e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:04/April-2024 Impact Factor- 7.868 www.irjmets.com

SIGN LANGUAGE CLASSIFICATION USING DEEP CONVOLUTION NEURAL NETWORK

Manpreet Kaur[*1], Er. Harjasdeep Singh[*2], Er. Nancy Mittal[*3]

[*1]Baba Farid College Of Engineering & Technology, Bathinda, Punjab, India.

[*2]Assistant Professor, Computer Science & Engineering, MIMIT Malout, Punjab, India.

[*3]Assistant Professor, Baba Farid College Of Engineering And Technology, Punjab, India.

DOI : https://www.doi.org/10.56726/IRJMETS53059

ABSTRACT

Sign language is the main form of communication for the speech-and hearing-impaired people. The general public finds it very difficult to understand or comprehend sign language fully.

The creation of a sign language recognition system is necessary to overcome this communication obstacle. Since wearable sensors are the foundation of the majority of sign language identification systems now in use, most people cannot afford the system for identification. We are constructing a sign language convolutional model on the different parameters in this research project. We obtained the sign language-based dataset from Kaggle.The numerical data in the dataset was converted into English letters, with the aid of Matplotlib. Our tests demonstrated the convolutional sign language system's accuracy. Additionally, we merged these methods and determined the models' mean value.

Keywords: Sign Language, Hand Gesture, English Letters, Hearing –Impaired Person, CNN.

I. INTRODUCTION

Communication is way of talk .people can share their thought with each other by verbal or Non-verbal ways. This research base on non-verbal communication .The sign language counts under the non-verbal communication. It is supports to deaf people population .sign language has own grammar rules that permission to users express their ideas as well as emotions by using hand motion, face and lips expression . Moreover, it is major problem for impaired populace .So; it is responsibility of computer science researchers to develop such models that can tackle this issue. Numerous research's had been established to help speech defeat people .But, it is not completely solve. Recent system of sign language is very costly and makes with leverage sensors, colorful gloves. The users do not conveying the message in real life .Therefore, we work on sign language with deep learning .So that, we can remove the obstacles of impaired people life .We prefer the convolution neural network since CNN is useful algorithm of deep learning. It is specially design for image recognition .In CNN algorithm divides the whole image into 2*2 matrixes then apply the filters on that matrixes. The filters are nothing ,it is numbers of convolution layers that extraction edges of object and gives the information about image .The result accuracy of CNN is very better than others deep learning algorithms. The rest of others algorithms of deep learning have some demerits such as computing time, over fitting and loss of information. Therefore, numbers of reasons why we choose CNN to complete us research work.

Sign language is a practical language that uses limb motions and spatial movements to express meaning. It is the most organic method of communication between the deaf and the outer world. The goal of visual sign language production (SLP) is to mechanically transform spoken words into suitable sign language videos. For the Deaf community, accurate and vivid SLP may greatly enhance communication quality. Intermediary terms that correspond to the literal meaning of spoken language are known as signal glosses. here are a pair of elements to our work:(1) Converting verbal sentence gloss sequences into their matching sign posture sequences. (2) Generating skin-based [1]. This research primarily investigates how to solve the aforementioned challenges by utilizing the most recent deep learning technique with RGB-D multimodal input. Our findings may provide useful insights for intelligent systems, tiny displacement behavior identification, and sign language recognition.[3].

II. LITERATURE REVIEW

1. E. Rajajaksmi et .al [2023] the author proposed the methodology of sign language gesture based on novel vision approach that was used for identification of India or Russian impaired sign action. The major goal of this proposed was built the framework for extracting and tracking .Moreover, spatial attribute removal the sign motion was install using 3D deep neural network accompanied by atrous convolution and consecution attribute extraction was accomplished by hiring BI-LSTM model. The result of proposed model recognition framework with hybrid neural network gave better accuracy than state of art-structure.
2. Samiya kabir youme et .al [2023] The writer was worked on sign language .even as, the deaf people are facing the problem when they are communication with other .They are not able to talk with them. The authors developed the vision based model that decodes the sign language into clear text and speech fixed system. All this worked was done by applying the machine learning algorithms and deep learning technique. They divided the dataset into test data or training data 9:1 ratio this job completed with the help of python language, CNN, IOT. In future, they can expand local language for other body language or improve the response time of prototype model.
3. Dhiraj Neupane et .al [2024]They presented a revolutionary system dubbed "Shine," which recognizes the automobile, the registration number, and handicap badges (henceforth referred to as cards, badges, or access badges) using an object recognition algorithm based on deep learning. By collaborating with the main server, the system verifies the driver's eligibility to use available parking spaces. With a mean average precision of 92.16%, our model is predicted to tackle the problem of abusing accessible parking spaces and make a substantial contribution to the efficient and effective administration of parking in metropolitan areas.
4. Arun Prasath Govindan et .al [2022]The author illustrated a reversible convolution neural network model for body language identification .This model ready for to identify the voice signal from shrug language .The accuracy of reversible system is verify by G-CNN,VGG-11/16 model above the training and testing surrounding .They considered two different dataset such as voice input dataset or ROBITA India shrug language to evaluated for determination the greatest accuracy of 97.89%and 94.38% with reversible CNN system than the other model .They suggested that the work will be extended with meta heuristic optimization methods to achieve overall solution.
5. Abdul Mannan et .al [2022] The author did the research on American sign language identification due to the increasing the complexity of intra object similarity. They proposed the 24 alphabets that was used in sign language and that suggested approach was based on deep convolution neural network to identify the sign speech alphabets. Firstly, they used a single layer of deep convolution neural network model that over fits the information. Then they added the two other layers to control this issue. Hence forward, the new research recognizes the sign language motion. The deep convolution neural network model demonstrated the 99.67% accuracy with 24 alphabets.
6. Abu Saleh Musa Miah et .al [2022] The writer represented the novel approach for identifying the Bengali sign speech alphabets numbers of research has done with satisfactory performance by using small database. But there system may fail to achieve the same performance for big database and different type of dataset. The mentioned approach had been evaluated by three different benchmark database likewise, I share-lip, KU-BDSL, and 38BDSL three separated steps were required to get the aim of research 1 segmentation 2 augmentation 3 convocational. By using their different dataset they got 99.60% accuracy.
7. Kujani.T* et .al [2022] The research represented a deep convolution neural network method had been developed for fetching 35 alphabets of Indian language into content in better arrangement make are of hand kinematics. The research was work managed straight forward way and result was approximately 92.85%. In Indian sign language dataset amount 42000 pictures were utilized for execution. By using this research work We can improve another countries gestures and translate the words into speech with IOT technologies.
8. Sanket Dhobale1 et .al [2022] the author demonstrated their for deaf and dump person because it is a general problem for them to communication accompanied by someone. Who do not know the sign language since the sign language is not official language. They tried to solve this problem. They were developed a

model that could detect the sign gesture by using web cam and they were created this model with the help of tensor flow and deep learning. Apart from that they were used the CNN to instruct model. This research had better future scope and they can task move on this research and will get good accuracy.

9. M. Daniel Nareshkumar et .al [2022] This work had investigated a revolutionary deep learning architecture that can rapidly and effectively detect the letters in American Sign Language (ASL) using recently published huge pre-trained picture models. In order to proved that it was feasible to obtain a high degree of classification accuracy on the data and that interpreters may be used in the actual world, the study was concentrated on isolated sign language. The core of this work was the recently suggested MobileNetV2 architecture. It was intended to operate on endpoints such as smart phones and quickly deduce signals (what does it deduce) from pictures. This research proposes an architecture that yields a 98.77% classification accuracy in Indian Sign Language (ISL) and In order to prove that a high degree of classification accuracy can be achieved on the data and that interpreters can be used in the actual world, the study was concentrated on isolated sign language. The core of this work was the recently suggested MobileNetV2 architecture. It was intended to operate on endpoints such as smart phones and quickly deduce signals (what does it deduce) from pictures. This work proposes an architecture that outperforms current state-of-the-art systems, achieving a classification accuracy of 98.77% in both American Sign Language (ASL) and Indian Sign Language (ISL).
10. Yohanssen Pratama, Ester Marbun et.al [2020] In this study, they attempted to gather hand gesture data and identify the genuine hand gestures using a straightforward deep neural network design that they refer to as model E. Our dataset was gathered via kaggle.com and came in the form of American Sign Language (ASL) datasets. They were comparing the accuracy of our model to other existing models, like Alex Net, in order to assess its robustness. They discover that altering the kernel size and number of epochs for every model also produces a distinct outcome. After comparing our model E with the Alex Net model, they discovered that it performs better, with an accuracy of 96.82%. They will test a model that can identify sign language by comparing static to dynamic movements as more study. As a result, the motion that the camera saw could be instantly recognized as a language.

III. METHODOLOGY

Collection of dataset: The dataset is source from the Kaggle website. There are training and testing subsets within the dataset.

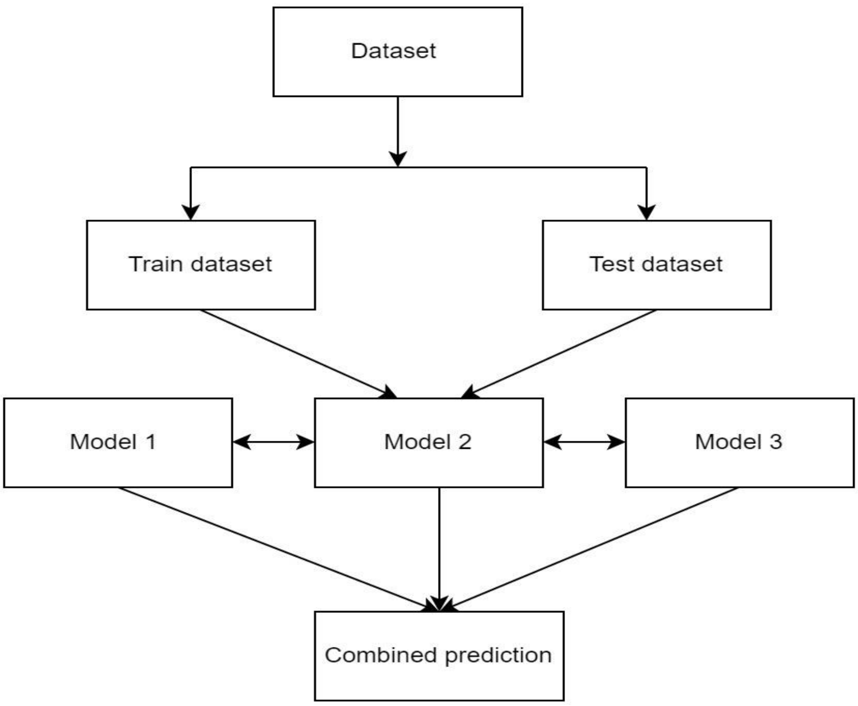

Figure 1: Methodology of purposed work.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:04/April-2024 Impact Factor- 7.868 www.irjmets.com

ALGORITHMS ARE USING IN RESEARCH WORK

Convolution neural network: A particular class of machine learning model known as a convolution neural network (CNN) is a deep learning technique that is particularly well-suited for the analysis of visual input. CNNs, also known as convents, extract features and recognize patterns in pictures using concepts from linear algebra, namely convolution processes. CNNs may be configured to handle audio and other signal data, even if processing pictures is their primary function. CNN was developed using the connection patterns observed in the human brain, namely in the visual cortex, which is essential for the perception and processing of visual inputs. Because the artificial neurons in a CNN are designed to effectively process visual input, these models are capable of understanding whole images. Because CNNs are so effective at acknowledging things, they are frequently employed for computer vision tasks like picture recognition and object detection. Face recognition, self-driving cars, and medical image analysis are examples of common application cases.

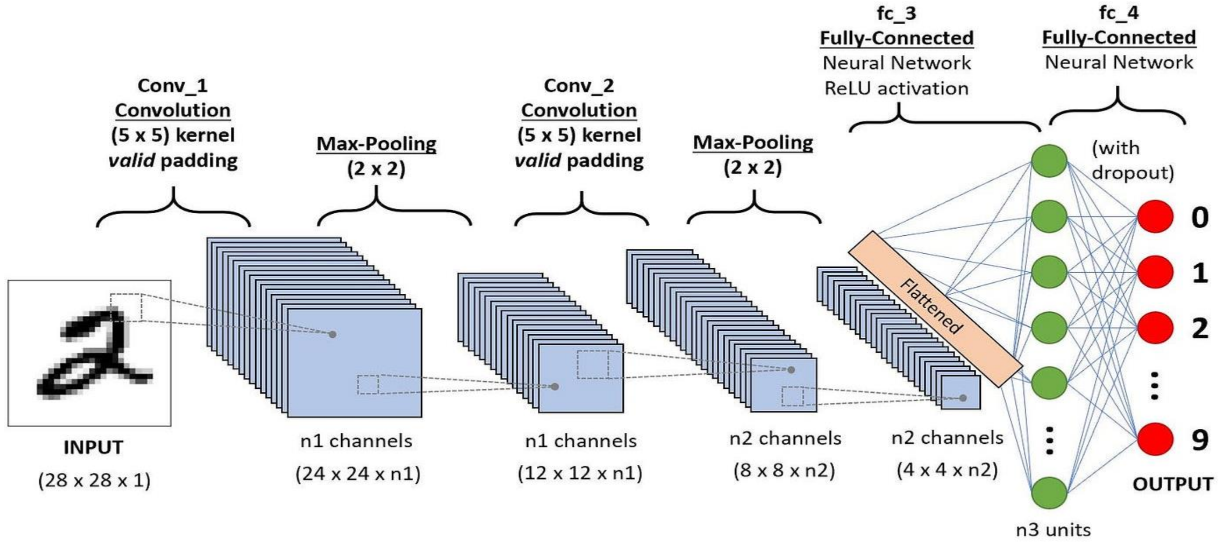

Figure 2: Convolutional neural network

We are utilizing the CNN deep learning method in this study. The CNN method works well for identifying patterns in images so that objects and classes may be identified. In order to determine which parameter yields the best result, we are providing different parameters to CNN1, CNN2, and CNN3.

Relu activation function: In neural networks, the Rectified Linear Unit is a widely used activation function, particularly for deep learning models. It endows the network with non-linearity, allowing it to discern complex patterns within the data. Put differently, the ReLU activation function is defined as follows:

$$f(x) = \mathrm{Max}(0,x)$$

Basically, it just returns the input value x, which is zero otherwise, if it is positive. The ReLU function appears graphically as a linear function with the negative term "rectified" to zero.

Max polling: Convolutional Neural Networks (CNNs) and Max Pooling: Max pooling is a down sampling technique used in convolutional layers of CNNs and deep learning systems. By dividing the input image into a series of non-overlapping rectangular sections, max pooling produces the maximum value for each of these sub-regions. Setting the maximum pooling size or rate in a CNN architecture is referred to as max polling.

Flatten layer: In neural network designs, the "flatten" operation is used to convert multi-dimensional arrays or tensors into one-dimensional vectors, particularly in convolutional neural networks (CNNs). This process is frequently used prior to data flowing into a thick or completely linked layer.

Dense layer: A fully connected layer, sometimes referred to as a dense layer, is an essential part of neural network topologies, which includes deep learning models. Every neuron in a thick layer is linked to every other neuron in the layer above it, creating a dense matrix of connections.

Dropout layer: In neural networks, dropout is a regularization approach that is frequently employed, especially in deep learning models, to increase generalization performance and reduce over fitting. The training process entails "dropping out" a part of neurons in a layer at random. Dropout is commonly used to describe

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:04/April-2024 Impact Factor- 7.868 www.irjmets.com

hidden layers in a network, such as convolutional layers and fully connected (dense) layers. It is a popular and powerful regularization method that has been demonstrated to enhance neural networks' performance across a range of tasks.

Output: A loss function is an equation that quantifies the difference between a model's predicted values and the actual values observed in the training set. It is also known as an objective function or a cost function. In machine learning and optimization, minimizing this loss function is frequently the aim as it enhances the model's performance or accuracy.

In many machine learning applications, including regression, classification, and reinforcement learning, loss functions are essential.

Categorical cross entropy: Categorical cross-entropy is a widely used loss function in classification tasks, especially in multi-class classification problems. The difference between the expected and real label probability distributions is computed.

Optimizer Adam: Neural network training uses the well-liked optimization technique Adam (Adaptive Moment Estimation). It combines the benefits of AdaGrad and RMSProp, two further stochastic gradient descent (SGD) additions. Adam uses the gradients' first and second moments to maintain per-parameter learning rates.

Activation Soft max: In the output layer of neural networks, the soft max activation function is frequently utilized for multi-class classification applications. It compresses a neural network's raw output scores, often known as logits, into a probability distribution spanning several classes. The likelihood that the accompanying class is the proper class is represented by each output value. All of these activities were carried out on CNN2, CNN 3 by changing the parameters alone. Next, we computed the average of these models by combining the accuracy of models 1, model 2, and model 3.

Training Accuracy: Based on data points that the model has been trained on, this statistic assesses the system accuracy on the training dataset. It is determined by contrasting the real labels of the training data points with the predictions made by the model.

Training Accuracy is calculated as

$$\frac{\text{Total number of samples in the training set}}{\text{No of correctly identified sample in the training set}}$$

When a model has a high training accuracy, it can accurately categorize the majority of the training data points. High training accuracy, however, does not ensure excellent performance on unknown data; it is possible that the model has only successfully learned the training instances by memorization.

Validation Accuracy: This statistic assesses the model's performance on an unrelated validation dataset that hasn't been used for training. The validation set's goal is to give a ballpark approximation of the model's performance on fresh, untested data.

Validation Accuracy is calculated as

$$\frac{\text{Total number of samples in the validation set}}{\text{No of successfully identified samples in the validation set}}$$

A high validation accuracy shows that the model is not over fitting to the training set and is instead generalizing effectively to new data. A large discrepancy between the validation and training accuracy might be an indication of over fitting, in which case the model is collecting noise in the training set instead of identifying the underlying patterns.

Training Loss: This statistic calculates the mistake or loss that the model experiences while being optimized during training. It gives an estimate of how well the model fits the training set. Common loss functions in classification tasks include mean squared error (MSE) for regression tasks and cross-entropy loss for categorical classification.

Usually, the average loss over all training samples or batches is used to calculate the training loss. By modifying the model's parameters (weights and biases) during training, optimization methods like gradient descent or its variations aim to minimize this loss.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:04/April-2024 Impact Factor- 7.868 www.irjmets.com

Validation Loss: The model's error or loss on an independent validation dataset—which it was not exposed to during training—is measured by this statistic. The validation set's goal is to give a ballpark approximation of the model's performance on fresh, untested data.

The same loss function that is used to calculate training loss is also used to compute validation loss. It is, however, computed using the validation dataset rather than the training dataset. The validation loss indicates if the model is over fitting or under fitting and helps track how effectively the model generalizes to new data.

IV. RESULTS AND DISCUSSION

We have put into practice the image-processing deep learning method. We used a variety of convolutional neural network techniques in this study.

In this research work we have used convolution neural network algorithm for the prediction of sign language. Three models of convolution neural network were used for task. Each model of CNN have multiple layers of convolution layer, max pooling and dense layers. Summary of CNN model 1 is as shown in the figure 3 filters used in first convolution layers are 32, kernel_size is (3,3), strides=(1, 1), padding used is "valid", activation function is relu, after that max pooling is used with pool size of 2,2. Drop out is 0.2. in the next convolution layer 64 filters were used of size(3, 3)and activation is 'relu'.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 26, 26, 32)        320

 max_pooling2d (MaxPooling2  (None, 13, 13, 32)        0
 D)

 dropout (Dropout)           (None, 13, 13, 32)        0

 conv2d_1 (Conv2D)           (None, 11, 11, 64)        18496

 max_pooling2d_1 (MaxPoolin  (None, 5, 5, 64)          0
 g2D)

 dropout_1 (Dropout)         (None, 5, 5, 64)          0

 conv2d_2 (Conv2D)           (None, 3, 3, 128)         73856

 max_pooling2d_2 (MaxPoolin  (None, 1, 1, 128)         0
 g2D)

 dropout_2 (Dropout)         (None, 1, 1, 128)         0

 flatten (Flatten)           (None, 128)               0

 dense (Dense)               (None, 128)               16512

 dense_1 (Dense)             (None, 25)                3225

=================================================================
Total params: 112409 (439.10 KB)
Trainable params: 112409 (439.10 KB)
Non-trainable params: 0 (0.00 Byte)
```

Figure 3: CNN model 1

The summery of Model 2 of convolution neural network is shown in the figure 4. 32 filters are used in first CNN layers of model 2, filter size is (3, 3) and input_ shape is (28, 28,1), activation function used is ' relu'. In second CNN layer 64 filters were used with filter size of (3, 3) and activation function is used ' relu' . In the 3rd CNN layer of model 2 , 128 filters were used.

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_3 (Conv2D)           (None, 26, 26, 32)        320

 conv2d_4 (Conv2D)           (None, 24, 24, 32)        9248

 max_pooling2d_3 (MaxPoolin  (None, 12, 12, 32)        0
 g2D)

 conv2d_5 (Conv2D)           (None, 10, 10, 64)        18496

 conv2d_6 (Conv2D)           (None, 8, 8, 64)          36928

 conv2d_7 (Conv2D)           (None, 6, 6, 64)          36928

 max_pooling2d_4 (MaxPoolin  (None, 3, 3, 64)          0
 g2D)

 conv2d_8 (Conv2D)           (None, 1, 1, 128)         73856

 conv2d_9 (Conv2D)           (None, 1, 1, 25)          3225

 flatten_1 (Flatten)         (None, 25)                0

 dense_2 (Dense)             (None, 25)                650

=================================================================
Total params: 179651 (701.76 KB)
Trainable params: 179651 (701.76 KB)
Non-trainable params: 0 (0.00 Byte)
```

Figure 4: CNN model 2

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_10 (Conv2D)          (None, 26, 26, 32)        320

 max_pooling2d_5 (MaxPoolin  (None, 13, 13, 32)        0
 g2D)

 dropout_3 (Dropout)         (None, 13, 13, 32)        0

 conv2d_11 (Conv2D)          (None, 11, 11, 64)        18496

 max_pooling2d_6 (MaxPoolin  (None, 5, 5, 64)          0
 g2D)

 dropout_4 (Dropout)         (None, 5, 5, 64)          0

 flatten_2 (Flatten)         (None, 1600)              0

 dense_3 (Dense)             (None, 25)                40025

=================================================================
Total params: 58841 (229.85 KB)
Trainable params: 58841 (229.85 KB)
Non-trainable params: 0 (0.00 Byte)
```

Figure 5: CNN model 3

IRJMETS
e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:04/April-2024 Impact Factor- 7.868 www.irjmets.com

Table 1: Accuries of all CNN model

Algorithms	Accuracy
CNN 1	93.94%
CNN 2	87.40%
CNN 3	91.77%
Purposed ensemble model	94.92%

The accuracy of the CNN model is described in the mention table above. CNN 3 had an accuracy of 91.77%, while CNN 1 and CNN 2 had 93.94% and 87.40% accuracy, respectively. In the end we applied average ensembled method and the average accuracy was 94.92%.

Epochs

It appears that you are discussing "epochs" in relation to neural networks or machine learning. An epoch in machine learning is a single sweep over the training dataset. The data is split up into batches for training, and the model modifies its parameters after each batch. One epoch ends after every batch has been processed. Several epochs are usually done during training in order to enable the model to efficiently identify patterns in the data.

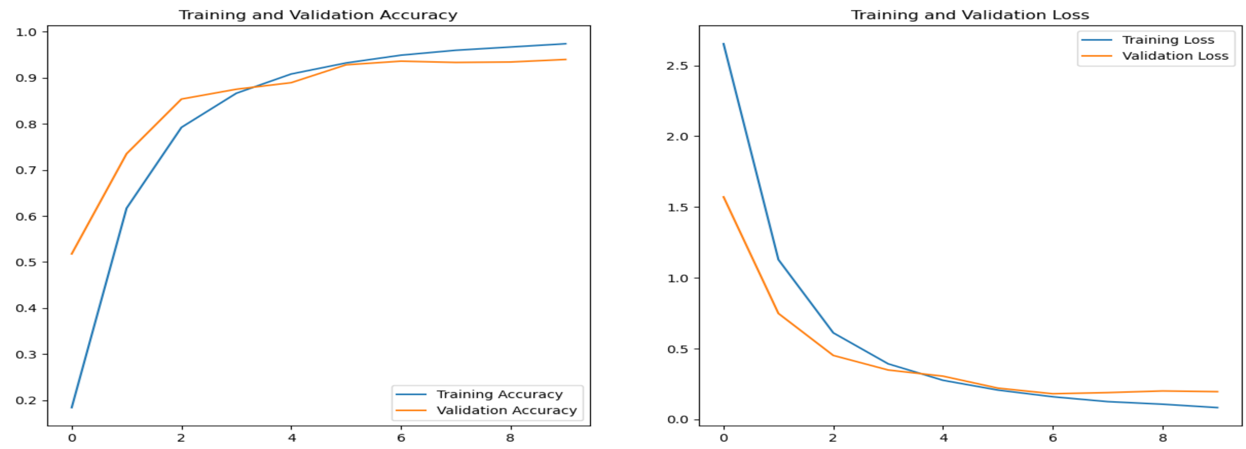

Figure 6:

In the fig 6 represent the training and validation accuracy and fig 7 showed the training and validation loss.

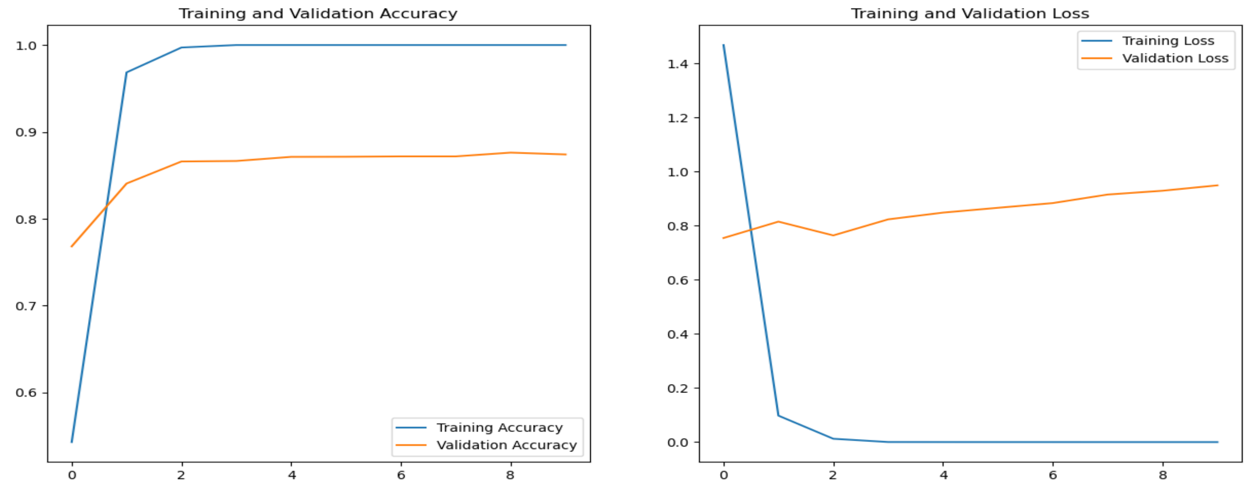

Figure 7:

Above mention diagram illustrated that the training and validation accuracy and training and validation loss

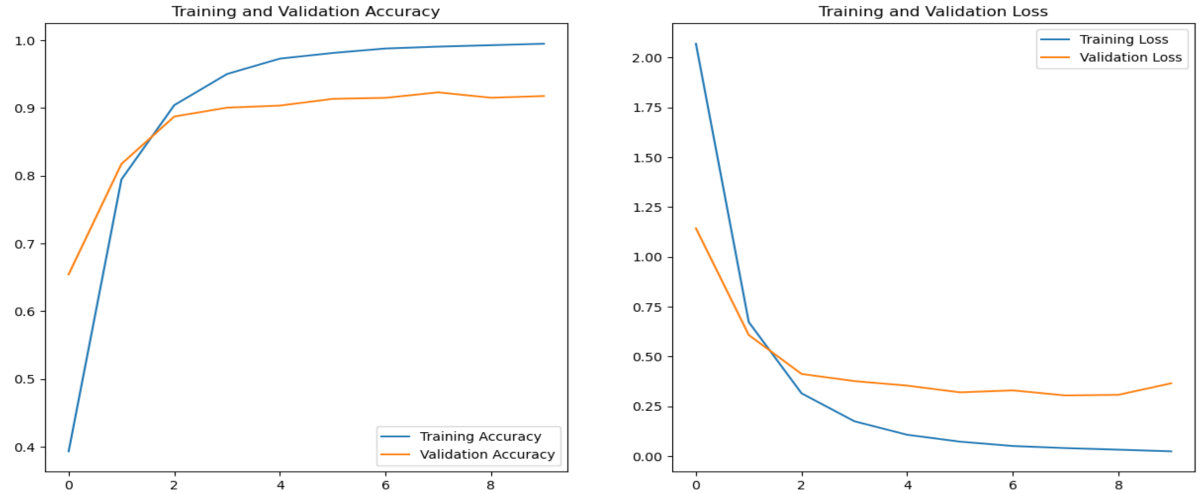

Figure 8:

V. CONCLUSION

In this study we were work on sign language with the help of deep learning algorithm because we know the deep learning approaches are better than machine learning algorithms and work on huge amounts of data .Moreover, in our research work weight and importance is equal .Future models can be assigned weight age according to their performance

VI. REFERENCES

[1] Rajalakshmi, E., Elakkiya, R., Subramaniyaswamy, V., Alexey, L.P., Mikhail, G., Bakaev, M., Kotecha, K., Gabralla, L.A. and Abraham, A., 2023. Multi-semantic discriminative feature learning for sign gesture recognition using hybrid deep neural architecture. IEEE Access, 11, pp.2226-2238..

[2] Youme, S.K., Chowdhury, T.A., Ahamed, H., Abid, M.S., Chowdhury, L. and Mohammed, N., 2021. Generalization of bangla sign language recognition using angular loss functions. IEEE Access, 9, pp.165351-165365.

[3] Neupane, D., Bhattarai, A., Aryal, S., Bouadjenek, M.R., Seok, U. and Seok, J., 2024. Shine: A deep learning-based accessible parking management system. Expert Systems with Applications, 238, p.122205.

[4] Arun prasath govinda,2021, A Reversible Convolutional Neural Network Model for Sign Language Recognition, International Journal of Intelligent Engineering and Systems, Vol.15, No.2,pp164-174.

[5] Mannan, A., Abbasi, A., Javed, A.R., Ahsan, A., Gadekallu, T.R. and Xin, Q., 2022. Hypertuned deep convolutional neural network for sign language recognition. Computational intelligence and neuroscience, 2022.

[6] Miah, A.S.M., Shin, J., Hasan, M.A.M. and Rahim, M.A., 2022. Bensignnet: Bengali sign language alphabet recognition using concatenated segmentation and convolutional neural network. Applied Sciences, 12(8), p.3933.

[7] Kujani.T*, DhilipKumar,2022, Multiple Deep CNN models for Indian Sign Language translation for Person with Verbal Impairment, nternational Journal of Intelligent Systems and Applications in Engineering,10(3), pp 382–389.

[8] Sanket Dhobale1 , Aniket Kandrikar2, International Journal of Advanced Research in Science, Communication and Technology , Real Time Sign Language Detector Using Deep Learning ,pp 182-185, Volume 2, Issue 1, March 2022

[9] M. Daniel Nareshkumar, 2022, A Light-Weight Deep Learning-Based Architecture for Sign Language Classification, Intelligent Automation & Soft Computing , vol.35, no.3 ,pp3501-3515

[10] Yohanssen Pratama, Ester Marbun, 2o2o ,Deep convolutional neural network for hand sign language recognition using model E , Bulletin of Electrical Engineering and Informatics, Vol. 9, No. 5, pp. 1873~1881