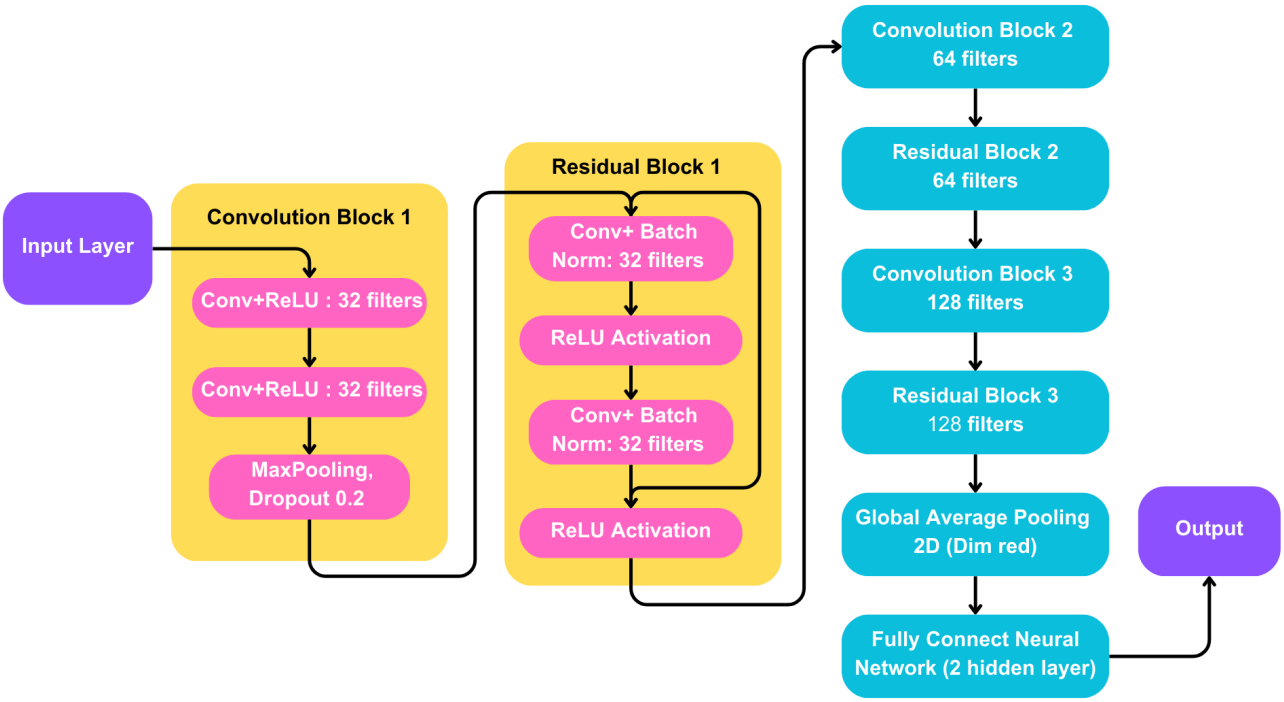
Data Preprocessing

1. **Image Augmentation**: The training dataset was augmented using ImageDataGenerator to simulate real-world variability. Augmentations included:
    - Random rotations (up to 15 degrees).
    - Width and height shifts (up to 10%).
    - Shear transformations.
    - Zoom variations (up to 10%).
    - Horizontal flips and nearest-neighbor filling for added robustness.
2. **Rescaling**: All images were normalized to a pixel range of [0, 1], ensuring uniformity in input data and stable convergence during training.
3. **Target Size**: Input images were resized to 64×6464 \times 6464×64 pixels with three color channels (RGB).

Model Architecture

The hybrid model combines convolutional layers and residual blocks to balance complexity and generalization, ensuring high accuracy while preventing overfitting. Key components include:

1. **Input Layer**: Accepts 64×64×364 \times 64 \times 364×64×3 input images.
2. **Feature Extraction**:
    - **Initial Convolutional Layers**: Two 3×33 \times 33×3 convolutional layers with 32 filters, followed by max pooling and dropout (20%).
    - **Residual Blocks**: Incorporated at increasing depths (32, 64, and 128 filters) to enhance feature extraction by learning residual mappings. Each block consists of two 3×33 \times 33×3 convolutional layers, batch normalization, and a skip connection for gradient flow.
    - **VGG-style Convolutions**: Intermediate convolutional layers with increasing filters (64 and 128) followed by max pooling and dropout, improving hierarchical feature learning.
3. **Global Feature Aggregation**:
    - A global average pooling layer reduces feature maps to a single vector, preserving spatially important features while reducing dimensionality.
4. **Fully Connected Layers**:
    - Two dense layers with 256 and 128 neurons, each followed by ReLU activation and dropout (50% and 30%, respectively).
5. **Output Layer**: A dense layer with 29 neurons (softmax activation) for multi-class classification of ASL alphabets, including special classes (SPACE, DELETE, NOTHING).



Training

- **Loss Function**: Categorical Cross-Entropy was used for multi-class classification.
- **Optimizer**: The Adam optimizer with a learning rate of 0.001 facilitated efficient gradient updates.
- **Callbacks**: Early stopping and model checkpointing ensured optimal convergence by halting training when validation loss ceased improving and saving the best-performing model.

Evaluation

The model was evaluated using a test set consisting of preprocessed ASL gesture images. Metrics included:

- **Accuracy**: Percentage of correctly classified images.
- **Classification Report**: Precision, recall, and F1-score for each class.
- **Confusion Matrix**: Visualization of class-wise performance to identify misclassification patterns.

Prediction and Visualization

To assess real-world applicability, the model was deployed to predict classes for unseen test images. The predictions were visualized alongside the input images, demonstrating the model's

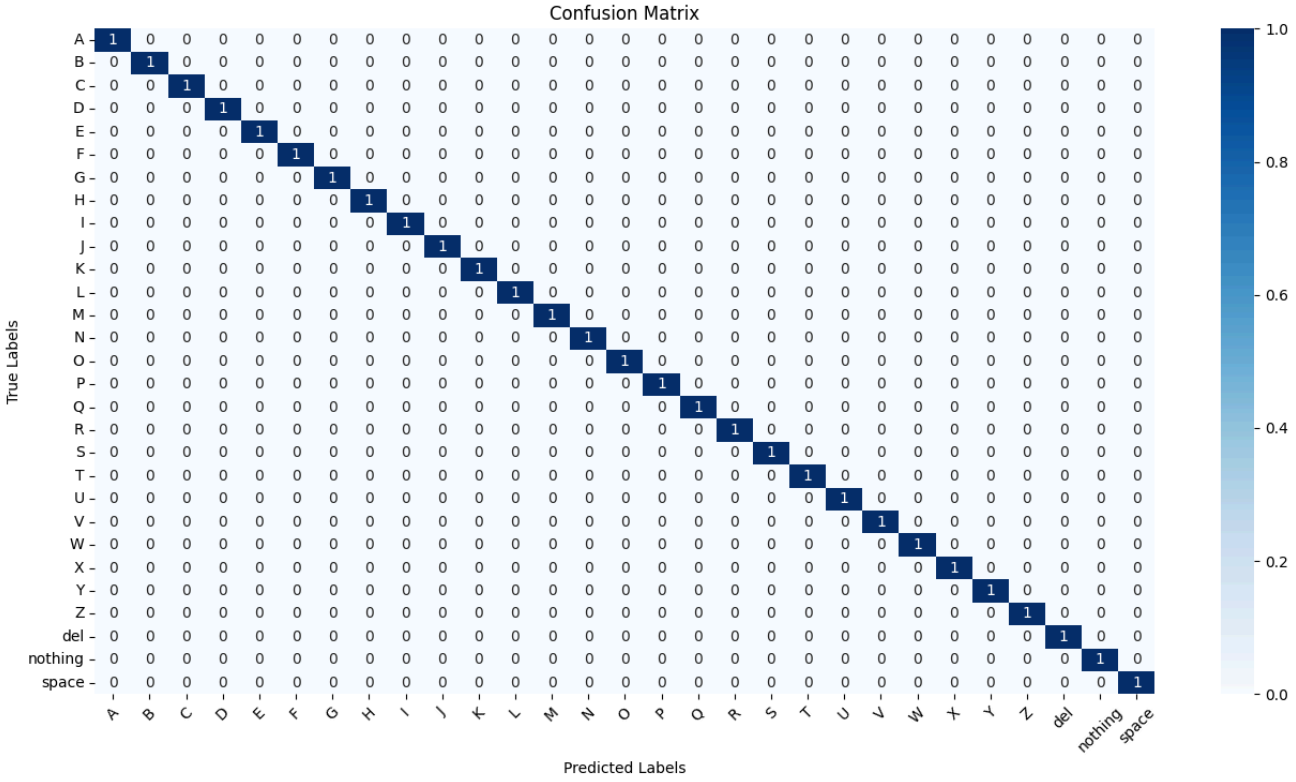
interpretability and accuracy.

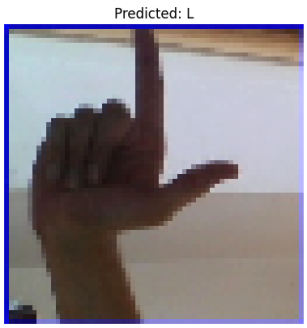
This hybrid architecture leverages the strengths of residual learning and VGG-style convolutions to achieve robust performance on the ASL alphabet dataset, offering a scalable solution for sign language recognition tasks.

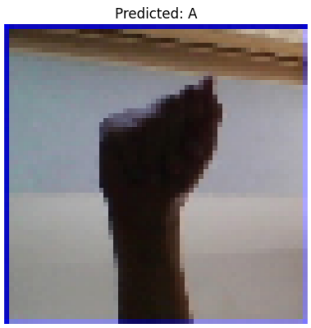
Results

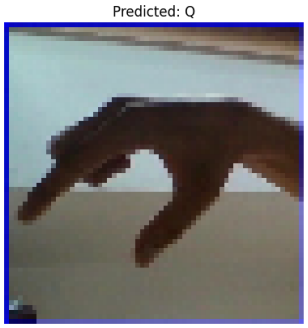
The hybrid deep learning model achieved remarkable performance on the ASL alphabet recognition dataset. The evaluation results are summarized below:

- **Accuracy:** The model achieved 100% accuracy on the test set, demonstrating its ability to correctly classify all 29 classes (26 alphabets and 3 special classes: space, delete, and nothing).
- **Classification Report:** Precision, recall, and F1-scores were 1.00 for all classes, indicating perfect classification across all categories.
- **Confusion Matrix:** The absence of misclassifications in the confusion matrix further validates the robustness of the model.









Conclusion

The hybrid model effectively combines VGG-style convolutional layers and residual blocks to extract robust features and overcome vanishing gradient challenges. By incorporating data augmentation techniques and leveraging dropout for regularization, the model generalizes well to unseen data, as reflected in the perfect classification scores.

The results demonstrate the model's suitability for real-world applications, such as assistive technologies for individuals with hearing or speech impairments. Future work could focus on extending the dataset with more images and incorporating temporal information for dynamic gesture recognition.

Bibliography

1. Chollet, F. (2017). Deep Learning with Python. Manning Publications.
2. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770-778.
3. Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. *International Conference on Learning Representations (ICLR)*.
4. TensorFlow Documentation: *ImageDataGenerator*.
5. Kaggle. ASL Dataset, Indian Sign Language dataset