**Abstract**

This paper presents a deep learning model which can recognize static hand gestures from the Indian Sign Language (ISL). This system model achieves 91.3% accuracy.

The proposed system overcomes the excessive pre-processing of the input image. Using this system the end user can easily communicate with deaf and mute people of our community without any difficulty.

The methodology consists of three phases, training, testing and recognition. The model is trained and tested on 9445 images belonging to 26 different classes (English Alphabets). The training and testing images are augmented before feeding to the model to create it more robust. The system takes a single frame as an input which is a RGB image of size 120x120, which is passed through different layers of the model which recognizes the alphabet according to the trained weights and activations and returns the class of the input image. The model used in this research is a Convolutional Neural Network (CNN). The model successfully classifies all the 26 alphabets of English language.

**Keywords:** Convolutional Neural Network (CNN), Static hand gestures, deaf and mute people, Indian Sign Language (ISL).

****

**Problem Statement**

Socializing is an important aspect of human life, without which one may develop many mental problems. Communication is the most important part of socializing, which certain groups (deaf and mute people) of our society are depraved of. Since they use sign language to communicate, one could either learn the sign language or hire a sign language interpreter (human) which is neither feasible nor an optimal solution. The lack of communication creates a social barrier between the normal population and the deaf and mute people. To overcome this barrier we created this system which recognizes the sign language and helps in communicating with the deaf and mute people at a negligible cost and a self sufficient method.

**Latest Literature**

There have been sufficient works in sign language detection using machine learning with a few works using deep learning.

In Rachana Patil [1] ICACC-2021 a CNN is trained on grey-scale images after performing certain preprocessing techniques like noise elimination, morphological transformation and blurring to reduce the size. The model takes in a 28x28 image and processes it with ReLU (rectified linear unit) activation function. To prevent loss saturation reducing learning rate is used. The system is trained on two optimization algorithms SGD (stochastic gradient descent) and Adam. It was observed that the SGD model performed better at classifying the 10 sign language digits.

In Kartik Shenoy [2] IEEE – 43488 use of external hardware like gloves or the Microsoft kinect sensor, rather it uses just a Smartphone. The system classifies 33 sign including 23 static alphabets and 10 digits. The system classifies static as well as gestures successfully. All the computations are performed on a server. Before transmitting the image to the server it is pre-processed. The pre-processing involves face removal, stabilization, skin color segmentation and noise reduction. Features from the contoured hand are extracted using Grid-based fragmentation, which creates clusters representing different orientations of hand poses. There is more than one cluster for the same hand poses. K-Nearest Neighbors algorithm is used for classification.

In Yogeshwar I. Rokade et al. [3] system many pre-processing steps are performed before feature extraction and classification of the input image. The proposed system classifies 17 alphabets of the Indian Sign Language (ISL) with an accuracy rate of 94.37% using Artificial Neural Network (ANN) and 92.12% using Support Vector Machine (SVM). A 320x240 image is fed as an input on which hand segmentation is performed for skin detection. Then filtering and various morphological operations are performed to reduce noise. From the resulting binary image, containing shape of the hand, features are extracted which are further fed to an ANN for classification.

Indian Sign Language Recognition System (2023) - written by Rupali Kadwade, Akanksha Tangade, Neha Pakhare [4] is a deep learning system for sign language prediction that takes a RGB image as an input and classifies it into one of the 36 categories, including alphabets and numbers. The system returns text as well as audio as an output. A Convolutional Nural Network (CNN) is used for classification.

Sukanya L, Tharun E, Anup Raj G, Shreyas Singh T, and Srinivas S [5] proposed a computationally simple Convolutional Neural Network with very few pre-processing of the input image. The image is converted to grey scale after compressing to reduce the load on the machine on which the model is trained. Model is trained on 4628 images, though the accuracy achieved is not high, the main purpose of this paper was to show that convolutional neural network can be used for sign language detection.

**Proposed System**

The proposed system is a Convolutional Neural Network with an input image size of 120x120. The model consists of multiple Conv2D and MaxPooling2D layer layers in such an arrangement that helps to extract most of the patters present in the image. The model does not require any image pre-processing due to which the complexity and computation time for the recognition of the input image has been reduced substantially.

In the initial layers ‘same’ padding is used to retain more spatial information from the edges of the image. We have used ‘he\_normal’ kernel initializer as it works best with ‘ReLU’ (rectified linear unit) activation function used in our model. To increase the computation speed and reduce the internal covariate shift of the inputs between different layers we have used batch normalization. To prevent the model from over-fitting two dropout layers have been used between the dense layers of the model having a 10% and 20% dropout rate respectively.

The model is optimized using Adam optimizer with an initial learning rate of 0.01. To prevent the model’s loss from reaching its saturation point we have used learning rate decay by a factor of 0.5, this helps in reducing the learning rate if the model’s loss does not reduce for two consecutive epochs. The final learning rate reached to 0.0025.

We were able to achieve 91.3% accuracy with 92.2% precision and 90.82% recall.

Conv2D:

This layer creates a convolution kernel that is convolved with the input and gives a tensor of outputs.

Output = floor value ((input shape – filter size + 2 \* padding) / stride) + 1

MaxPooling2D:

This layer down-samples the input image along its height and width by taking the maximum value over an input window for all the channels.

Output = floor value ((input shape – pool size) / stride) + 1

Batch Normalization:

It makes the training of neural network faster and more stable by normalizing the output mean close to 0 and standard deviation close to 1. It has a regularization effect. It add some noise to the z (w\*x + b) values.

z^~ = gamma \* z + beta

Dense:

This layer calculates the activation of dot product of input and weights plus bias.

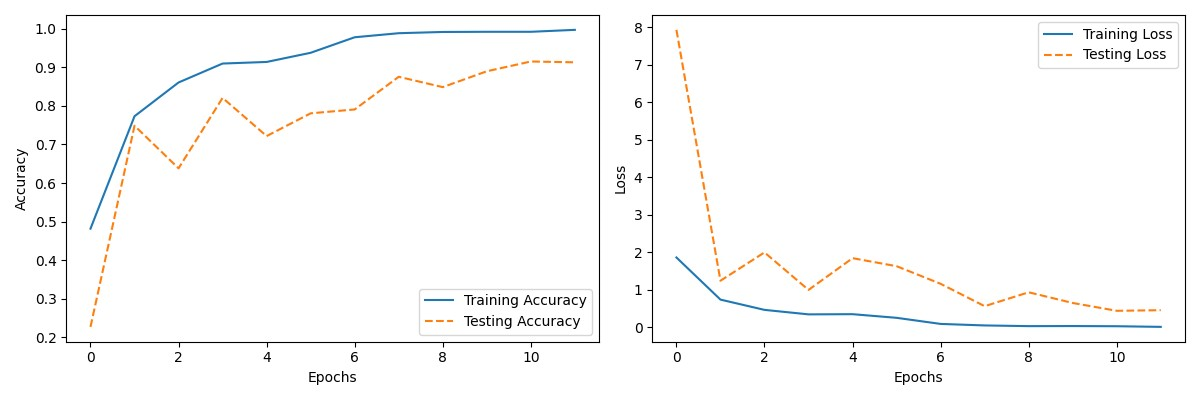
Output = activation ((dot (input, weight matrix) + bias)

Dropout:

This layer randomly set the specified percentage of the input neurons to 0 to prevent over-fitting while training the model.

Experimental Details:

Training and testing graph of accuracy and loss.



Model Architecture

Model: "sequential"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Parameters

===========================================================================

Conv2d (None, 120, 120, 64) 4864

Max\_pooling2d (None, 40, 40, 64) 0

Conv2d\_1 (None, 40, 40, 128) 204928

Conv2d\_2 (None, 40, 40, 128) 147584

Max\_pooling2d\_1 (None, 13, 13, 128) 0

Conv2d\_3 (None, 9, 9, 256) 819456

Batch Normalization (None, 9, 9, 256) 1024

Conv2d\_4 (None, 5, 5, 256) 1638656

Max\_pooling2d\_2 (None, 2, 2, 256) 0

Batch Normalization\_1 (None, 2, 2, 256) 1024

Flatten (None, 1024) 0

Dense (None, 250) 256250

Dropout (None, 250) 0

Dense\_1 (None, 250) 62750

Dropout\_1 (None, 250) 0

Dense\_2 (None, 26) 6526

===========================================================================

Total parameters: 3,143,062

Trainable parameters: 3,142,038

Non-trainable parameters: 1,024

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Conclusion**

Understanding the sign language is a difficult task which prevents us to communicate with the deaf and mute people. The proposed system overcomes this problem by recognizing the signs. The system does not require any pre-processing of the input image which reduces the complexity and time to recognize the signs. The Convolutional Neural Network recognizes all the 26 signs of the Indian Sign Language with an accuracy of 91.3%. This system eliminates the use of hardware devices like gloves or Microsoft Kinect Sensor for sign recognition.

**References**

[1] Rachana Patil, Vivek Patil, Abhishek Bahuguna, and Mr. Gaurav Datkhile, (2021) “Indian Sign Language Recognition using Convolutional Neural Network”.

[4] Indian Sign Language Recognition System(2023) - written by Rupali Kadwade, Akanksha Tangade, Neha Pakhare

[5] Sukanya L, Tharun E, Anup Raj G, Shreyas Singh T, and Srinivas S “Indian Sign Language Recognition using Convolutional Neural Network” E3S Web of Conferences 391, 01058 (2023)

[6] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton “ImageNetClassificationwithDeepConvolutional NeuralNetworks”