IJIRAE:: International Journal of Innovative Research in Advanced Engineering ISSN: 2349-2163
Volume 11, Issue 12, December 2024 https://www.ijirae.com/archives

Hand Sign Detection for the hearing and speech Impaired using Convolutional Neural Network

Adil Perwez
Dept. of Information Technology,
Greater Noida Institute of Technology (Engg. Institute),
Greater Noida, INDIA
adilperwez092@gmail.com
Akhilesh Kr.Sharma
Dept. of Information Technology,
Greater Noida Institute of Technology (Engg. Institute),
Greater Noida, INDIA
akhileshkrsharma11@gmail.com
Amit Kr.Pandey
Dept. of Information Technology,
Greater Noida Institute of Technology (Engg. Institute),
Greater Noida, INDIA
1603amitpandey@gmail.com
Archna T.Singh
Dept. of Information Technology,
Greater Noida Institute of Technology (Engg. Institute),
Greater Noida, INDIA
Dr.Vikas Singhal
Professor & Head,Dept. of Information Technology,
Greater Noida Institute of Technology (Engg. Institute),
Greater Noida, INDIA
hod.it@gniot.net.in
Dr.Ajay Sahu
Professor, Dept. of Information Technology,
Greater Noida Institute of Technology (Engg. Institute),
Greater Noida, INDIA
ajaykrsahu.it@gniot.net.in

Publication History
Manuscript Reference No: IJIRAE/RS/Vol.11/Issue12/DCAE10088
Research Article | Open Access | Double-Blind Peer-Reviewed | Article ID: IJIRAE/RS/Vol.11/Issue12/DCAE10088
Received:10,November 2024,Revised: 21,November 2024,Accepted: 30, November 2024, Published Online: 13,December 2024. https://www.ijirae.com/volumes/Vol11/iss-12/09.DCAE10088.pdf
Article Citation: Adil, Akhilesh, Amit, Archna, Dr.Vikas, Dr.Ajay(2024), Hand Sign Detection for the hearing and speech Impaired using Convolutional Neural Network. IJIRAE:: International Journal of Innovative Research in Advanced Engineering,Volume 11, Issue 11 of 2024 pages 888-893
Doi:> https://doi.org/10.26562/ijirae.2024.v1112.09
BibTeX Key: Adil@2024Hand

Abstract: Learning and utilizing sign languages to communicate with deaf and mute persons has been easier because to developments in computer vision technology. The development of a global platform for communication in a variety of sign languages is an attractive area of research. In this research, we introduce a Deep Learning-based method that uses an image as input to recognize special sign in Indian Sign Language. When the user enters digits 0 through 9, the system can guess their signs. The Convolutional Neural Network's training time and storage needs are effectively decreased by using image processing to transform RGB data to grayscale images. The experiment's goal is to identify a less sophisticated deep learning architecture and image processing is used to implement the system on embedded single-board computers or mobile applications. The database is trained from scratch using small networks like LeNet-5 and AlexNet, as well as deeper networks like Vgg16 and MobileNet v2. Comparing the accuracy of recognition is covered in the study. A dropout layer that improved training and testing accuracy to 91.37% and 87.5%, respectively, is one of just 10 layers that comprise the final design that was selected.

Keywords: recognition of hand gestures, sign language, and convolutional neural networks

I. INTRODUCTION

This research focuses on American Sign Language (ASL), which is mostly used by people with hearing impairments in North America, even though there are other sign languages used throughout the world. For efficient communication, ASL uses both hand gestures and face expressions.

Despite its effectiveness, ASL users often encounter difficulties when interacting with those unfamiliar with the language. This communication gap underscores the urgent need for a system that can interpret ASL in real-time. Such a system would not only aid people with hearing loss but also promote smoother interactions with the hearing population. Recognizing ASL signs and expressions is a complex task that traditional algorithms struggle to handle effectively. However, recent technological advancements, particularly in (CNNs), offer promising solutions. CNNs, a type of deep learning algorithm, are designed to process they are well qualified to identify the complex movements in sign language because they can decipher and interpret visual information. With this paper we thoroughly analyze various neural network models to evaluate their effectiveness in recognizing ASL. Our goal is to identify models that balance high recognition accuracy with efficient operation on lower-end devices and smartphones. This balance is crucial for developing a portable, real-time sign language recognition system that can be used anywhere, anytime. By ensuring these models can function on a range of devices, we aim to make sign language interpretation more accessible and practical for everyday use.

II. RELATED WORK

Research on gesture recognition is a fast-moving, cutthroat field that includes a variety of strategies, including the use of sensory hardware. On the other hand, hardware-based solutions can be more expensive and less useful in daily situations. Because of this, scientists are concentrating on computer vision methods to attain the maximum accuracy of recognition. The effectiveness of several vision-based and glove-based sign language identification techniques was contrasted. [5] Deep learning methods are preferred because of their effectiveness in extracting features from unprocessed data. The combination of CNN layers and the depth of Convolutional Neural Networks (CNNs) can have a substantial impact on the system's performance. In order to recognize signs from mobile phone selfie videos, gesture recognition modules have been developed [6]. Five participants performed 200 signs from various viewing angles in a newly produced dataset for Indian Sign Language.Each of the four convolution layers in the CNN architecture has a variable filter size, which improves accuracy and performance. The advantages of max and mean pooling are combined in a stochastic pooling layer to improve feature extraction. Although this method gives deaf and mute users a way to connect visually, it can make it more difficult to do the signals because the user must hold the phone in one hand.In order to improve model accuracy, skin detection techniques are useful for eliminating superfluous backdrops [7]. The skin detection method lengthens the recognition time while simultaneously lowering image noise. Although using quick GPUs can help with this problem, the system would cost more. The purpose of skin models is to mitigate the effects of uneven illumination [8]. The skin model then uses the Gaussian Mixture Model to create a binary picture after adjusting hand orientation, which further improves the CNN's validation accuracy. As interest in sign language recognition research grows, more researchers and students are being urged to create their own architectures OF CNN or employ the transfer learning methodology. The need for the Indian Sign Language (ISL) database is therefore increasing. A number of computer vision researchers and labs have opened up their datasets to the public, encouraging creativity and the creation of new solutions [9].

III. SUGGESTED GESTURE RECOGNITION METHOD

Convolutional Neural Networks (CNNs) are being used by researchers more and more to extract features from gesture datasets. A comprehensive examination of existing designs and an evaluation of available alternatives within various constraints guide our approach. We propose a design that integrates image pre-processing and deep learning to achieve our desired objectives. This user-friendly system enables individuals to effortlessly capture the photo of a hand sign and input it into the system. The CNNs handle the feature extraction process, ensuring a seamless user experience.

A. System Description

A multilayer perceptron is a good way to characterize the neural network model utilized in this work. Fig. 1 displays the system's block diagram. Each block's description is provided below.

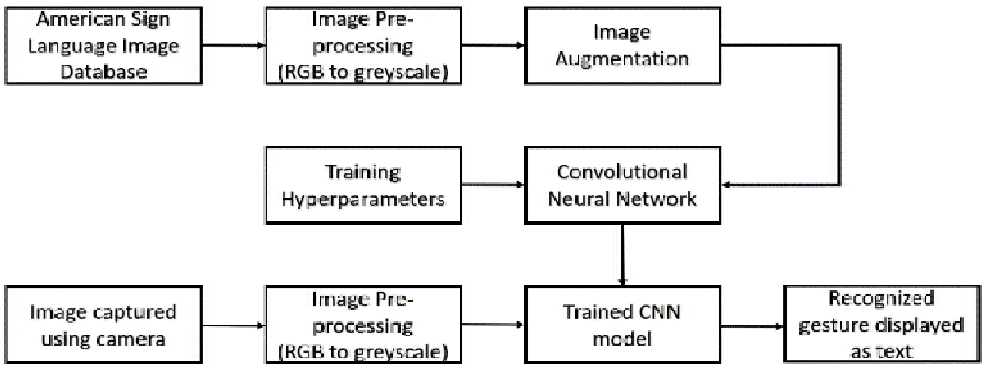

Figure 1. System Block Diagram

- **Image Database**: The databases with these types of pictures of various hand signals. These photos were taken from several users and are repeated several times. There may be variations in the image resolution. For Indian Sign Language, there are various databases available.
- **Image Pre-processing:** Using raw images for training can result in suboptimal performance. Therefore, applying straightforward image processing techniques can help achieve maximum accuracy.
- For instance, converting RGB images to grayscale can decrease training time and power consumption. Additionally, these pre-processing steps can effectively remove noise from the images.

- **Image Augmentation**: For small databases, data augmentation can be highly beneficial. This process involves various image operations to enhance the dataset. Some common techniques include:
- **Mirroring**: Flipping the image horizontally.
- **Snipping:** Process of cutting out specific portions of an image.
- **Rotation, Shearing, Local Warping**: Applying transformations to alter the image's orientation and shape.
- **Color Shifting**: Modifying pixel values in an RGB dataset to change the image's color balance.
- By implementing these methods, the dataset can be effectively expanded, improving the model's ability to generalize and perform accurately.
- **CNN Training & Training Options**: For this project, deep learning techniques are used. Certain training parameters are set before any CNN architecture is used to train the database. These variables include the learning rate, the number of epochs, and the maximum batch size.
- **Image Acquisition:** Any type of camera, including a basic webcam of laptops, can be utilized to capture the image for recognition. This is because, ultimately, the captured image will be scaled down to match the input size required by the Convolutional Neural Network (CNN). Therefore, a high-resolution camera is not necessary.
- **Display Output:** The recognized sign can be shown as text or delivered through an audio description.

B. Hardware Components

An image of the plant must be taken with a camera in order to supply input to the system. There are no strict specifications; any camera with respectable quality and resolution will work. The resolution of the camera can be kept to a minimum.

C. Software Used

- OpenCV: This is used to capture the images and process it.
- Tensor Flow and Keras: Used to train the plant dataset.
- NumPy: Used to hold the group of images to test the model.

IV. IMPLEMENTATION

A. Database

During the tests, one of the main obstacles we encountered was locating an appropriate and extensive dataset for sign language. Turkey Ankara Ayrancı Anadolu High School made the American Sign Language (ASL) dataset publicly available, thus we choose to use it. There are roughly 205 pictures in each class in this dataset, showing hand gestures for the numbers 0 through 9. The pictures have a resolution of 100x100 pixels and are in RGB format.

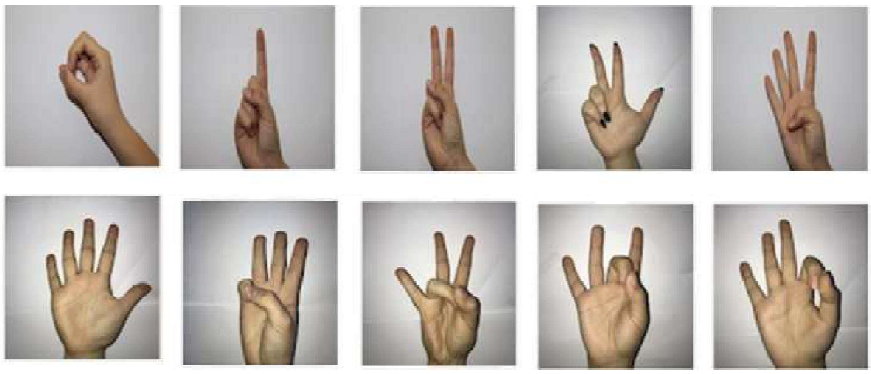

Figure 2. Images used for Indian Sign Language Dataset [10]

B. Image Preprocessing

To improve the outcomes, we applied and contrasted three image processing methods after the images were converted to grayscale. The first method required a grayscale source image and was called simple thresholding. Using a predetermined threshold value that the programmer has established, this approach divides pixel values into two groups: 0 and 1. A binary image is the result of this technique. Thresholding of Otsu's, which automatically extracts a value of threshhold from the image histogram, was the second method. The basic version of this method, which is used for automatic image thresholding, separates pixels into foreground and background classes by returning a single intensity threshold. When two normal distribution curves are combined, two peaks are created, giving the image a bimodal distribution. The third and last method was a thorough five-stage algorithm called Canny Edge Detection. Noise Reduction (smoothing the image and removing noise with a Gaussian filter), Gradient Calculation (finding the image's intensity gradients), Non-maximum Suppression (removing spurious responses to detection of edges), Double Threshold (finding possible edges), and Hysteresis Thresholding (confirming edge detect using suppressing weak edges or those unrelated to strong edges) are some stages. Despite their implementation, these techniques failed to generate accurate output images for the full dataset. Consequently, we decided to leave the original grayscale image database unaltered.

C. Convolutional Neural Networks

Visual data, such as pictures and movies, can be interpreted by Convolutional Neural Networks (CNNs). When compared to conventional machine learning algorithms, these networks offer higher recognition accuracy due to their exceptional capabilities in feature extraction and data learning.

Applications for CNNs are numerous and include data analysis, medical imaging, robotics, signal processing, and business intelligence. Even with smaller, unaffected datasets, CNNs can produce astounding outcomes.

CNNs are built using a number of layers, which serve as the basic building blocks of deep learning. This architecture uses pre-established hyperparameters to interpret input data and learns from the data to calculate weights and biases. During the learning process, a validation dataset a subset of the original data set aside prior to training is used to assess the model's accuracy or loss after each iteration. The layers that were employed in the experiments are described as follows:

Convolution layer Features are extracted from the input photos by convolution layers. Convolution filters are applied to the images to accomplish this, producing feature maps. Generally speaking, the number of maps that are featured and the number of filters utilized match. These filters frequently take the shape of matrices with resolutions like 3x3, 5x5, and so forth. One of the most used activation functions is ReLU, which is applied to the feature maps before they are sent to the following layer.

Pooling Layer: The pooling layer chooses the maximum or average value of pixel from a collection of image pixel to aid in image reduction. During this operation, the picture regions' pooling areas do not overlap. Especially helpful for reducing computational burdens are pooling layers.

Flatten Layer: After being extracted from the preceding layer, the output nodes are either split or flattened. Every one of these distinct nodes has a weight. The flatten layer is used in this procedure to restructure the array or tensor of nodes.

Dense Layer: The code's provided number of units determines the dense layer's output shape. Applying activation and generating an output, this layer operates similarly to a typical neural network layer.

Dropout Layer: Overfitting is a problem with CNN architecture that occurs when the model is too closely matched to the trained data. During each learning cycle iteration, a technique known as dropout is employed to randomly eliminate selected nodes in order to prevent overfitting.

V. RESULTS

A.Result comparison on obtained for existing CNN architectures

Table I shows the results of using a dataset with ten classes and the transfer learning approach. The Convolutional Neural Networks (CNNs) were given the raw image dataset. The ImageNet database was used to pre-learn the weights used in this procedure. The MobileNetV2 model's final output layer was modified for our particular data. Overfitting is evident from the large difference between the training and validation accuracy. The MobileNetV2 training graph is shown in Figure 3.

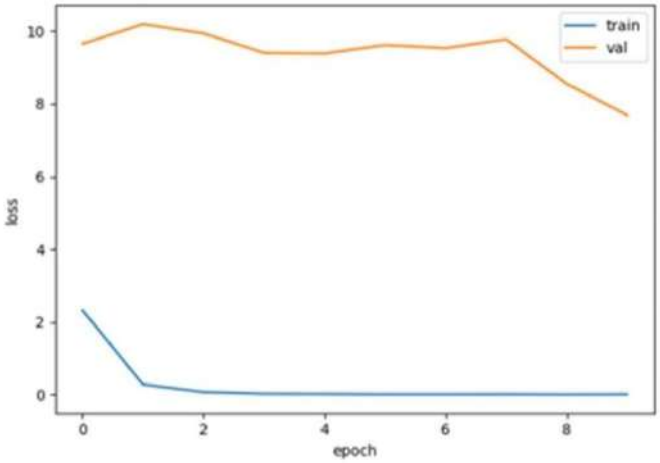

Figure 3. Over-fitting image with MobileNetV2

Over-fitting occurs when a model performs exceptionally well on the training subset but fails to generalize accurately to new, unseen test data. Despite achieving nearly perfect accuracy during training, the model may struggle when evaluated with the validation dataset. After encountering this issue with our initial approach, we decided to train some Convolutional Neural Network (CNN) architectures from scratch on our dataset, starting with randomly initialized weights. The results of training the Vgg16 and LeNet-5 architectures are presented in Table I.

Features of Vgg16 [12]:

- There are 16 layers in all, including a stack of convolution and max-pooling layers, three fully-connected layers, and softmax layers at the end.
- It only employs tiny 3x3 filters with a stride of 1.
- Approximately 138 parameters need to be taught, and 96 MB of RAM per image is needed.

In contrast, LeNet-5 consists of only 5 layers [13]. As Neither the Vgg16 nor the LeNet-5 architectures work well with our data, as seen in Table I. Thus, in order for the CNN design to be effective, it should have a moderate depth, somewhere in between that of Vgg16 and LeNet-5.

Architecture		MobileNetV	2	Vgg16	LeNet-5
Validation	0.1	0.3	0.4	0.3	0.3
Optimize	Adam	SGD(LR =0.01)	SGD(LR =0.01)	SGD(LR=0.01)	SGD(LR =0.01)
Batch length	125	127	33	67	63
Epochs No.	4	4	4	4	4
Training	99.70%	98.97%	89.79%	11.58%	11.20%
Validation	24.35%	9.99%	8.24%	10.64%	8.98%

B. Designed of CNN Model

A model with a amount of moderate layers is created. As seen in Fig. 4, the final architecture performed better on the data. Ten layers make up the CNN, including two dense layers, three convolution layers, three maxpooling layers, flatten, and dropout layers. To prevent overfitting, the dropout layer is used to remove or skip a few randomly chosen nodes. In most experiments, a minor dropout value is of aprox 20% to 50% of neuron is utilized. The validation and training accuracy with varying dropout values displayed.

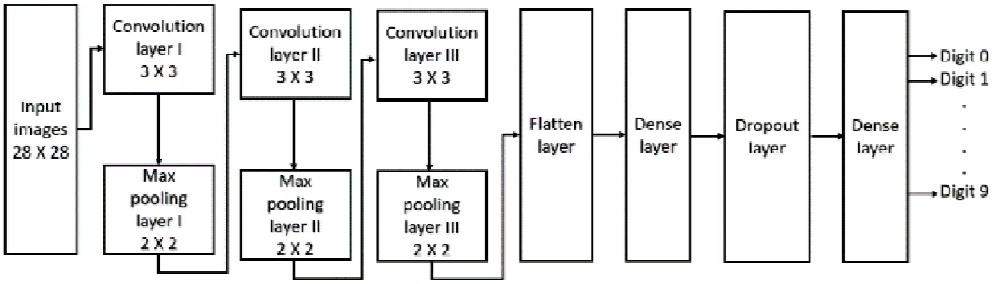

Figure 4. CNN selected Architecture with 10 Layers

The validation accuracy consistently remains around 90% (±2%). Up to this point, test dataset is not separated out or evaluated according to the test accuracy for any models. The training data consists of images fed into the CNN to train the model's weights and biases. After every training cycle, the model's accuracy is confirmed using the validation data, which is an additional component of the entire dataset. The test data is not incorporated in the training process; it is only used to assess the final model. The ultimate accuracy of the CNN architecture as a whole is provided. After that, we separated the test dataset from the rest of the data. We reserved 20 photos for each class, for a total of 200 photos, to test the model. This led to 91.37% training accuracy, 86.30% validation accuracy, and 87.50% testing accuracy, as illustrated in Figure 5. The other settings didn't change.

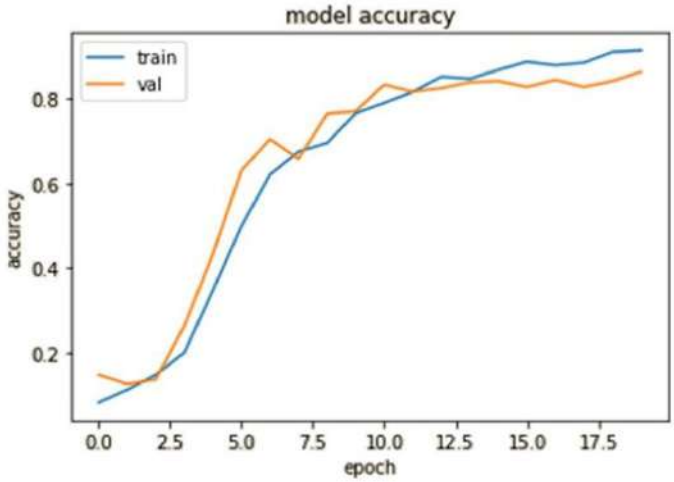

Figure 5. Accuracy model and Loss model for Selected Architecture

As a result, 91.37% accuracy of training and 86.30% accuracy of validation were attained. The resulting confusion matrix is displayed in Table III.

TABLE III. CONFUSION MATRIX OBTAINED

		Predicted labels									
True labels		0	1	2	3	4	5	6	7	8	9
	0	20	-	-	-	-	-	-	-	-	-
	1	1	19	-	-	-	-	-	-	-	-
	2	-	2	17	-	-	-	-	1	-	-
	3	-	1	2	17	-	-	-	-	-	-
	4	1	-	-	-	14	3	1	-	-	1
	5	-	-	-	-	2	18	-	-	-	-
	6	-	-	-	-	1	-	19	-	-	-
	7	-	1	-	-	1	-	1	16	1	-
	8	-	-	-	-	-	-	-	1	18	1
	9	-	-	-	-	1	-	-	-	2	17

VI. CONCLUSION

The data of Indian Sign Language (ISL) in our studies proved to be unsuitable for the transfer learning strategy, which uses pre-trained weights from ImageNet. Results from both smaller and deeper neural network designs were not up to par. This demonstrated that a somewhat deep Convolutional Neural Network (CNN) architecture tailored to this dataset was required. The chosen CNN architecture has an accuracy of 87.5% in the recognition test. Preprocessing images was essential for lowering storage and processing complexity. The model's performance was greatly enhanced by carefully adjusting the training hyperparameters. With its ability to improve identification performance while keeping its size and complexity reasonable, this improved CNN model is particularly beneficial for embedded devices and mobile applications.

