A Comparative Analysis of Deep Learning Models for Indian Sign Language Recognition

[1]Nimisha D R, [2]Kavitha K V

[1]PG Scholar [2]Associate Professor
Department of Computer Science and Engineering
Sree Chitra Thirunal College of Engineering, Kerala, India
[1]drnimi23@gmail.com, [2]kavitha@sctce.ac.in

***Abstract*—This paper presents GestureNet, a deep learning framework for the recognition of Indian Sign Language (ISL) gestures using five prominent convolutional neural network architectures: InceptionV3, ResNet50V2, InceptionResNetV2, VGG19, and MobileNetV2. The system aims to identify the most effective model for ISL recognition through performance metrics such as accuracy, precision, recall, and F1-score. Results show MobileNetV2 achieves the highest accuracy (99.14%), offering a lightweight and efficient solution for real-time applications. A Flask-based web interface allows users to select a model and upload images for gesture prediction.**

***Index Terms*—Indian Sign Language, Gesture Recognition, Deep Learning, CNN, MobileNetV2, Flask UI**

I. INTRODUCTION

Sign language recognition is a vital component in bridging the communication gap between hearing and non-hearing individuals. Indian Sign Language (ISL) is used by millions in India, yet tools for its automated recognition are limited. This work utilizes state-of-the-art deep learning architectures to build an accurate and accessible ISL recognition system.

CNNs have emerged as effective tools for image-based classification tasks. They extract hierarchical spatial features, making them suitable for detecting complex hand gestures. This study evaluates and compares the performance of five models for static ISL image recognition.

Deep learning models have shown significant promise in sign language detection tasks due to their ability to learn complex patterns and features from large datasets[1]. **Convolutional Neural Networks (CNNs)**, in particular, are widely used for image-based tasks, including sign language detection, as they are effective at capturing spatial hierarchies of features in images, making them suitable for recognizing hand shapes and movements[10]. Different deep learning architectures can perform disparately in terms of accuracy, speed, robustness, and generalization ability on sign language detection tasks[2]. A **comparative analysis** of these models is vital to identify the most effective approach for a given application, provide insights into how different architectures learn and represent sign language gestures, and identify resource-efficient methods.

This work specifically focuses on **Indian Sign Language (ISL)**, which is the primary sign language used by the deaf community in India. ISL is a unique visual-gestural language that uses handshapes, facial expressions, body movements, and spatial orientation to convey meaning. For instance, each letter of the alphabet is represented by a specific handshape, formed using fingers, thumbs, and palm.

II.METHODOLOGY

A. Dataset

The dataset includes over 30,000 images representing the 26 English alphabet gestures, 10 digits (0–9), space, and full-stop symbols. It was collected from Kaggle and contains RGB images with a resolution of 256x256 pixels. The dataset is well-balanced and divided into training (70%), validation (15%), and test (15%) sets.

B. Preprocessing and Augmentation

All images are resized to 256x256 pixels. Preprocessing involves normalization to scale pixel values between 0 and 1, and one-hot encoding of class labels. Data augmentation techniques include:

- Horizontal and vertical flipping
- Random zoom (up to 30%)
- Rotation (±40 degrees)
- Width and height shift (up to 20%)
- Brightness adjustment and shearing

These transformations help improve the robustness of the models and mitigate overfitting.

C. Model Architectures

Data Collection: The dataset used comprises a comprehensive collection of **Indian Sign Language (ISL) hand gestures**, representing the 26 letters of the English alphabet, numerals 0-9, as well as a full stop and a space. This dataset was sourced from Kaggle.

Data Preprocessing: This stage is crucial for preparing the input data for model training[]. It involves tasks such as **removing irrelevant or noisy data**, **handling missing values**, and **standardizing or normalizing data**[3]. The primary aim is to improve data quality for accurate analysis.

Data Augmentation: To enhance the model's generalization ability and stability, data augmentation is employed to diversify the dataset. This involves applying various transformations to the original images, including **zooming, rotating, shifting in height and width, shearing, adjusting brightness, and horizontal and vertical flipping**[9]. These transformations create variations that help the model learn to generalize better.

Feature Extraction: Input images are processed through the convolutional layers of a pre-trained network to **extract meaningful features**, which are then used for model training. This process generates **feature maps** that capture significant patterns and structures, serving as a compact representation of the visual content[5]. These extracted features serve as input for the neural network's subsequent layers to classify images.

Model Training: The five deep learning models — InceptionV3, ResNet50V2, InceptionResNetV2, VGG19, and MobileNetV2 — were each trained for **10 epochs**.

InceptionV3: This CNN architecture, developed by Google, optimizes accuracy and efficiency through **factorization of convolutions** (breaking larger convolutions into smaller ones), **asymmetric convolutions**, and **label smoothing** to prevent overconfidence and improve generalization. Its ability to use multiple filter sizes (1x1, 3x3, 5x5) in parallel allows it to capture features at various scales, which is advantageous for recognizing both global and local patterns in hand gestures for sign language detection. Despite its depth, InceptionV3 is computationally efficient due to Inception modules that have fewer parameters[2].

ResNet50V2: An improved version of the original ResNet50, this deep CNN architecture introduces **pre-activation residual blocks** where batch normalization and ReLU activation precede convolutional layers, enhancing training stability and gradient flow. Its use of residual connections helps alleviate the vanishing gradient problem, enabling efficient training of very deep networks. The 50-layer depth allows it to recognize intricate patterns and hierarchical representations of hand movements and gestures[6].

Inception-ResNet-v2: This model combines the strengths of Inception networks (efficient multi-scale feature capture) and Residual networks (training benefits of residual connections). It integrates **residual connections into Inception modules**, maintaining high computational efficiency while effectively training very deep networks. Its ability to capture multi-scale features and robust training characteristics make it suitable for distinguishing subtle differences in hand shapes and movements in sign language[3].

VGG19: Developed by the Visual Geometry Group, this deep CNN architecture is known for its simplicity and efficiency[8]. Its 19 layers and use of small 3x3 filters excel at **hierarchical feature extraction**, from edges to complex patterns, which is ideal for recognizing minute details of sign language gestures. VGG19's effectiveness in handling spatial hierarchies and its ability to process high-resolution images are crucial for accurate sign language detection[4].

MobileNetV2: Designed by Google researchers for efficient deep learning on mobile and embedded systems, MobileNetV2 introduces **inverted residual blocks** and **linear bottlenecks**[6]. It reduces parameters and computations using **depthwise separable convolutions** and global average pooling. Its lightweight and efficient design makes it ideal for real-time sign language detection on mobile platforms, allowing for rapid inference, easier deployment, and energy savings[7].

We use transfer learning with pre-trained ImageNet weights for all five CNN models. The top layers are removed and replaced with:

- Global Average Pooling layer
- Fully Connected Dense layer (256 units, ReLU)
- Dropout layer (rate=0.5)
- Final Dense layer with Softmax activation (number of units = number of classes)

Each model is trained using the Adam optimizer with categorical cross-entropy loss. The initial learning rate is set to 0.0001, and early stopping is employed to prevent overfitting.

D. Training Details

Each model is trained for 10 epochs with a batch size of 32. Training is performed in a GPU-enabled environment. Accuracy and loss are monitored on both training and validation sets. Models are evaluated on the test set after training.

E. Evaluation Metrics

To assess model performance, we use:

- **Accuracy** = (TP + TN)/(TP + TN + FP + FN)
- **Precision** = TP/(TP + FP)
- **Recall** = TP/(TP + FN)
- **F1 Score** = 2 * (Precision * Recall)/(Precision + Recall)
- **Confusion Matrix**: To visualize classification performance per class

III. Results and Analysis

The comparative analysis of the deep learning models' accuracy revealed the following:

MobileNetV2: 99.14%
Inception-ResNet-v2: 98.86%
InceptionV3: 94.70%
ResNet50V2: 93.13%
VGG19: 89.84%

From these results, **MobileNetV2 performed the best**, demonstrating high precision. Inception-ResNet-v2 was a close second, achieving high accuracy by combining Inception and residual connections. InceptionV3 and ResNet50V2 showed good performance but were less accurate than MobileNetV2 or Inception-ResNet-v2. VGG19, despite being a deep network, showed the lowest accuracy among the evaluated models.

IV. Conclusions

This study demonstrates the effectiveness of CNNs for ISL gesture recognition. The selection of the appropriate model should consider the balance between accuracy, computational efficiency, and the specific requirements of the task. Among the tested models, MobileNetV2 stands out for its accuracy and efficiency.

Comparing deep learning models for sign language detection facilitates progress in the field by identifying effective techniques, improving understanding of model behavior, and guiding the development of more accurate and efficient systems. Future work will focus on dynamic gesture recognition and real-time video integration.

REFERENCES

[1] N. Sarhan and S. Frintrop, "Transfer learning for videos: From action recognition to sign language recognition," in 2020 IEEE International Conference on Image Processing (ICIP), 2020.
[2] Szegedy et al., "Rethinking the Inception Architecture," CVPR, 2016.
[3] He et al., "Deep Residual Learning for Image Recognition," CVPR, 2016.
[4] Carreira & Zisserman, "Quo Vadis, Action Recognition?," CVPR, 2017.
[5] Simonyan & Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," ICLR, 2015.
[6] Sandler et al., "MobileNetV2: Inverted Residuals and Linear Bottlenecks," CVPR, 2018.
[7] A. Ali, M. Zakariah, W. Hatamleh, H. Tarazi, V. Tripathi, and E. Amoatey, "Humancomputer interaction with hand gesture recognition using resnet and mobilenet," Computational Intelligence and Neuroscience, 2022.
[8] A. Das, S. Gawde, K. Suratwala, and D. Kalbande, "Sign language recognition using deep learning on custom processed static gesture images," in 2018 International Conference on Smart City and Emerging Technology (ICSCET), pp. 1–6, IEEE, 2018.
[9] P. Rathi, R. K. Gupta, S. Agarwal, and A. Shukla, "Sign language recognition using resnet50 deep neural network architecture," in 5th International Conference on Next Generation Computing Technologies (NGCT-2019), 2020
[10] R. G. Rajan and M. Judith Leo, "American sign language alphabets recognition using hand crafted and deep learning features," in 2020 International Conference on Inventive Computation Technologies (ICICT), pp. 430–434, 2020.