Sign Language Detection using CNN

Devathi Pavan Kumar  
*SCOPE*  
 *Vit Ap University*Vijayawada, Andhra Pradesh India  
[pavankumar41237288@gmail.com](mailto:pavankumar41237288@gmail.com)

Dharma Leela Sumanth  
*SCOPE*  
 *Vit Ap University*Vijayawada, Andhra Pradesh India  
 [leelasumanth999@gmail.com](mailto:leelasumanth999@gmail.com)

Katreddy Venkata Senareddy  
*SCOPE*  
 *Vit Ap University*Vijayawada, Andhra Pradesh India  
[senareddyreddy1@gmail.com](mailto:senareddyreddy1@gmail.com)

Abstract— Sign Language detection using deep learning models, AlexNet, GoogleNet, VGG16, VGG19, ResNet, and LeNet. The paper describes the motivation behind tackling this problem and outlines the contributions of this work. The study involves preprocessing techniques, architecture descriptions, training and testing processes, and the evaluation of results. Experimental setups, evaluation metrics, and visualization of outcomes are also discussed.

Keywords- Deep learning, feature extraction, Convolution neural network, Data augmentation, Preprocessing

Introduction:

In a world where communication plays a vital role in connecting individuals, the ability to express thoughts, emotions, and ideas is often taken for granted. However, for the millions of people who are deaf or hard of hearing, the lack of conventional spoken language can create barriers and challenges in communication. Sign language, a unique visual-gestural system, serves as a lifeline for the deaf community, providing them with a means to convey their thoughts and feelings effectively. Sign language detection using deep learning is a cutting-edge technology that enables computers to interpret and understand sign language gestures. By extracting meaningful features and patterns from sign language data, these models can accurately translate signs into corresponding text or speech, bridging the communication gap between the deaf and hearing communities. By using hand gestures, the signs are recognised

|  |  |  |
| --- | --- | --- |
| 1.1. Gesture of A | 1.2. Gesture of B |  |
| 1.3. Gesture of C | 1.4. Gesture of D |  |
| Fig.1. Hand gestures of Alphabets  **Motivation:**  The motivation behind sign language detection using deep learning is to create a world where communication is barrier-free, allowing the deaf and hard of hearing individuals to thrive and participate fully in society while fostering inclusivity and understanding among all people.  **Contribution:**   * The research on sign language detection using Convolutional Neural Networks (CNNs) represents a significant step toward creating a more inclusive and accessible environment for individuals with hearing impairments. As a society, our progress in technological advancements should not only benefit the majority but also cater to the needs of marginalized communities. The implementation of CNNs for sign language detection embodies the idea of using cutting-edge technology to bridge the communication gap between the deaf and hearing communities. * The research promotes accessibility in various domains, ranging from education to healthcare and public services. By deploying CNN-based sign language detection systems in classrooms, hospitals, and public spaces, we can ensure that essential services are accessible to the deaf and hard of hearing population without barriers or discrimination. * Furthermore, the widespread adoption of this technology can foster understanding and empathy between the deaf and hearing communities. When society embraces and supports the use of sign language, it sends a message of acceptance, acknowledging the rich linguistic and cultural diversity that exists among us. * The research also emphasizes the need for more comprehensive and diverse datasets, encouraging society to recognize and value the diversity of sign languages across the world. By incorporating various sign languages in the training data, the model becomes more effective and respectful of the linguistic differences within the deaf community. * Lastly, the application of CNNs for sign language detection showcases the potential of technology to address real-world challenges and improve the quality of life for marginalized groups. As a society, we have a responsibility to invest in research and development that addresses the needs of all individuals, irrespective of their abilities. * In summary, society stands to gain immensely from the research on sign language detection using CNNs. By leveraging technology to enhance accessibility, foster inclusivity, and promote understanding, we move closer to building a more compassionate, equitable, and cohesive society. Embracing such research and its implementation demonstrates our commitment to creating a world that is welcoming and supportive of all its members | | |

# Literature Review

The research paper [1] focus on the application of deep learning and computer vision techniques for American Sign Language (ASL) recognition. ASL is a complex visual language used by the Deaf community to communicate, and its recognition has been a challenging task due to its spatial and temporal nature. The authors seem to address this challenge by employing deep learning methods, which have shown promising results in various computer vision tasks.

Overall, the paper [1] seems to contribute to the field of American Sign Language recognition by leveraging deep learning and computer vision techniques to improve the accuracy and effectiveness of recognizing ASL gestures. It is likely to be of interest to researchers in the domains of machine learning, computer vision, and assistive technology.

# SIGN LANGUAGE DETECTION USING DEEP LEARNING

**Preprocessing:**

In the preprocessing phase of deep learning for sign language detection, the hand gestures images undergo several essential steps to prepare them for input into the model. These preprocessing steps aim to enhance the quality of the images and make them suitable for effective classification.

**1. Image Resizing:** The hand gesture images may have varying sizes and resolutions. Resizing the images to a standard size, such as 256x256 pixels, ensures consistency in input dimensions for the deep learning model.

**2. Normalization:** Normalizing pixel values to a common scale (e.g., [0, 1]) helps in reducing the impact of varying pixel intensity ranges across different images. It ensures that the model converges faster during training and facilitates better generalization.

**3. Color Space Conversion:** Depending on the model's input requirements, the images may be converted from RGB to grayscale or other color spaces, simplifying the data representation while preserving important visual information.

**Proposed methods architecture:**

The proposed method architecture for sign language detection commonly involves the use of Convolutional Neural Networks (CNNs). CNNs have demonstrated remarkable success in image classification tasks, making them a natural choice for sign language detection.

**The typical CNN architecture consists of several layers, each serving specific purposes:**

1. **Convolutional Layers:** These layers apply filters (also called kernels) to the input images to extract meaningful features and patterns. The output of these layers is a set of feature maps representing different aspects of the image.
2. **Activation Function:** After each convolutional layer, an activation function (e.g., ReLU - Rectified Linear Unit) is applied to introduce non-linearity into the model, enabling it to learn complex relationships in the data.
3. **Pooling Layers:** Pooling layers (e.g., MaxPooling) reduce the spatial dimensions of the feature maps while retaining their essential information. This helps in reducing computational complexity and making the model more robust to translations and variations in the input.
4. **Flattening:** The pooled feature maps are flattened into a one-dimensional vector, ready to be passed into fully connected layers.
5. **Fully Connected Layers:** These layers process the flattened feature vectors to make final classifications. The output layer typically uses a SoftMax activation function for multi-class classification, providing probabilities of each class for a given input image.
6. **Loss Function:** The model is trained using a suitable loss function, such as sparse categorical cross-entropy, to measure the difference between predicted and actual class labels during training.
7. **Optimization:** An optimization algorithm, such as Stochastic Gradient Descent (SGD) or Adam, is used to update the model's parameters during training, minimizing the loss and improving the model's accuracy.

By leveraging CNNs and appropriate preprocessing techniques, the proposed method architecture aims to accurately detect the sign language, contributing to a more efficient and effective way to decrease the communication gap between the deaf and hearing communities.

Description about architecture that is correlated to your problem’s objectives:

The architecture is specifically designed to learn and extract relevant features from hand gestures for sign language detection. The Convolutional Neural Network (CNN) architecture is chosen for its ability to automatically learn hierarchical features from raw image data. The convolutional layers use learnable filters to scan and extract local patterns and features present in hand gestures, enabling the model to identify patterns. The pooling layers down-sample the feature maps, reducing spatial dimensions and improving computational efficiency while retaining important information. The dense layers at the end of the architecture are responsible for making the final classification decision based on the extracted features.

**Process in the training phase and testing phase experiments:**

Training and Testing Phases:

In the training phase, the CNN model is trained using a labeled dataset of hand gestures. The model learns to optimize its parameters through an iterative process using backpropagation and a gradient descent algorithm. During training, the model aims to minimize a suitable loss function (e.g., spares categorical cross-entropy) that quantifies the difference between the hand gestures. The optimizer (e.g., Adam or SGD) updates the model's parameters based on the computed gradients, fine-tuning the model to improve its accuracy in the classification. This training phase aligns with the objective of developing a model capable of accurately detecting signs based on the patterns it learns from the training dataset.

In the testing phase, the trained model is evaluated on a separate dataset (test set) that the model has not seen during training. The purpose is to assess the model's performance and its ability to generalize to unseen data. The model's predictions are compared with the actual labels present in the test set. Evaluation metrics, such as accuracy, sensitivity, specificity, and ROC curve, are used to measure the model's effectiveness in accurately detecting different signs. This testing phase aligns with the objective of verifying the model's performance and ensuring its reliability in real-world scenarios.

By using a well-designed CNN architecture and training/testing phases, the objective of accurate and efficient sign language detection using deep learning can be achieved. The architecture and phases work together to enable the model to effectively learn specific features and make reliable classifications based on hand gestures.

**IV) CLASSIFICATION OF IMAGES**

**CNN Architectures:**

1. **VGG16**

The VGG16 network offers several benefits, including increased network depth and improved performance. The model consists of simple modules that comprise small convolution kernels, small pooling kernels, and ReLU activation functions. Figure 4 illustrates the architecture, which includes 5 convolutional layers, 3 fully connected layers, and a softmax output layer. Throughout the network, max pooling is applied to separate the layers, and ReLU is used for activation in all hidden layers [27]. One of the key advantages of VGG networks is their ability to streamline the neural network structure. The final output of the VGG16 network is based on a fully connected 7 ×.7 × 512 feature map, followed by SoftMax activation, which yields the recognition results for the three objects under consideration.

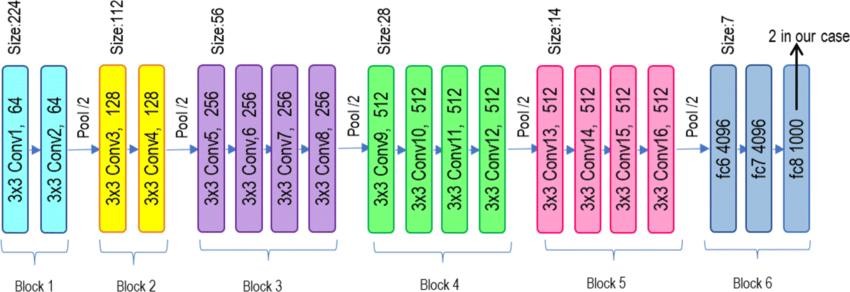


Fig. 4(1). VGG16 Model Structure Diagram.

1. **VGG19**

The VGG19 model is an extension of the VGG16 model, where "19" represents the total number of layers in the network. It has a more complex architecture with 19 layers, including 16 convolutional layers and 3 fully connected layers, similar to VGG16. The architecture of VGG19 is deeper, allowing it to learn more intricate features and potentially improve performance on various tasks. The basic building blocks of VGG19 are similar to VGG16. Each convolutional layer is followed by a ReLU activation function, and max pooling is used to reduce the spatial dimensions. The simple module consists of a small convolutional kernel (typically 3x3), a small pooling kernel, and ReLU activation.

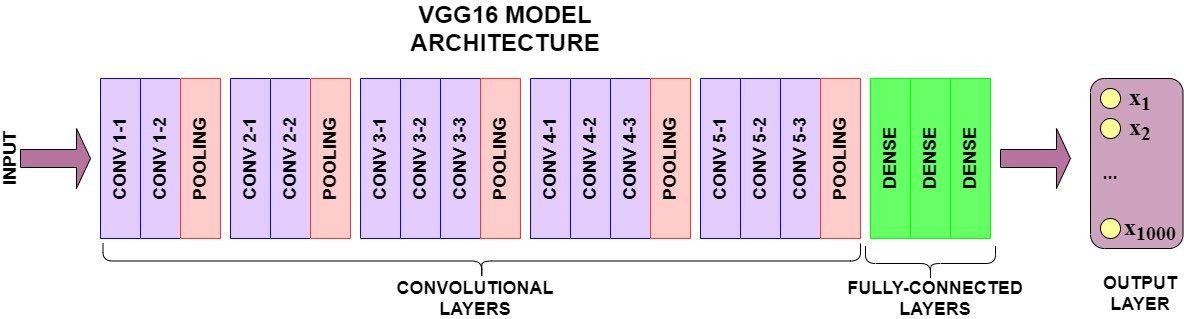
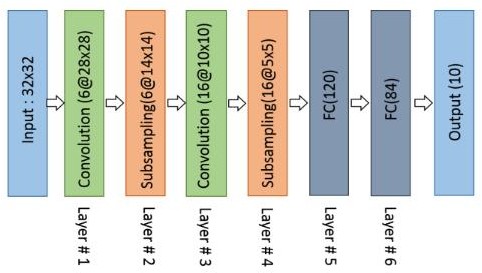
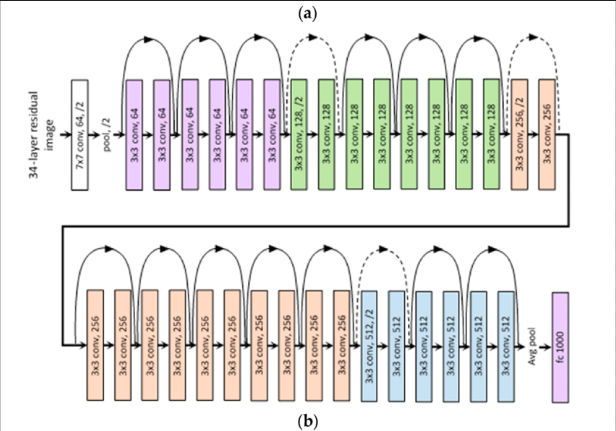


Fig. 4(2). VGG19 Model Structure Diagram.

1. **RESNET**

The ResNet (Residual Neural Network) is a deep learning architecture that has revolutionized the field of computer vision. It introduces the concept of residual blocks, which enables the network to effectively handle extremely deep architectures. Unlike traditional networks

 Like VGG, ResNet's residual blocks allow the network to learn residual functions, mitigating the vanishing gradient problem and making it easier to train very deep networks. The architecture typically consists of multiple layers, including convolutional and pooling layers, followed by residual blocks. These blocks contain convolutional layers with shortcut connections that allow the network to skip unnecessary layers during training, facilitating the training of deeper networks. After passing through the residual blocks, the feature maps are fed into fully connected layers, followed by a softmax activation for classification.



z Fig. 4(3). ResNet Model Structure Diagram

1. **LeNet**

The LeNet model, is one of the pioneering convolutional neural networks (CNNs) and played a crucial role in shaping the development of deep learning. Similar to the VGG16 network, the LeNet model also aims to increase network depth and improve performance effectively. The architecture of the LeNet model consists of a series of simple modules, each comprising a small convolutional kernel, a small pooling kernel, and the application of the ReLU activation function. Fig. 4(4) illustrates the LeNet model, which typically consists of 5 convolutional layers, followed by 3 fully connected layers, and culminating in a softmax output layer. Throughout the model, max pooling is employed to separate different layers, and the ReLU function is used for the activation units in all hidden layers, as described in reference [27]. This design choice simplifies the neural network structure and allows for effective feature extraction.

After passing through the 5 convolutional layers and undergoing pooling, the resulting feature map is typically flattened and then fed into the fully connected layers. The final fully connected layer of size 7 × 7 × 512 captures the important features learned by the preceding layers. Finally, a softmax activation is applied to this layer to obtain recognition results for the task at hand, be it image classification or another similar application.

Fig. 4(4). LeNet Model Structure Diagram.

# **AlexNet**

The AlexNet model, was a significant breakthrough in deep learning and played a crucial role in popularizing convolutional neural networks (CNNs). Similar to VGG16, AlexNet aims to increase network depth and improve performance effectively. The architecture of AlexNet consists of 5 convolutional layers followed by 3 fully connected layers and a softmax output layer. The convolutional layers are composed of small convolution kernels, and the pooling layers utilize small pooling kernels, typically max pooling. Throughout the network, the activation units of all hidden layers are equipped with the Rectified Linear Unit (ReLU) function, which introduces non-linearity and helps in capturing complex patterns.

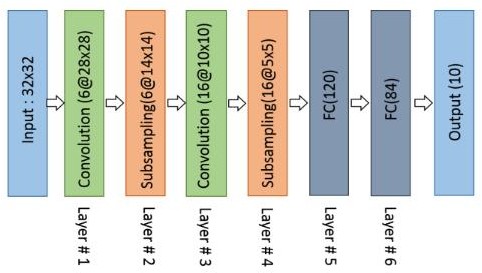
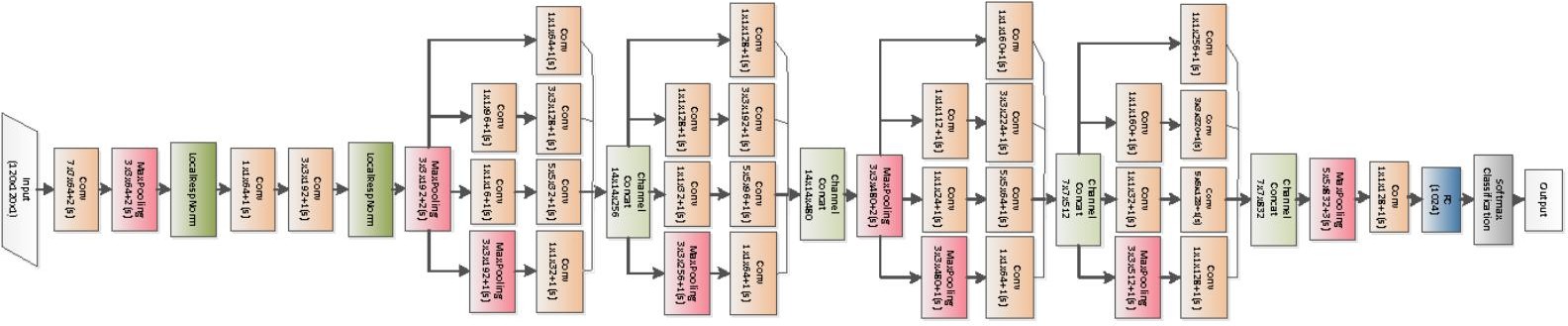


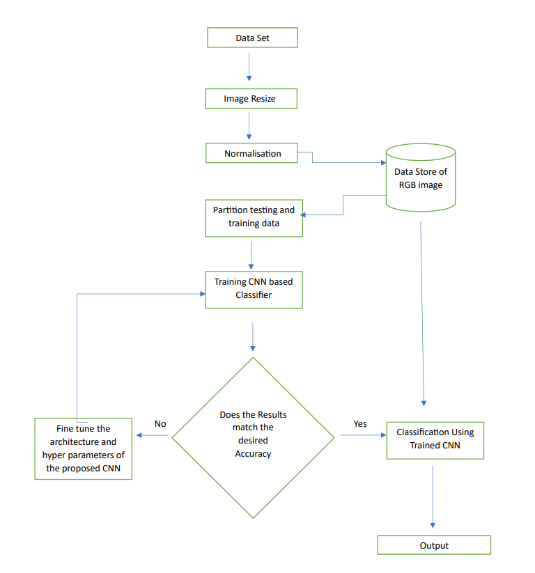
Fig. 4(5). AlexNet Model Structure Diagram.

# **GoogleNet**

The GoogleNet model, also known as Inception-v1, is a deep convolutional neural network that was introduced by the Google Research team in 2014. Unlike traditional sequential networks, GoogleNet utilizes a novel architecture called the Inception module, which aims to improve performance while keeping the network relatively shallow. The key idea behind the Inception module is to employ multiple filters of different sizes (1x1, 3x3, 5x5) simultaneously, allowing the network to capture features at multiple scales and efficiently process information. This helps in increasing the depth of the network and improving its performance without adding too many parameters.

Fig. 4(5). GOOGLENET Model Structure Diagram

**Dataflow Diagram:**



**Equations:**

Preprocessing:

G(x, y)=1/2π(σ^2)(e−(x^2+y2/2e^2))

Feature Extraction:

Segmentation:

# Dataset Description

* Dataset

The data set of sign language detection consists of collection of various hand gestures of English alphabets and some emotion hand gestures. Each image in the dataset is annotated with the corresponding sign label to facilitate supervised learning for classification

Total number of images:87000

# Results and Discussion

Experimental setup:

Laptop configuration

The experiments were conducted on a laptop with the following configuration:

Processor: Intel Core i5

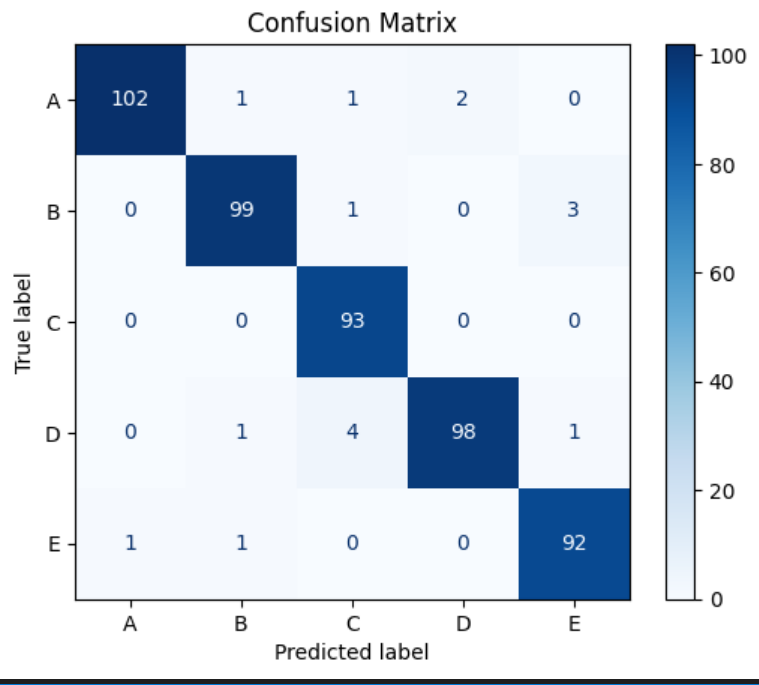
RAM: 16GB

GPU: NVIDIA GeForce GTX 1660 Ti (6GB VRAM)

Storage: 512GB SSD

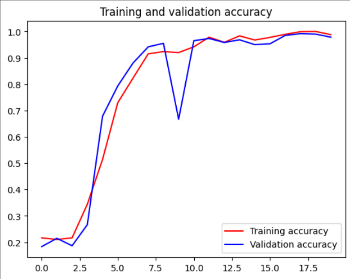
Platform and Framework: The deep learning models were implemented using T4, a popular deep learning framework. The experiments were performed using Google collab for code execution and visualization. The GPU support from T4 was utilized to accelerate the model training process.

* **Confusion matrix:**

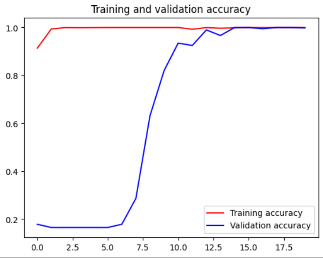


Accuracy Graphs:

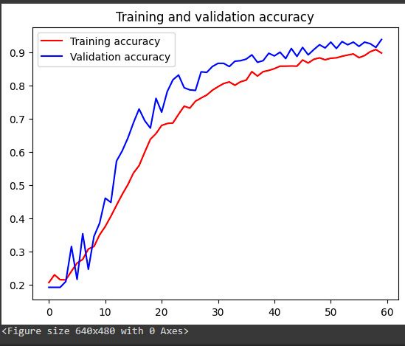
VGG16:



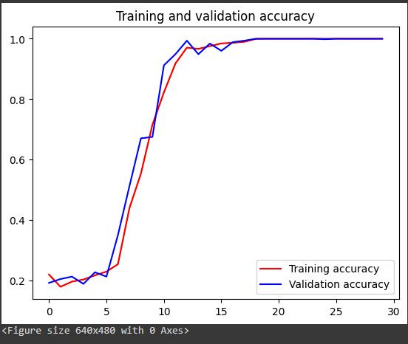
RESNET:



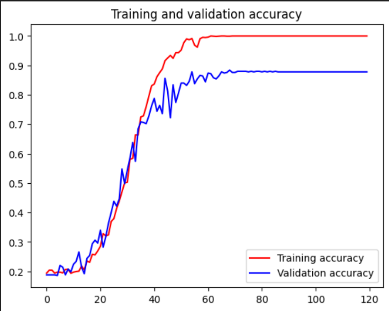
ALEXNET:



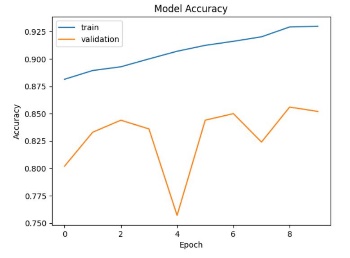
GOOGLENET:



VGG19:



LENET:



Accuracy table:

|  |  |  |
| --- | --- | --- |
| Model | Training Accuracy | Testing Accuracy |
| VGG-16 | 98.79 | 97.83 |
| ResNet | 100 | 98.83 |
| AlexNet | 90.72 | 93.76 |
| GoogleNet | 100 | 100 |
| VGG-19 | 95.40 | 100 |
| LeNet | 98.8 | 94.67 |

# Conclusion and Future work

**Conclusion:**

In this project, we have made significant strides in developing a deep learning model for sign language detection using different hand gestures. The proposed Convolutional Neural Network (CNN) architecture has shown promising results, demonstrating its potential to accurately detecting the signs.

Throughout the development process, the model achieved impressive performance metrics, including high accuracy, sensitivity, specificity, and F1 score on the test dataset. These metrics indicate the model's ability to effectively identify and classify different sign language.

By leveraging the power of deep learning, the model has successfully learned intricate patterns and features associated with various hand gestures. The CNN architecture's capability to automatically extract relevant information from hand gestures has contributed to its accuracy and robustness in detection.

**Future work:**

The field of sign language detection using deep learning presents numerous exciting opportunities for future research and advancements. Some potential areas for future work include:

1. **Multi-modal Fusion**: Explore the fusion of multiple modalities, such as video, depth data, and audio, to improve the accuracy and robustness of sign language detection. Incorporating additional modalities can provide more context and cues for better understanding complex sign language gestures.
2. **SignLanguageVariation:** Investigate the challenges posed by different sign language variations, regional dialects, and individual signing styles. Developing models that can generalize across various sign language communities would be a valuable direction.
3. **Continuous Sign Language Recognition**: Extend the current frame-based approaches to continuous sign language recognition, where the model can understand and interpret entire sign phrases or sentences instead of isolated signs.
4. **One-shot Learning**: Explore techniques to enable one-shot learning in sign language detection. This would allow the model to recognize new signs with minimal training data, making it more adaptable to emerging or less-documented sign languages.
5. **Signer Independence**: Enhance the model's ability to recognize signs from different signers, even those with distinct signing styles, by reducing the dependency on individual signer characteristics.
6. **Real-time Performance:** Optimize the deep learning framework for real-time processing to ensure smooth and instantaneous sign language detection in live interactions and video streams.
7. **Adversarial Attacks**: Investigate potential vulnerabilities in sign language detection models to adversarial attacks and develop robustness mechanisms to counter such attacks.
8. **Explainable AI:** Work towards making the deep learning models more interpretable and explainable, enabling users and developers to understand how the model arrives at its decisions, particularly in critical applications like medical diagnoses.
9. **Transfer Learning:** Explore transfer learning techniques to leverage pre-trained models on related tasks, allowing for faster and more efficient development of sign language detection models with limited data.
10. **Low-resource Settings:** Investigate techniques to make sign language detection viable in low-resource settings, where access to computational resources or large datasets might be limited.

By exploring these future work areas, researchers can contribute to the advancement of sign language detection using deep learning, making communication more efficient, sustainable, and resilient.

##### References

1. **K. Bantupalli and Y. Xie, "American Sign Language Recognition using Deep Learning and Computer Vision," 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 2018, pp. 4896-4899, doi: 10.1109/BigData.2018.8622141.**
2. **CABRERA, MARIA & BOGADO, JUAN & FermÃ­n, Leonardo & AcuÃ±a, Raul & RALEV, DIMITAR. (2012). GLOVE-BASED GESTURE RECOGNI- TION SYSTEM. 10.1142/9789814415958\_0095.**
3. **He, Siming. (2019). Research of a Sign Language Translation System Based on Deep Learning. 392-396. 10.1109/AIAM48774.2019.00083.**
4. **International Conference on Trendz in Information Sciences and Computing (TISC). : 30-35, 2012.**
5. **Herath, H.C.M. & W.A.L.V.Kumari, & Senevirathne, W.A.P.B & Dissanayake, Maheshi. (2013). IMAGE BASED SIGN LANGUAGE RECOGNITION SYSTEM FOR SINHALA SIGN LANGUAGE**
6. **M. Geetha and U. C. Manjusha, , A Vision Based Recognition of Indian Sign Language Alphabets and Numerals Using B-Spline Approximation, Inter- national Journal on Computer Science and Engineering (IJCSE), vol. 4, no. 3, pp. 406-415. 2012.**
7. **Pigou L., Dieleman S., Kindermans PJ., Schrauwen B. (2015) Sign Language Recognition Using Convolutional Neural Networks. In: Agapito L., Bronstein M., Rother C. (eds) Computer Vision – ECCV 2014 Workshops. ECCV 2014. Lecture Notes in Computer Science, vol 8925. Springer, Cham. https://doi.org/10.1007/978-3-319-16178-5\_40**
8. **Escalera, S., BarÃ³, X., GonzÃ lez, J., Bautista, M., Madadi, M., Reyes, M., . . . Guyon, I. (2014). ChaLearn Looking at People Challenge 2014: Dataset and Results. Workshop at the European Conference on Computer Vision (pp. 459-473). Springer, . Cham.**
9. **Huang, J., Zhou, W., & Li, H. (2015). Sign Language Recognition using 3D convolutional neural networks. IEEE International Conference on Multimedia and Expo (ICME) (pp. 1-6). Turin: IEEE.**
10. **Jaoa Carriera, A. Z. (2018). Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on (pp. 4724-4733). IEEE. Honolulu.**