Volume 13, Issue 6
December 2024

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X

Deep Learning Models for Devnagari Sign Language Recognition

[1]Deepali R. Naglot and Deepa Deshpande

[1,2]Department of Computer Science and Engineering, JNEC MGM University,
Chhatrapati Sambhajinagar, Maharashtra, India
E-mail: [1]dnaglot@mgmu.ac.in, [2]ddeshpande@mgmu.ac.in

ABSTRACT

Sign language serves as the primary mode of communication for individuals experiencing hearing and speech impairments. This research paper focuses on developing a novel deep learning model for a vision-based system to recognize and interpret Devnagari Sign Language (DSL), which has not been extensively explored. Deep learning, particularly deep Convolutional neural networks, has garnered enduring popularity among researchers, with various architectures continually emerging. However, selecting the optimal network from these architectures remains challenging due to their reliance on finely tuning optimization hyperparameters, a nontrivial endeavor. This research marks the initial comprehensive evaluation of pre-trained deep learning models, including InceptionV3, AlexNet, VGG16, and ResNet50, specifically tailored for recognizing static Devnagari sign language alphabets. A unique dataset comprising 47 DSL alphabets captured from different age groups using mobile phone camera and evaluated using metrics such as precision, recall, F1-score, and support. InceptionV3 achieved the highest testing accuracy of 94.98% among the four pre-trained deep models.

Keywords: *Convolution Neural Network CNN, Sign language recognition SLR, Devnagari Sign Language DSL*

1. INTRODUCTION

Hearing loss stands out as the primary sensory obstacle confronting individuals today. WHO statistics indicate that around 63 million peodple in India grapple with substantial auditory deficiencies, constituting approximately 6.3% of the nation's populace. Recent NSSO data underscores a prevalence rate of 291 individuals per one lakh population suffering from severe to profound hearing impairments, with a significant portion being children aged 0 to 14 years. This sizable demographic of young Indians facing hearing challenges poses a significant hindrance to both physical and economic productivity. Additionally, a noteworthy segment of the population experiences less severe forms of hearing loss, including unilateral impairment [26, 28].

Sign language, a form of communication rooted in visual expressions, relies on hand movements and facial cues to convey meaning. It serves as a comprehensive mode of exchange of information for people with hearing disabilities, enabling them to articulate letters, words, and sentences through a repertoire of hand signs. This linguistic system, dating back to ancient civilizations, predates conventional writing methods and remains an integral means of expression globally. There persists a significant communication gap between hearing-impaired individuals and the broader society. Addressing this gap is imperative, particularly the millions in India affected by varying degrees of hearing loss. To bridge this disparity, this research endeavors to develop a machine learning-driven conversational interface, tailored to incorporate Devnagari Sign Language (DSL) gestures, thereby facilitating communication for the differently-abled population in India.

In contemporary times, significant works have been dedicated to the advancement of technologies capable of categorizing gestures from various sign languages into predefined categories. These systems have demonstrated utility across diverse domains, including gaming, virtual reality simulations, robotic operations, and facilitating natural language interactions. Currently, Devnagari Sign Language (DSL) recognition systems are undergoing development, with no real-time sign recognition systems available for complete alphabets and numbers. Consequently, there exists a pressing demand for the creation of comprehensive recognition systems capable of identifying DSL signs efficiently.

Devnagari Sign Language (DSL) is a visual-gestural communication system used by deaf and hard-of-hearing individuals in India who communicate using sign language. DSL chiefly founded on the Devnagari script used in written and spoken languages such as Hindi, Marathi and Sanskrit in India. The Devnagari script consists of 47 core characters, encompassing 14 vowels and 33 consonants, supplemented by diverse diacritic marks to denote extra sounds and alterations. It is notable for the unique horizontal line atop each letter and is traditionally written from left to right [27]. However, sign languages are distinct from written or spoken languages, as they have their grammar, syntax, and vocabulary.

The research aims to improve DSL gesture recognition under constrained conditions. The novelty of

this research is to classify and recognize all the 47 alphabets based on Devnagari Sign Language using different approaches. The contributions of the research can be summarized as follows:

1. Comprehensive dataset has been meticulously curated, encompassing diverse age groups and genders. This extensive dataset provides a thorough analyses and ensuring the reliability of subsequent findings.
2. The research conducts a detailed examination and comparison of the efficiencies exhibited by various pre-trained deep Convolutional Neural Network (CNN) models. By systematically evaluating these models, the study sheds light on their respective strengths and weaknesses, thus providing perspectives for future research.
3. The research introduces a robust classification methodology specifically tailored for Devnagari sign language alphabets.

This paper is organized as follows: Section 1 covers the introduction Section 2 discusses the related works carried out across various kinds of systems for recognizing sign language Section 3 illustrates the pre-trained deep model and proposed system. The algorithm is implemented and results are discussed in Section 5. The observations are concluded and future possibilities to improve the performances are outlined in Section 6.

2. RELATED WORK

Oyedotun & Khashman (2017) proposed CNN and stacked denoising autoencoders (SDAEs) for American Sign Language (ASL) hand gestures recognition. Their systems achieved recognition rates of 91.33% and 92.83%, respectively, showcasing the efficacy of deep learning approaches in ASL recognition tasks [12].

Chong & Lee (2018) developed a Leap Motion Controller-based prototype for recognizing alphabets and digits of American Sign Language (ASL). Their system achieved commendable recognition rates for ASL letters and digits, highlighting the feasibility of using motion sensing devices for sign language recognition [8]. Tao et al. (2018) introduced a multiview augmentation and inference fusion approach for ASL recognition, addressing challenges such as complexity and occlusions in sign gestures. System achieved high recognition accuracies on benchmark datasets, highlighting the efficacy of multiview approaches in sign language recognition tasks [16].

Pansare and Ingle (2018) explored Devnagari Sign Language (DSL) recognition, employing template-based and clustering-based algorithms. Their approach achieved remarkable detection and recognition rates against complex backgrounds, contributing to the advancement of DSL recognition technology [2,3]. Avola et al. (2019) proposed a methodology for American Sign Language (ASL) recognition utilizing Recurrent Neural Networks (RNNs). Their system achieved remarkable accuracy in analyzing hand gestures over time, showcasing the potential of RNNs in capturing temporal dependencies in sign language sequences [7].

Wadhawan & Kumar (2020) focused on Indian Sign Language (ISL) recognition, utilizing a deep learning-based model trained on 35,000 sign images dataset representing 100 different signs. Their system achieved high training accuracy, showcasing the capability of deep learning in ISL recognition [4]. Deshpande & Kalbhor (2020) addressed Marathi Sign Language (MSL) recognition CNN. Their system demonstrated impressive accuracy for MSL alphabets, highlighting the efficacy of CNNs in sign language recognition tasks [1].

Sharma & Singh (2021) utilized a G-CNN model, along with modified versions of VGG11 and VGG-16 architectures, for accurate classification of sign language hand gestures. Their system attained notable classification accuracies for different categories of gestures, contributing to the advancement of sign language recognition technology [5]. Lee et al. (2021) presented a real-time recognition system for ASL using the Leap Motion controller, employing a LSTM Recurrent Neural Network with KNN for classification. Their system attained high recognition rates for ASL alphabets, demonstrating the potential of LSTM networks in accommodating both static and dynamic signs in sign language recognition [11]. Wangchuk et al. (2021) created a digit recognition system for British Sign Language (BSL) using CNN. Their model exhibited high training and testing accuracy for BSL recognition tasks [18].

Kasapbaşi et al. (2022) developed a dataset and CNN-based sign language interface system, achieving high accuracy with minimal loss. Their system showcased comprehensive datasets and durable model architectures in sign language recognition tasks [9]. Katoch et al. (2022) introduced an innovative approach for classifying and recognizing Indian Sign Language (ISL) signs, achieving impressive accuracy through the integration of Support Vector Machines and CNNs. Their system addressed challenges such as rotation invariance and background dependency, contributing to the advancement of ISL recognition technology [10].

Ansari & Harit developed a functional ISL recognition system using a Microsoft Kinect camera, achieving above 90% recognition rates for various ISL gestures, including fingerspelling. Their system provided a low-cost and user-friendly approach to interpreting sign language [6].

Table 1: Approaches of Different Sign Languages

Author	Sign Language	Sign Language Recognition Method	Recognition Rate
Oyedotun & Khashman (2017)	24 hand gesture of American Sign Language	CNN and Stacked Denoising Autoencoders (SDAEs)	91.33% (CNN), 92.83% (SDAEs)
Chong & Lee (2018)	26 letters of American Sign Language	Support vector machine (SVM) and a deep neural network (DNN)	80.30%(SVM) and 93.81%(DNN)
Wadhawan & Kumar (2020)	100 static signs of Indian Sign Language	Deep learning model	98.70%
Deshpande & Kalbhor (2020)	25 alphabets of Marathi Sign Language	Convolutional Neural Networks (CNNs)	99%
Sharma & Singh (2021)	26 alphabets of Indian Sign Language	SVM, VGG-16	98.52% (One-Hand) 97% (Two-Hand)

Numerous research studies have researched on different sign languages, including British, Australian and New Zealand Sign Language (BANZSL), American Sign Language (ASL), French Sign Language (LSF), Arabic Sign Language, Chinese Sign Language (CSL), Indian Sign Language (ISL), and Devnagari Sign Language (DSL) as shown in Table 1. Very few researchers worked on the Devnagari Sign Language so proposed research work is concentrate on Devnagari Sign Language by introducing first comparative analysis for DSL.

3. PROPOSED METHODOLOGY

This section delineates the proposed methodology for evaluating Pre-trained deep machine learning models and customized CNN. The data is captured from various people for the collection of Devnagari alphabet sign using mobile phone camera. The signers were both male and female with different age groups. All Devnagari signs are captured with a consistent background and with good lighting conditions. Data augmentation techniques are implemented to mitigate over fitting, thereby enhancing the robustness and generalization capabilities of the model. Proposed system Pre-trained models InceptionV3, ResNet50, VGG16, and AlexNet are selected based on their performance. Moreover, a tailored three-layered CNN architecture is designed, trained from the ground up, and compared with other models. The objective is to evaluate the deep learning architectures, optimization techniques, and parameter configurations concerning Devnagari Sign Language. The evaluation process follows steps as shown in Fig 1.

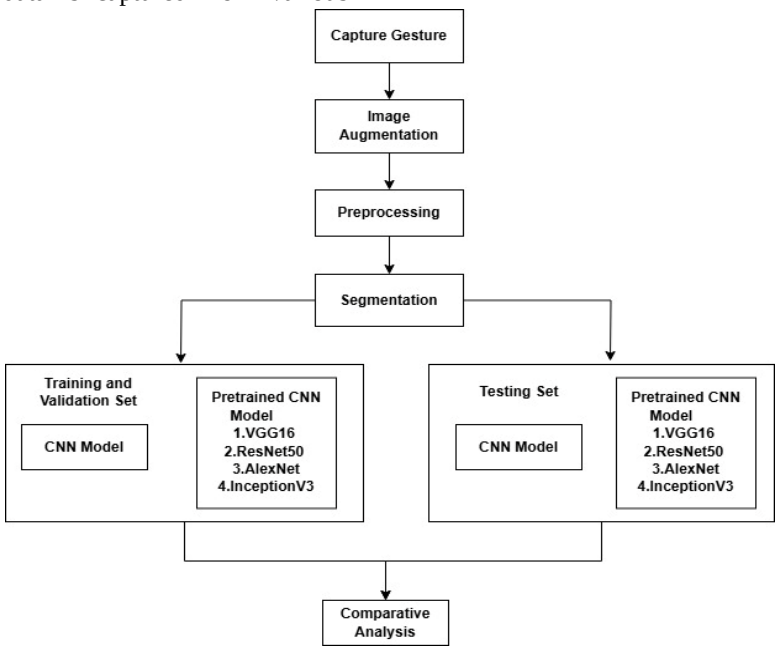

Fig 1. Flow of Proposed System

Volume 13, Issue 6
December 2024

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X

3.1. Dataset:

The database contains images for alphabets of the Devnagari Sign Language from (Aa) to (Dnya) as shown in Fig.2 [2,3]. The data is captured from various people for the collection of Devnagari alphabet signs. The signers were both male and female with different age groups. All Devnagari signs are captured with a uniform background and with good lighting conditions.

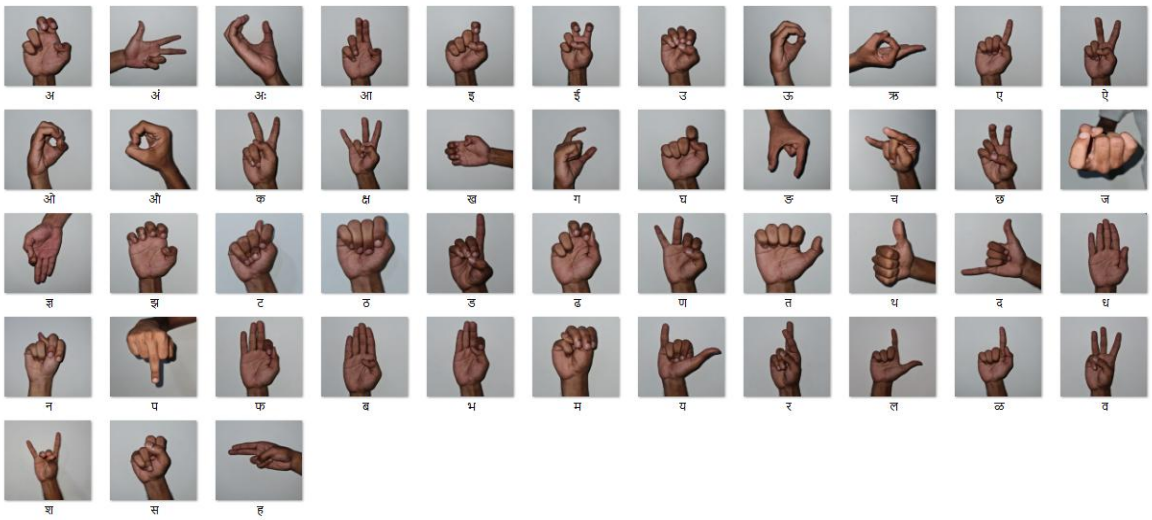

Fig. 2. Devnagari Alphabet Signs

3.2. Image Augmentation

Image augmentation is a technique used in computer vision & machine learning to artificially expand the size of a training dataset by applying various transformations to existing images. It enhances the performance and generalization capability of deep learning models by introducing variations in the training data, effectively mitigating overfitting and improving model robustness. In proposed system rotation clockwise or counterclockwise following image augmentation is applied by rotating images clockwise or counterclockwise by the specified angles, scaling transformations to the images by resizing them, and Shearing images by skewing them horizontally.

3.3. Segmentation and Edge Detection:

In the preprocessing stage, a series of systematic steps were employed to enhance the images for segmentation and edge detection. Initially, the images were converted from the RGB color space to the YCrCb color space. This conversion facilitates better isolation and characterization of skin tones. Following this, precise lower and upper thresholds for skin color within the YCrCb color space were defined as (0, 138, 67) and (255, 173, 133) respectively. These thresholds delineated a specific range of skin tones present in the images. Subsequently, a color thresholding technique was applied to create a binary mask. Pixels falling within the specified skin color range were assigned a value of white (255), while pixels outside the range were assigned black (0). To refine the resulting mask and improve its accuracy, morphological operations were performed. These included an opening operation to eliminate small noise regions and smooth the edges of the mask, enhancing segmentation quality. Additionally, a closing operation was applied to fill in any small holes present within the segmented skin region, ensuring a more complete representation. Finally, the resulting mask was applied to the original image, resulting in the generation of a segmented image wherein the hand gestures were accurately isolated from the background. This segmented image serves as the basis for further analysis and interpretation of Devnagari Sign language gestures, laying the groundwork for subsequent stages of gesture recognition and analysis. Through this systematic approach, the process of preprocessing Devnagari Sign images for segmentation and edge detection was effectively executed, contributing to advancements in sign language recognition technologies.

4. DEEP LEARNING MODELS

This section extensively explores the structures of pre-trained deep models alongside the custom three-layered CNN model utilized for extracting features and classifying Devnagari signs.

4.1 INCEPTIONV3

Prior to the advent of inception models, the primary approach to enhancing neural networks involved

increasing both the depth and width of the model, resulting in significant computational complexity, susceptibility to overfitting due to limited training data, and rapid gradient disappearance. In an attempt to mitigate these issues, the first iteration of the inception model, InceptionV1, opted to expand the model's width rather than its depth. This was achieved by incorporating convolutional filters of various sizes to capture both local and global image information. Furthermore, the integration of two auxiliary classifiers positioned at the output of inception modules within softmax layers was intended to tackle the vanishing gradient problem inherent in the model's structure. InceptionV2 subsequently enhanced this strategy by decomposing large convolutions into smaller and asymmetrical ones, broadening the filter banks to mitigate representation bottlenecks, and decreasing the grid size through parallel pooling, convolution, and concatenation.

In 2015, the InceptionV3 model was introduced with 42 layers and demonstrating a lower error rate compared to its predecessors. It introduced several enhancements, including factorizing 7 × 7 convolutions, employing RMSProp as an optimizer, applying batch normalization to the side layer of the network containing auxiliary classifiers, and implementing label smoothing regularization to reduce the model's confidence in its predictions and mitigate overfitting. This paper worked on InceptionV3 due to its widespread popularity and impressive performance [20].

4.2 RESNET50

ResNet-50 represents a milestone in the evolution of convolutional neural networks (CNNs) by introducing a novel architectural design that enables training of significantly deeper networks. With its 48 convolutional layers, ResNet-50 surpasses previous architectures in depth, facilitating the learning of intricate hierarchical features from input images. The incorporation of skip connections is a key innovation, allowing gradients to propagate more effectively during training by mitigating the vanishing gradient problem. By integrating the original input with the output of convolutional blocks, ResNet-50 enables the training of over 150-layer networks while maintaining performance and convergence. This architectural breakthrough has had a profound impact across multitude applications, encompassing tasks such as image recognition, object detection, and semantic segmentation. ResNet-50's depth and skip connections have led to state-of-the-art performance on benchmark datasets, demonstrating its effectiveness in learning complex visual representations. Moreover, ResNet-50's success has inspired further research in designing deeper and more efficient neural networks, fueling advancements in deep learning and pushing the boundaries of what is achievable in computer vision applications [23,24].

4.3 VGG16

VGG, a convolutional neural network devised by Karen Simonyan and Andrew Zisserman, its name derives from the Visual Geometry Group at Oxford University, their research group. The VGG architecture accepts RGB images with dimensions of 224x224 pixels as input. Before inputting images into the VGG convolutional network, the mean RGB value across all training set images is calculated and utilized for preprocessing. Convolutional operations within the network employ either 3x3 or 1x1 filters with fixed strides. There are various VGG models, ranging from 11 to 19 layers in length. All VGG configurations adopt block structures. Each VGG block involves a series of convolutional layers followed by a max-pooling layer. Consistently, a 3 × 3 kernel size is employed across all convolutional layers, with a padding size of 1 to maintain output size. Additionally, a max-pooling operation of size 2 × 2 with stride 2 is used to reduce resolution after each block. Every VGG model features two hidden fully connected layers and one output fully connected layer. The proposed research work uses VGG16, encompasses 13 convolutional layers, 5 max-pooling layers, and 3 fully connected layers, totaling 16 layers with adjustable parameters. The number of filters in the initial block starts at 64 and doubles in subsequent blocks until reaching 512. The model concludes with two fully connected hidden layers, each containing 4096 neurons, and an output layer comprising 1000 neurons corresponding to the categories in the Imagenet dataset [21,22].

4.4 ALEXNET

The network architecture consists of a eight layers with weights, where the initial five layers are convolutional and three are fully connected. The ultimate fully connected layer outputs to a 1000-way softmax, generating a distribution across 1000 class labels. The objective is to maximize the multinomial logistic regression objective, equivalent to maximizing the average log-probability of the correct label across training cases under the prediction distribution. In the network design, the kernels of the second, fourth, and fifth Convolutional layers are exclusively linked to those kernel maps within the preceding layer residing on the same GPU, while the third Convolutional layer kernels are interconnected with all kernel maps in the second layer. Fully connected layers establish connections with all neurons in the preceding layer. Response-normalization layers follow the first and

Volume 13, Issue 6
December 2024

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X

second Convolutional layers, and max-pooling layers, as described, succeed both response-normalization layers and the fifth Convolutional layer. The Rectified Linear Unit (ReLU) non-linearity is utilized on output of every Convolutional and fully connected layer. The network initiates with the first Convolutional layer filtering the input image sized at 227×227×3 with 96 kernels sized 11×11×3 and a stride of 4 pixels. Subsequently, the second Convolutional layer processes the output of the first Convolutional layer through 256 kernels sized 5 × 5 × 48. The third, fourth, and fifth Convolutional layers are interconnected without any intermediate pooling or normalization layers. The third Convolutional layer employs 384 kernels sized 3 × 3 × 256 connected to the outputs of the second Convolutional layer. Additionally, the fourth Convolutional layer utilizes 384 kernels sized 3 × 3 × 192, and the fifth Convolutional layer employs 256 kernels sized 3 × 3 × 192. Finally, each fully connected layer consists of 4096 neurons [25].

4.5. Convolutional Neural Network:

Convolutional Neural Networks represent a category of deep learning architectures designed to handle and interpret visual data, rendering them especially adept for endeavors such as object detection, image recognition and classification. CNNs have showcased outstanding efficacy across a spectrum of computer vision applications [5].

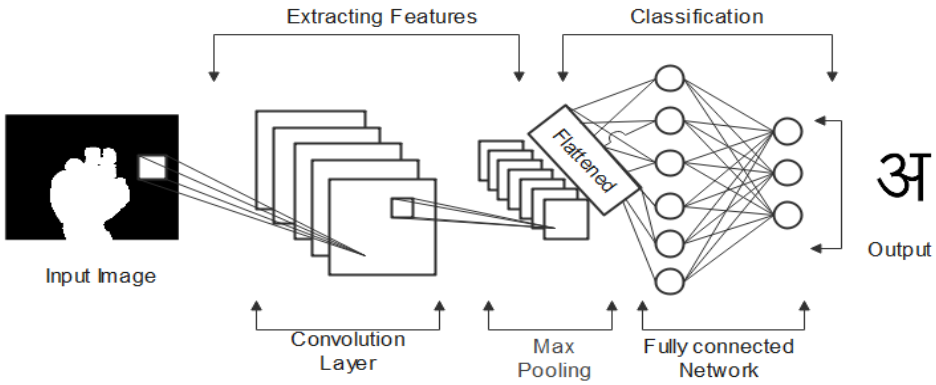

Fig 3. CNN Architecture

The Convolutional Layer initiates with 64 filters, each having dimensions of (3, 3) to convolve the input image, aiming to extract localized features as shown in Fig 3. The Rectified Linear Unit (ReLU) activation function, denoted as f(x) = max(0,x) introduces non-linearity by rectifying negative values to zero, facilitating the capture of intricate data patterns. Mathematically, the convolutional operation can be represented as Eq (1):

$$z_{ij} = \sum_{m=0}^{M-1} \cdot \sum_{n=0}^{N-1} x_{(i+m)(j+n)} \cdot w_{mn} + b \quad (1)$$

where z_{ij} denotes the output feature map, $x_{(i+m)(j+n)}$ represents the input image patch, w_{mn} signifies the filter weights, and b denotes the bias term. Subsequently, the Max Pooling Layer is introduced to decrease the spatial dimensions of the output. Utilizing a (2, 2) pooling window, this layer down samples the input by selecting the maximum value within each window, as illustrated by Eq (2):

$$z_{ij} = max\ (x_{(2i)(2j)}, x_{(2i)(2j+1)}, x_{(2i+1)(2j)}, x_{(2i+1)(2j+1)} \quad (2)$$

Following the Convolutional and Max Pooling layers, a Flatten layer is employed to convert the 2D feature maps into a 1D vector, preparing the data for input into the fully connected layers.

The Fully Connected Layers (Dense) comprise two dense layers. The initial dense layer comprises of 256 units, activated by ReLU, represented as $f(x)$ = max(0,x). Mathematically, the operation of the dense layer expressed as Eq(3):

$$z = f(W.x + b\ ) \quad (3)$$

where z denotes the output, x represents the input vector, W signifies the weight matrix, and b denotes the bias vector.

Subsequently, the second dense layer is configured with a number of units equal to the number of classes in the classification task, employing softmax activation, which assigns probabilities to each class, indicating the model's confidence in predicting each class. The softmax function is mathematically defined as Eq (4):

$$P(yi \mid x) = \frac{e^{zi}}{\sum_{j=1}^{N} e^{zj}} \quad (4)$$

Where $P(yi|x)$ represents the probability of class yi given input x, zi denotes the logit corresponding to class yi, and N signifies the total number of classes[4].Structure of proposed CNN is given in table 2.

Volume 13, Issue 6
December 2024

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X

Table 2: Structure of Proposed CNN

Layer type	Output Size	Parameters
Conv2d_1	(62,62,64)	1792
Maxpolling2d_1	(31,31,64)	0
Conv2d_2	(29,29,128)	73856
Maxpolling2d_2	(14,14,128)	0
Conv2d_3	(12,12,256)	295168
Maxpolling2d_3	(6,6,256)	0
Flatten	9216	0
Dense_1	256	2359552
Dense_2	47	12079
Total Parameters	2742447	
Trainable Parameters	2742447	
Non-trainable Parameters	0	

5. EXPERIMENTAL RESULTS

We used a 64-bit architecture laptop Windows 10 operating system, 16 GB of RAM with Intel Core i7-9700, and 3.00 GHz with x-64 based processor. The Devnagari sign recognition model is developed using Jupyter notebook with tensorflow, scikit-learn library for machine learning. The dataset is captured using mobile camera. The dataset comprised 4230 images distributed across 47 classes as 90 images of each alphabet. To train and assess the model, we allocated 80% images for training and 20% images testing. The training process extended over 20 to 100 epochs, employing the Adam optimizer alongside the categorical cross-entropy loss function.

The Devnagari Sign Language Recognition model's performance was assessed using precision, recall, and F1-score metrics.

$$\text{Accuracy}=\frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (5)$$

Here, TP and TN corresponds True Positives and Negatives, FP and FN represents False Positives and Negatives.

Precision: The precision reflecting the percentage of accurately predicted positive instances in relation to the total instances predicted as positive.

$$\text{Precision}=\frac{TP}{(TP + FP)} \quad (6)$$

Recall (Sensitivity): The recall indicating the model's capacity to accurately recognize positive instances amidst all genuine positive instances.

$$\text{Recall}=\frac{TP}{(TP + FN)} \quad (7)$$

F1-Score: The F1-score is represents a balanced measure of the model's performance by calculating the harmonic means of recall and precision.

$$\text{F1-Score} = 2 * \frac{(\text{precision} * \text{recall})}{(\text{precision} + \text{recall})} \quad (8)$$

Proposed system used Adam optimizer to dynamically adjust learning rates per parameter by computing exponentially decaying averages of gradients and squared gradients, corrected for bias. The moments' moving averages (m and v) begin at zero to initialize the optimization. Gradients of the loss function relative to model parameters are computed to guide parameter updates. These averages evolve using exponentially weighted averages, merging historical moments with the current gradient. Bias correction is employed to counteract initialization biases, crucial especially early in optimization. Parameter updates are determined by normalizing the bias-corrected first moment with the square root of the bias-corrected second moment, with a small constant (epsilon) for numerical stability. This iterative process continues until convergence, refining model parameters and optimizing the loss function [29].

The Sparse Categorical Cross entropy loss function is used in proposed system for multi-class classification. It calculates the cross-entropy loss between the true labels and the predicted probabilities. The formula for Sparse Categorical Cross entropy loss function is:

$$L(y\_true, y\_pred) = -\sum(y\_true * \log(y\_pred)) \quad (9)$$

Where: y_true is the true labels in sparse format. y_pred is the predicted probabilities for each class. log denotes the natural logarithm.∑ sums over all classes. Fig 4. shows sample of predication of Devnagari Sign language alphabets

Volume 13, Issue 6
December 2024

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X

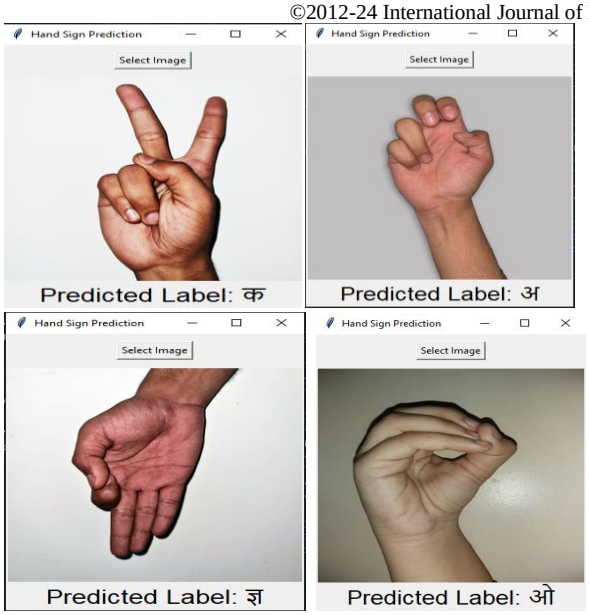

Fig 4. Sample of Predication of Devnagari Sign language alphabets

5.1. Performance Comparison, Limitations and Challenges Faced by Deep Learning Models

The comparative analysis of deep learning models reveals distinct performance characteristics across the models as shown in Table 3 and 4. The Table 3 presents results for four Pre-trained deep models (InceptionV3, ResNet50, AlexNet, and VGG16) alongside a novel Convolutional Neural Network (CNN) architecture proposed for the task. InceptionV3 and ResNet50 demonstrate high accuracy rates exceeding 97% in training and validation phases, indicative of their adeptness in capturing intricate patterns within the training data. These models also showcase robust generalization abilities, as evidenced by their testing accuracies of 94.98% and 91.63% respectively Conversely, AlexNet and VGG16 exhibit relatively lower training accuracies at 91.88% and 91.22% respectively. However, both models demonstrate notable improvements in validation accuracy, suggesting reasonable adaptability to unseen data.

The proposed CNN starts with lower accuracy rates but consistently improves throughout training. By the 200th epoch, it achieves a training accuracy of 95.97% and a validation accuracy of 97.07%, indicating effective learning and potential for competitive performance. The metrics also show an upward trend, highlighting the model's ability to adapt and learn from data. The final testing accuracy of 88.41% indicates good generalization to unseen data. Fig 5, 6, 7, 8, 9 shows Accuracy and Loss Graph of ResNet50, InceptionV3, AlexNet, VGG16, and Customized CNN. Comparative analysis of proposed Devnagari Sign Language with other sign language and classifier is shown in Table 5. To Train deep learning models like InceptionV3, ResNet50, AlexNet, and VGG16 requires to longer training times. Some Classes, such as 'ठ' and 'थ' is misclassified by all the classifier.

Table 3: Evaluation metrics of Pre-Trained deep models

Deep models	Train Accuracy	Train Loss	Val Accuracy	Val Loss	Precision	Recall	F1-Score	Testing Accuracy	Epochs
InceptionV3	97.67	0.04	97.82	0.03	95	95	95	94.98	20
ResNet50	97.49	0.04	97.67	0.04	92	92	91	91.63	20
AlexNet	91.88	0.28	96.12	0.1	89	88	88	88.17	20
VGG16	91.22	0.27	91.52	0.22	91	92	91	91.51	20

Table 4: Evaluation metrics of Proposed Convolutional Neural Network

Deep models	Train Accuracy	Train Loss	Val Accuracy	Val Loss	Precision	Recall	F1-Score	Testing Accuracy	Epochs
Proposed CNN	87.34	0.38	89.16	0.28	84	82	82	82.19	20
	91.73	0.21	94.27	0.14	87	86	86	86.14	50
	95.34	0.1	96.48	0.07	89	88	88	87.93	100
	95.97	0.08	97.07	0.05	89	88	89	88.41	200

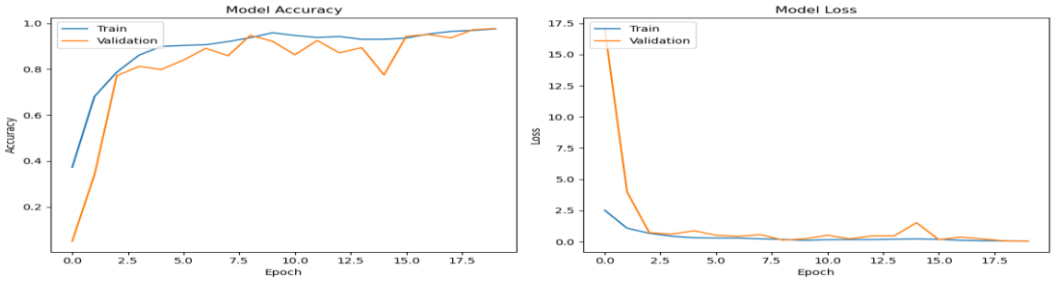

Fig 5. Learning graphs of ResNet-50 for 20 epochs: (a) Accuracy; (b) Loss.

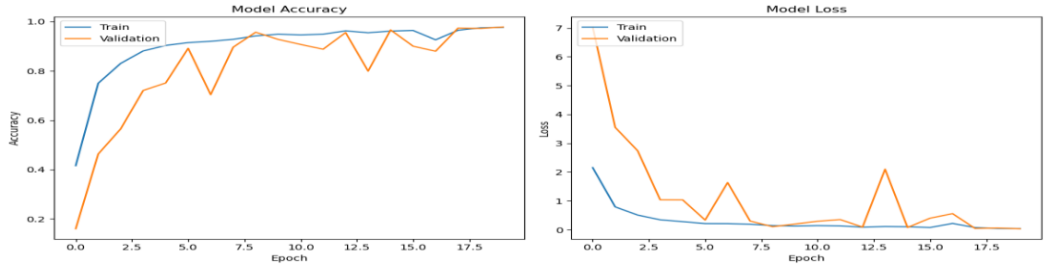

Fig 6. Learning graphs of InceptionV3 for 20 epochs: (a) Accuracy; (b) Loss.

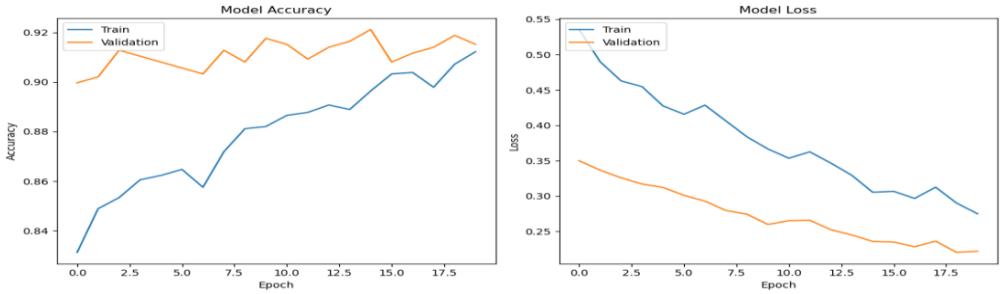

Fig 7. Learning graphs of VGG16 for 20 epochs: (a) Accuracy; (b) Loss.

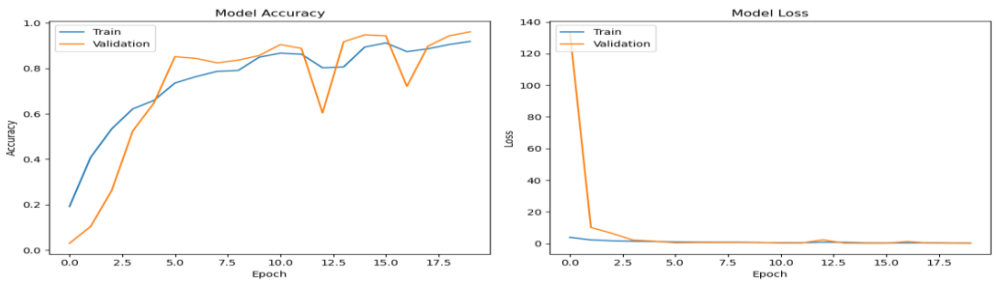

Fig 8. Learning graphs of AlexNet for 20 epochs: (a) Accuracy; (b) Loss.

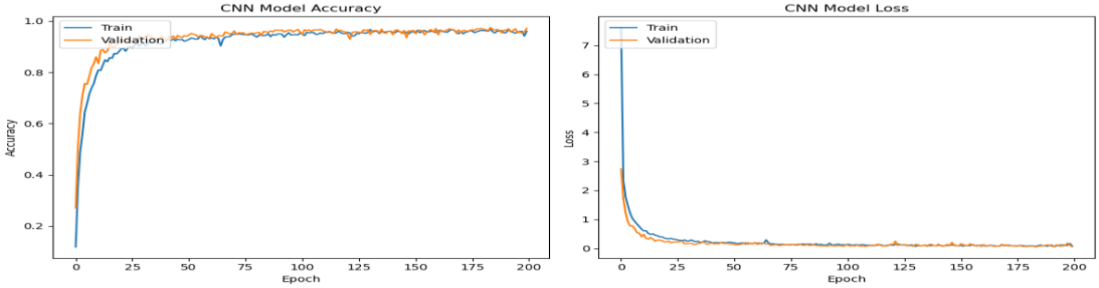

Fig 9. Learning graphs of Customized CNN for 200 epochs: (a) Accuracy; (b) Loss.

Table 5: Comparative Analysis with related studies

Author	Sign Language	No of Alphabets	Technique Used	Accuracy
Das A[30]	American Sign Language	26	Inception V3	90%
A. Hussain[31]	American Sign Language	26	Inception V3	90%
Saini B[22]	Indian Sign Language	26	Inception V3	70.54%
			CNN	83.00%
Proposed Inception V3	Devnagari Sign Language	47	Inception V3	94.98%
Proposed CNN	Devnagari Sign Language	47	CNN	88.41%

6. CONCLUSION AND FUTURE SCOPE

This research marks a significant milestone in the domain of static Devnagari Sign Language (DSL) recognition through a comprehensive evaluation of various deep learning techniques. By employing pre-trained classifiers such as InceptionV3, VGG16, AlexNet, ResNet50, and a custom three-layered CNN model, we have established a robust DSL recognition system. Among these, InceptionV3 emerged as the top performer with a training accuracy of 97.67% and a testing accuracy of 94.98%, providing valuable insights for researchers aiming to select suitable deep learning models for DSL recognition. The contributions of this study lie in its thorough first comparative analysis of multiple deep learning models and the introduction of a customized CNN architecture tailored for recognition of unique dataset with different lightning conditions comprising 47 DSL alphabets.

For future research, several avenues can be explored to further enhance the proposed approach. Refining and fine-tuning the models through extensive experimentation with hyperparameters could improve recognition accuracy and robustness. Expanding the dataset to include a broader range of DSL gestures such as barakhadi is essential for developing more comprehensive and inclusive recognition systems. Additionally, integrating real-time video processing techniques could facilitate the development of practical DSL recognition applications, making them more applicable in real-world scenarios.

Declaration of competing interest

The authors declare that they have no known financial or non-financial competing interests in any material discussed in this paper.

Funding information

No funding was received from any financial organization to conduct this research.

REFERENCES

[1]. Deshpande, A.M., Kalbhor, S.R. (2020). Video-Based Marathi Sign Language Recognition and Text Conversion Using Convolutional Neural Network, Lecture Notes in Electrical Engineering, vol 569. Springer, Singapore.

[2]. Pansare, J., & Ingle, M. (2018). A Real-Time Devnagari Sign Language Recognizer (α-DSLR) for Devnagari Script. In Lecture Notes in Networks and Systems (Vol. 18, pp. 75–84). Springer. https://doi.org/10.1007/978-981-10-6916-1_8

[3]. Pansare, J. R. (n.d.). Real-time Hand Gesture and Sign Language Recognition for Speech Impaired Children.

[4]. Wadhawan, A., & Kumar, P. (2020). Deep learning-based sign language recognition system for static signs. Neural Computing and Applications, 32(12), 7957–7968.

[5]. Sharma, A., Sharma, N., Saxena, Y., Singh, A., & Sadhya, D. (2021). Benchmarking deep neural network approaches for Indian Sign Language recognition. Neural Computing and Applications, 33(12), 6685–6696. https://doi.org/10.1007/s00521-020-05448-8

[6]. Ansari, Z. A., & Harit, G. (n.d.). Nearest neighbour classification of Indian sign language gestures using kinect camera. In Sadhana (Vol. 41, Issue 2).

[7]. Avola, D., Bernardi, M., Cinque, L., Foresti, G. L., & Massaroni, C. (2019). Exploiting Recurrent Neural Networks and Leap Motion Controller for the Recognition of Sign Language and Semaphoric Hand Gestures. IEEE Transactions on Multimedia, 21(1), 234–245. https://doi.org/10.1109/TMM.2018.2856094

[8]. Chong, T. W., & Lee, B. G. (2018). American sign language recognition using leap motion controller with machine learning approach. Sensors (Switzerland), 18(10). https://doi.org/10.3390/s18103554

[9]. Kasapbaşi, A., ELBUSHRA, A. E. A., Omar, A.-H., & Yilmaz, A. (2022). DeepASLR: A CNN based human computer interface for American Sign Language recognition for hearing-impaired individuals. Computer Methods and Programs in Biomedicine Update, 2, 100048.

[10]. Katoch, S., Singh, V., & Tiwary, U. S. (2022). Indian Sign Language recognition system using SURF with SVM and CNN. Array, 14(March), 100141. https://doi.org/10.1016/j.array.2022.100141

[11]. Lee, C. K. M., Ng, K. K. H., Chen, C. H., Lau, H. C. W., Chung, S. Y., & Tsoi, T. (2021). American sign language recognition and training method with recurrent neural network. Expert Systems with Applications, 167. https://doi.org/10.1016/j.eswa.2020.114403

[12]. Oyedotun, O. K., & Khashman, A. (2017). Deep learning in vision-based static hand gesture recognition. Neural Computing and Applications, 28(12), 3941–3951. https://doi.org/10.1007/s00521-016-2294-8

[13]. Pinto, R. F., Borges, C. D. B., Almeida, A. M. A., & Paula, I. C. (2019). Static Hand Gesture Recognition Based on Convolutional Neural Networks. Journal of Electrical and Computer Engineering, 2019. https://doi.org/10.1155/2019/4167890

[14]. D. R. Kothadiya, C. M. Bhatt, A. Rehman, F. S. Alamri and T. Saba, "SignExplainer: An Explainable AI-Enabled Framework for Sign Language Recognition With Ensemble Learning," in IEEE Access, vol. 11, pp. 47410-47419, 2023, doi: 10.1109/ACCESS.2023.3274851.

[15]. Sharma, S., & Singh, S. (2021). Vision-based hand gesture recognition using deep learning for the interpretation of sign language. Expert Systems with Applications, 182. https://doi.org/10.1016/j.eswa.2021.115657

[16]. Tao, W., Leu, M. C., & Yin, Z. (2018). American Sign Language alphabet recognition using Convolutional Neural Networks with multiview augmentation and inference fusion. Engineering Applications of Artificial Intelligence, 76, 202–213. https://doi.org/10.1016/j.engappai.2018.09.006

[17]. D. Naglot and M. Kulkarni, "Real time sign language recognition using the leap motion controller," 2016 International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 2016, pp. 1-5, doi: 10.1109/INVENTIVE.2016.7830097.

[18]. Wangchuk, K., Riyamongkol, P., & Waranusast, R. (2021). Real-time Bhutanese Sign Language digits recognition system using Convolutional Neural Network. ICT Express, 7(2), 215–220. https://doi.org/10.1016/j.icte.2020.08.002

[19]. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, et al. Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2015. p. 1–9. doi:10.1109/CVPR.2015.7298594

[20]. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference oncomputer vision and pattern recognition; 2016. p. 2818–26.

[21]. Le K (2021) An Overview of VGG16 and NiN Models [online]. Available at: https://medium.com/mlearning-ai/anoverview-of-vgg16-and-nin-models-96e4bf398484 (Accessed: 22 June 2022).

[22]. Saini B, Venkatesh D, Chaudhari N, et al. (2023) A comparative analysis of Indian sign language recognition using deep learning models. Forum for Linguistic Studies 5(1): 197–222. DOI: 10.18063/fls.v5i1.1617

[23]. Rathi P, Gupta RK, Agarwal S, and Shukla A (2020) Sign language recognition using ResNet50 deep neural network architecture. Social Science Research Network. DOI: 10.2139/ssrn.3545064.2

[24]. He, K., Zhang, X., Ren, S. and Sun, J. (2016) Deep Residual Learning for Image Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770-778.https://doi.org/10.1109/CVPR.2016.90

[25]. Krizhevsky, A.; Sutskever, I.; Hinton, G.E.Imagenet classification with deep Convolutional neural networks. In Proceedings of the Twenty-Sixth Annual Conference on Neural Information Processing Systems. Lake Tahoe, NY, USA, 3–8 December 2012; pp. 1097–1105.

[26]. Varshney, Saurabh (2016-04-01). "Deafness in India". Indian Journal of Otology. 22 (2): 73. doi:10.4103/0971-7749.182281.

[27]. Daniel Holender, “Synchronic Description Of Present-Day Writing Systems: Some Implications For Reading Research”,Eye Movements from Physiology to Cognition,Elsevier,1987,Pages 397-420.

[28]. https://nhm.gov.in/

[29]. Kingma DP, Ba J. Adam: A method for stochastic optimization. 2014, arXiv preprint arXiv:1412.6980.

[30]. Das, A., Gawde, S., Suratwala, K., & Kalbande, D. (2018). Sign Language Recognition Using Deep Learning on Custom Processed Static Gesture Images. 2018 International Conference on Smart City and Emerging Technology (ICSCET). doi:10.1109/icscet.2018.8537248.

[31]. A. Hussain, S. U. Amin, M. Fayaz and S. Seo, "An efficient and robust hand gesture recognition system of sign language employing finetuned inception-v3 and efficientnet-b0 network," Computer Systems Science and Engineering, vol. 46, no.3, pp. 3509–3525, 2023.

AUTHOR PROFILES

Deepali Naglot received B.E. Degree in Computer Engineering from Pimpri Chinchwad College of Engineering, Pune, India and done M.Tech. in computer science from Vishwakarma Institute of Technology, Pune, India. She is currently pursuing Ph.D. from MGM University, India in Computer Science & Engineering. Currently, she is an Assistant Professor at Jawaharlal Nehru Engineering College, MGM University, Chhatrapati Sambhajinagar, India. Her research interests include Data Science, Image Processing and Machine learning.

Deepa Deshpande received Ph.D. degree in Computer Engineering from Shri Guru Gobind Singhji Institute of Engineering and Technology, Nanded, India and done M.Tech. in Computers from College of Engineering Pune, India. She is currently working as the Professor and Head of Department of Computer Science and Engineering at Jawaharlal Nehru Engineering College, MGM University, Chhatrapati Sambhajinagar, India. Her research areas are Data Mining and Machine Learning.