A Systematic Investigation Of Artificial Intelligence-Based Approaches On Indian Sign Language Recognition

Major Project report submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

in

Computer Science and Engineering

By

Akankshya Pany (Regd. No. -1801109022)
Amit Kumar Parhi (Regd. No. -1801109040)
Bipin Gouda (Regd. No. -1801109094)

8th semester, Department of CSE

Under the Esteemed Guidance of

Dr. Niranjan Panigrahi
Assistant Professor
Department of Computer Science and Engineering
Parala Maharaja Engineering College, Berhampur

PARALA MAHARAJA ENGINEERING COLLEGE, BERHAMPUR
BIJU PATNAIK UNIVERSITY OF TECHNOLOGY, ODISHA, INDIA

Declaration

We hereby declare that the minor project work entitled **A Systematic Investigation Of Artificial Intelligence-Based Approaches On Indian Sign Language Recognition”** in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering submitted to the Department of Computer Science & Engineering, Parala Maharaja Engineering College, Berhampur is an authentic record of our work carried out under the supervision of **Dr. Niranjan Panigrahi**, Assistant Professor, Department of Computer Science and Engineering, Parala Maharaja Engineering College, Berhampur. The matter embodied in this major project report has not been submitted to any University or Institution for any degree or diploma.

Date: 27th June 2022

Akankshya Pany (1801109022)
Amit Kumar Parhi (1801109040)
Bipin Gouda (1801109094)

PARALA MAHARAJA ENGINEERING COLLEGE, BERHAMPUR, INDIA-761003

Certificate

This is to certify that the minor project report entitled **"A Systematic Investigation Of Artificial Intelligence-Based Approaches On Indian Sign Language Recognition"** submitted by **Akankshya Pany (1801109022), Amit Kumar Parhi (1801109040) Bipin Gouda(1801109094),** in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering submitted to Department of Computer Science & Engineering, Parala Maharaja Engineering College, Berhampur is a bonafide record of their original work. The matter embodied in this major project report has not been submitted to any other University or Institution for any degree or diploma.

(Signature of Guide)

(Signature of HOD, CSE

Date: 27th June 2022

Acknowledgment

We would like to express our sincere thanks and gratitude to the Principal, PARALA MAHARAJA ENGINEERING COLLEGE, Berhampur for allowing us to present the minor project on the topic "**A Systematic Investigation Of Artificial Intelligence-Based Approaches On Indian Sign Language Recognition**". We sincerely appreciate the inspiration, support, and guidance of everyone who has been instrumental in making this seminar to be a successful one. We extend our heartfelt thanks to Dr. Niranjan Panigrahi (Project Guide, Assistant professor, CSE), Dr. Debasish Mohapatra (HOD, CSE) for providing us with guidance at every moment during our project and giving us valuable suggestions. Their inspiration and wholehearted cooperation in carrying out our project is much appreciated. They took the pain of guiding us at every stage of learning on the project, despite their busy schedule. We also extend our deep sense of gratitude to our classmates, friends, and seniors who helped us directly or indirectly in the completion of this major project work.

Akankshya Pany (1801109022),
Amit Kumar Parhi (1801109040),
Bipin Gouda (1801109094),

Computer Science & Engineering Department

ABSTRACT

Communicating with a person having a hearing disability is always a major challenge. The work presented in this project is an exertion(extension) towards examining the difficulties in the classification of characters in Indian Sign Language (ISL). Sign language is not enough for communication of people with the hearing ability or people with speech disabilities. The gestures made people with disability get mixed or disordered for someone who has never learned this language. Communication should be in both ways. In this project, we introduce a Sign Language recognition using Indian Sign Language. The user must be able to capture images of hand gestures using a web camera in this analysis, and the system must predict and show the name of the captured image. The captured image undergoes s series of processing steps which include various Computer vision techniques such as the conversion to gray-scale and threshold operation. Artificial Neural Network (ANN), Convolutional Neural Network (CNN) and Transfer Learning methods are used to train our model and identify the pictures to predict the signs (HPC was also used for analysis). This will help us to make communication better with people having speaking and hearing disability.

Contents

List of Figures

1. Introduction

Indian sign language is a predominant sign language Since the only disability D&M people have is communication-related problems and they cannot use spoken languages hence they use sign language to communicate.

Communication is the process of the exchange of thoughts and messages in various ways such as speech, signals, behavior and visuals. Deaf and dumb(D&M) people make use of their hands to express their emotions using different gestures to express their emotions with other people.

Gestures are nonverbally exchanged messages and these gestures are understood with vision. This nonverbal communication of deaf and dumb people is called sign language.

In our project, we focus on producing a model which can detect Alphabets based on the hand gestures. The gestures we aim to train are as given in the image below.

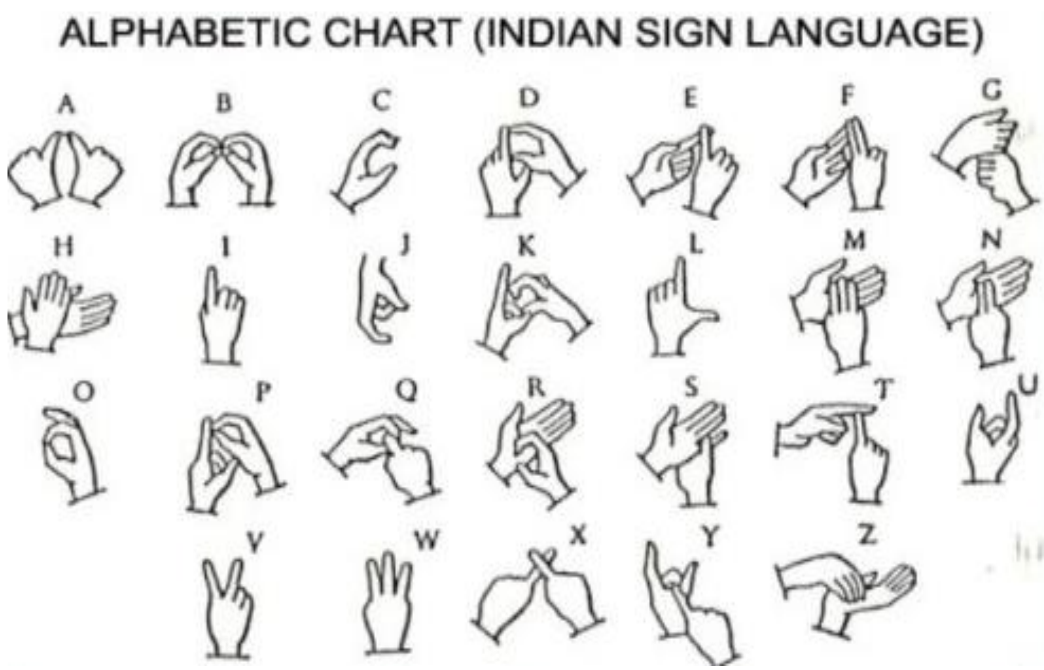


Figure 1. Alphabets of Indian Sign Language

2. Literature Survey

Sl. No	Author-Year	Methodology	Dataset	Country Sign Language
1	Gupta & kumar-2021	LP-based SLR	20,000 samples collected with multiple sensors	Indian sign Language
2	Prathum Arikeri- 2021	Deep Learning Neural Network	Manually prepared a dataset containing 42,745 RGB images belonging to 35 sign classes. There are 9 numeral signs (1-9) and 26 alphabet signs (A-Z).	Indian sign Language
3	Sharma et al.,2020	Hierarchical Network	Manually prepared dataset (150,000 images of all 26 categories)	Indian sign Language
4	Aly & Aly, 2020	Multiple Deep Learning Architectures (deep Bidirectional Long Short-Term Memory (Bi LSTM), recurrent neural network)	Arabic Sign Language database for 23 words	Arabic Sign Language
5	Badhe & Kulkarni-2020	Artificial Neural Network	Created own dataset	Indian sign Language
6	Bhavsar et al., 2020	Neuro-Fuzzy classifier	Self-Generated datasheet of 100 words	American sign Language
7	Yuan et al., 2019	Deep Learning Neural Network	Created Chinese Sign Language Dataset (CSLD) by discussing with expert	Chinese Sign Language
8	Sahoo et al., 2019	Artificial Neural Network And kNN	Datasheet 5000 (digits), 2340 (alphabets), 1250 (words)	Indian sign Language
9	Mariappan and Gomathi-2019	Fuzzy c-means clustering	Self-Generated data sheet of 130 words	Indian sign Language
10	Rao & Kishore-2018	MDC & Artificial Neural Network (ANN)	Dataset of 1313 frames.	Indian sign Language
11	Oyedotun & Khashman, 2017	Convolutional Neural Network (CNN)	Created own dataset of 320 images for 32 signs, 10 images for each	German Sign Language
12	Kishore and Anil -2016	Contour based model Using ANN	Self-Generated datasheet of 60 words	Indian sign Language
13	Wazalwar and Shrawankar-2017	Pseudo 2-dimensional Hidden Markov Model	Self-Generated data sheet of 60 words	Indian sign Language
14	Dixit & Jalal-2013	Multi-class Support Vector Machine (MSVM)	720 images, data set created	American sign Language
15	Jing-hao Sun-2013	Convolutional Neural Network (CNN)	The proposed system has a dataset of total 1600 pictures	Japanese Sign Language

The author in [1] collected 20,000 samples collected with multiple sensors. They have shown a proposed approach with LP-based SLR as classifier. The Sign language recognition (SLR) system measures and predicts human actions. The overall accuracy achieved using LP-based SLR is 92.27%.

The author in [2] used to train the deep learning model for ISL recognition. This dataset is available openly under creative commons licensing and can be accessed easily. The dataset contains 42,745 RGB images belonging to 35 sign classes. There are 9 numeral signs (1-9) and 26 alphabet signs (A-Z). The number of samples in the *'C'*, *'O'*, and *'I'* classes is 1,447,1,429, and 1,379, respectively the overall accuracy achieved using Deep Learning Neural Network is 75%.

The author in [3] Manually developed the dataset manually and used pre-trained VGG16 for training. More than 150,000 images of all 26 alphabets were included in the dataset. To keep the data consistent, the same background was used for each image.94.52% accuracy was obtained using the Deep Convolutional Neural Network (DCNN) for ISL recognition.

The author in [4] manually prepared a dataset of 23 arabic signs of 23 words. They have shown a proposed approach of Multiple Deep Learning Architectures (deep Bidirectional Long Short-Term Memory (Bi LSTM), recurrent neural network. The overall accuracy achieved using Bi-LSTM is 96%.

The author in [5] manually prepared a dataset of Indian sign and symbols. They have shown an proposed approach of Artificial Neural Network predicts the sign languages on basis of hand gestures. The overall accuracy achieved using of Artificial Neural Network is 85%.

The author in [6] proposed an approach to classify word signs using Neuro-Fuzzy approach. Authors displayed the final word using Natural Language Processing (NLP) technique and achieved 95% accuracy.

The author in [7] manually prepared a dataset of chinese symbols. They have shown an proposed approach of Deep Learning Neural Network which predicts the sign languages. The overall accuracy achieved using of Deep Learning Neural Network is 89%.

The author in [8] created 1250 video for word signs, where each category has 125 videos. They have shown comparison of proposed approach with kNN and Artificial Neural Network (ANN) as classifier. For each video 2-5 frames are considered in sequence to identify words. The overall accuracy achieved using kNN technique is 96.70% and with Artificial Neural Network is 93.70%.

The author in [9] discussed approach based on Fuzzy c-means clustering for recognizing sign words using 800 samples from different 10 signers. The signs captured using front camera of a mobile phone. 10 signers have performed the sign of words and these words are arranged in specific order to create sentences The overall accuracy achieved using Fuzzy c-means clustering is 96.70%

The author in [10] gave an average accuracy 85.58% performance for Word Matching Score (WMS) using a minimum distance classifier (MDC) and performance increased up to 90% for Artificial Neural Network (ANN) after making some little variations. It is possible to obtain more accurate classifiers using neural networks

The author in [11] gave an average accuracy achieved 98.125% in their proposed Deep learning model of CNN which is widely used for image extraction and preprocessing to predict the hand gestures and gives the output.

The author in [13] used Backpropagation algorithm on continuous sign language sentences using active contour hand shape features. Though the authors achieved 90.172% by capturing videos from four different signers, linguistic rules are not considered while performing sign language gestures.

The author in [12] proposed an algorithm for converting ISL sentences to English text. He used Pseudo 2-dimensional Hidden Markov Model (P2DHMM) for feature extraction, which is proven better than simple Hidden Markov Model. For converting recognized signs in English text, LALR parser was used and achieved 92% accuracy on their dataset.

The author in [14] worked on American Sign Language (ASL) to recognize the finger-spelling alphabets automatically. They also used Multi-class Support Vector Machines (MSVM) to classify the gestures. The overall accuracy achieved using Multi-class Support Vector Machines (MSVM) is 96%.

In Author in [15] The human hand was separated from the complex context, and the Cam Shift algorithm of CNN was used to detect real-time hand gestures. Then, using a convolutional neural network, the region of hand movements that was observed in real time is recognized, resulting in the identification of 10 common digits. The proposed system has dataset of total 1600 pictures for training dataset, 4000 hand gesture, 400 images for each type. This experiment shows accuracy about 98.3 percent.

3. Problem Statement & Objectives

3.1 PROBLEM STATEMENT

To do a systematic investigation of AI based approaches to recognize hand gesture-based ISL. (Indian sign language)

3.2 OBJECTIVES

The major objectives of the project are as follows:

- To collect the hand gestures image dataset for ISL.
- To implement a method to extract useful features.
- To train ML and DL models for classification of hand gesture.
- To study the model performance using evaluation metrics.

4. Challenges

There were many challenges to develop project.

- Handling environmental disturbance (e.g.: lighting sensitivity, background, and camera position)
- Selection of appropriate filter which could apply on ISL images for noise removal and reducing the sharpness of the image in the ROI region.
- Availability of sufficient image dataset for ISL to train the ML and DL model to improve the accuracy of the models.

5. Methodology

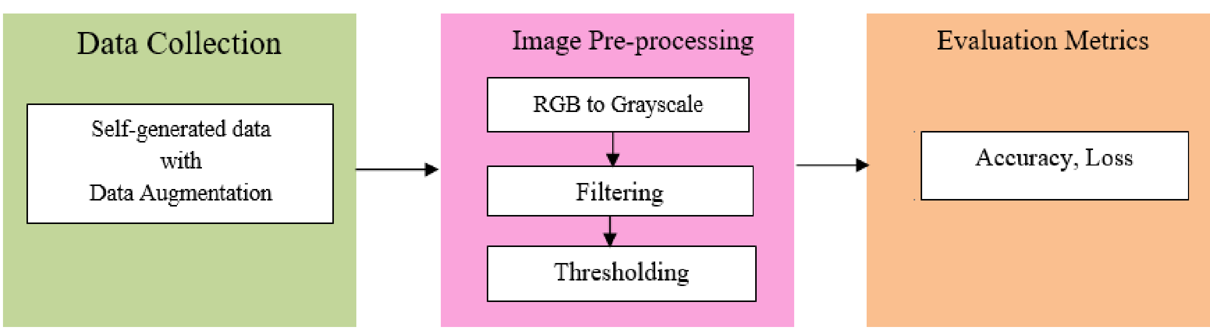


Figure 2. The framework of the proposed methodology

5.1. Data Collection & Augmentation

- The dataset used in our project was self-generated, the images were collected using the webcam and then data augmentation techniques were used to successfully generate the dataset

5.2. Image Pre-Processing

Firstly, we have to create some ROI around our hand or any object before Recognition.

We are doing image processing in three vital steps

1) Gray Scale conversion
2) Filtering
3) Thresholding

5.2.1 RGB To Gray conversion

Grayscale is the simplest model since it defines colors using only one component that is lightness. The amount of lightness is described using a value ranging from 0 (black) to 255 (white).
Reason for converting image from RGB to Grayscale.

1)Colour complexity
2)Learning image processing becomes easier
3)Easier visualization

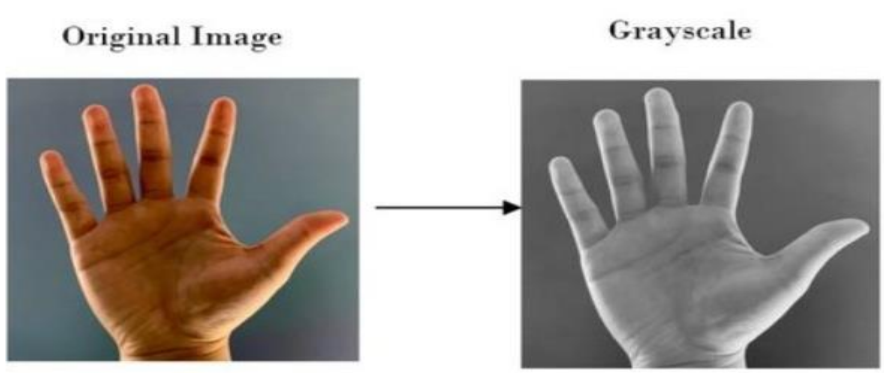


Figure 3. RGB to Grayscale Image

5.2.2 Filtering The Image:

- A Gaussian filter is a linear filter. It's usually used to blur the image or to reduce noise, contrast.

$$G(x,y) = \frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}}$$

- The Median filter is a non-linear filter that is most commonly used as a simple way to reduce noise in an image.
- The advantage a Gaussian filter has over a median filter is that it's faster because multiplying and adding is probably faster than sorting. It will be effective to find a counter after using gaussian blur.
- Median filter further excludes noise pixels but it loses a lot of image-structure information and image details.

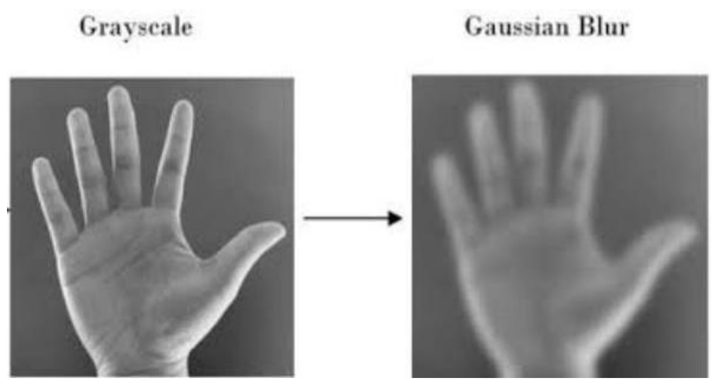


Figure 4. *Image after applying filter*

5.2.3 Thresholding Using OTSU Algorithm

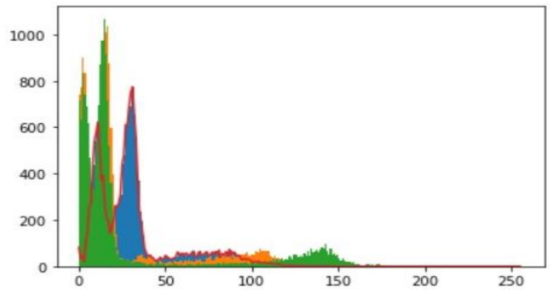
It is a process using which we separate an image into background and foreground, where in a normal RGB image background is converted to black and foreground to white, in our case the image was already converted to black and white in the thresholding stage so only the separation of background and foreground takes place.

Histograms

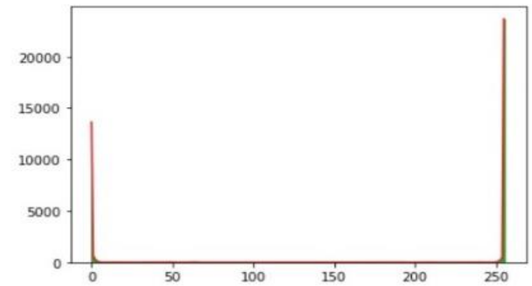
RGB Histogram

Black and White (binary image)

A SIGN-

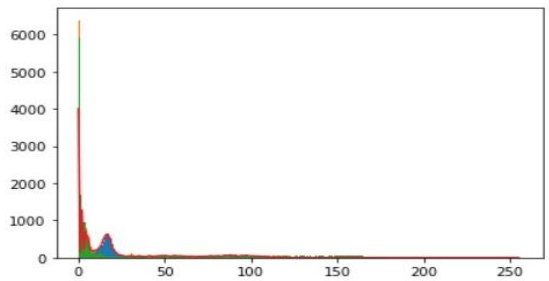


RGB histogram for A

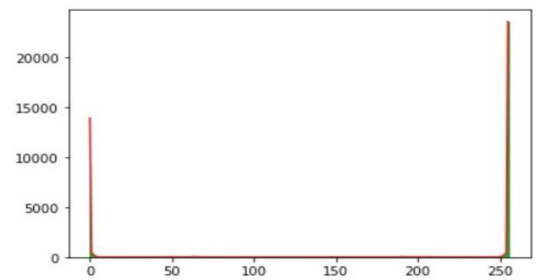


Binary image histogram for A

B SIGN-

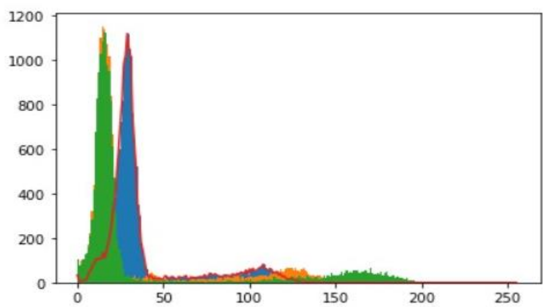


RGB histogram for B

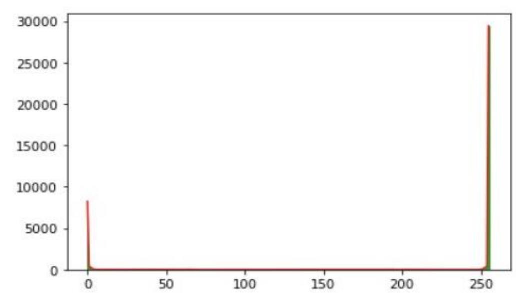


Binary image histogram for B

C SIGN



RGB histogram for C



Binary image histogram for C

OTSU ALGORITHM

- The algorithm iteratively searches for the threshold that minimizes the within-class variance, defined as a weighted sum of variances of the two classes (background and foreground) based on a threshold.

- So, If we choose a threshold of 100, then all the pixels with values less than 100 becomes the background and all pixels with values greater than or equal to 100 becomes the foreground of the image.

The formula for finding the within-class variance at any threshold *t* is given by:

$$\boxed{\sigma2(t)=\omega bg(t)\sigma2bg(t)+\omega fg(t)\sigma2fg(t)}$$

- $\omega bg(t)$ and $\omega fg(t)$ represents the probability of number of pixels for each class at threshold t
- $\sigma2$ represents the variance of colour values.

To understand what this probability means, Let,

- Pall be the total count of pixels in an image,
- PBG(t) be the count of background pixels at threshold *t*,
- PFG(t) be the count of foreground pixels at threshold *t*

So the weights are given by,

$\omega bg(t)=PBG(t)Pall$
$\omega fg(t)=PFG(t)Pall$

The variance can be calculated using the below formula:

$$\boxed{\sigma2(t)=\sum(xi-x)2N-1}$$

xi is the value of pixel at i in the group (Bg or Fg)
x is the means of pixel values in the group (Bg or Fg)
N is the number of pixels.

Bg-background
Fg- foreground

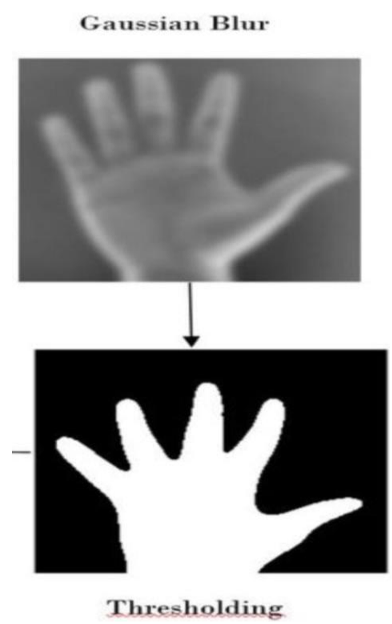


Figure 5. Filtered Image after Thresholding

5.3. Models Implemented

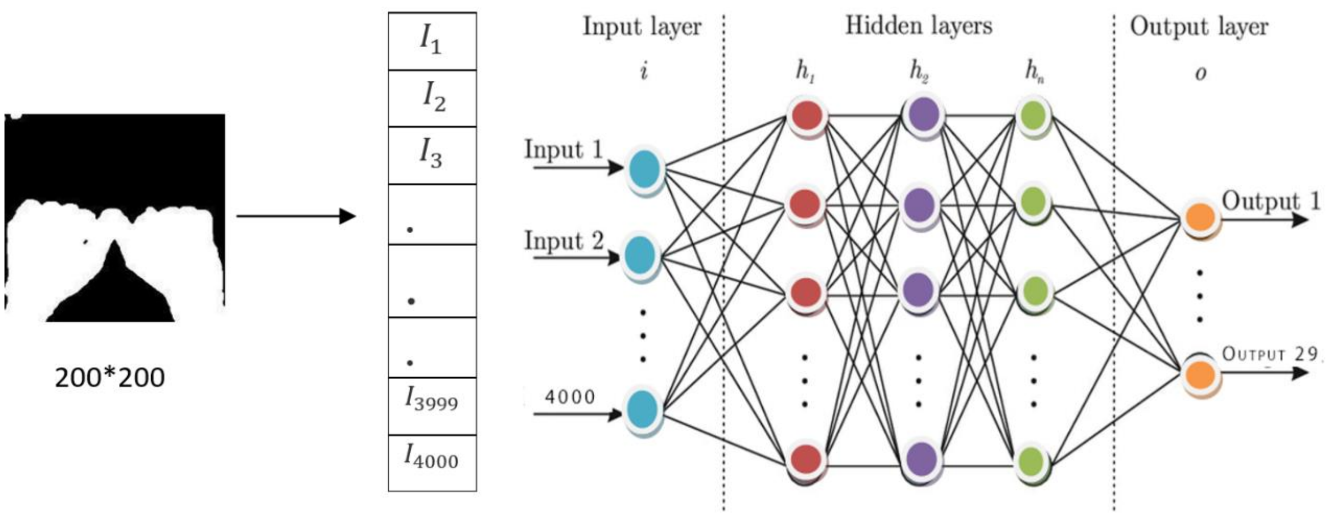
5.3.1. ANN-based model

Working: At First, information is passed to the input layer which then transfers it to the hidden layers, and interconnection between these two layers assigns weights to each input randomly at the starting point then bias is added to each input neuron, after this, the weighted sum which is a combination of weights and bias is passed through the RELU function. RELU Function has the responsibility of which node to fire for feature extraction and finally output is calculated. This whole process is known as Forward Propagation.

$$\sum_{i=1}^{n} W_i * X_i + b$$

Weight * Input +bias

Fig 6: Mathematical representation of an ANN



Input Layer:

A **Flatten Layer** is the input layer of our ANN model which takes the dimensions of the image and the image channel as input and converts it into a 1D array which is fed to the hidden layers, in our case the input is (200,200,1)

Hidden Layers:

The hidden layers consist of **Dense layers** and **Dropout layers** in our model, Dense layers are fully connected layers i.e., each neuron is connected to every other neuron of the next layer and Dropout layers are used to remove the unwanted neurons (the neurons whose removal doesn't affect the accuracy of the model). Activation function used in the Dense layers is **ReLU. [1]**

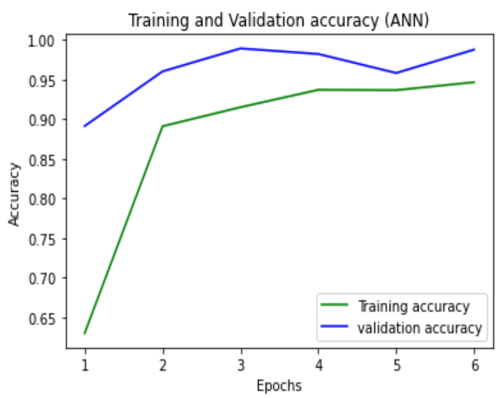
ReLU: It is an activation function which is defined by the function '*max(0, x)'* i.e. if the input is less than 0 the output is 0 else the output is same as input.

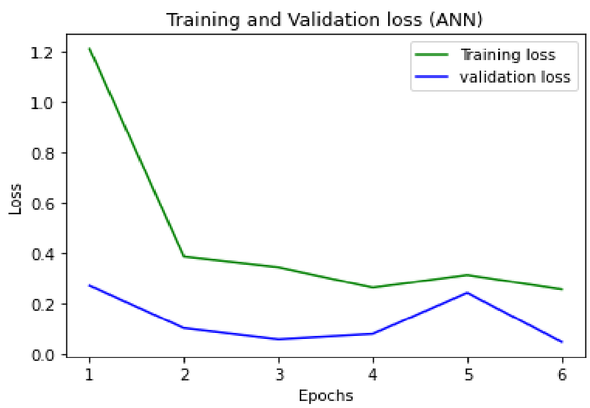
Output Layer:

The output layer is always a Dense layer in our case it has 29 neurons as we have a classification type problem with 29 classes, the activation function used is **SoftMax.**

SoftMax: The SoftMax function is an activation function that turns real values into probabilities.

Results





Disadvantages of ANN:

1) Treats local pixels same as pixels far apart.

2) Sensitive to location of an object in an image.

5.3.2. CNN-based model

A convolutional neural network or CNN, is a deep learning neural network planned for processing structured arrays of data such as images.

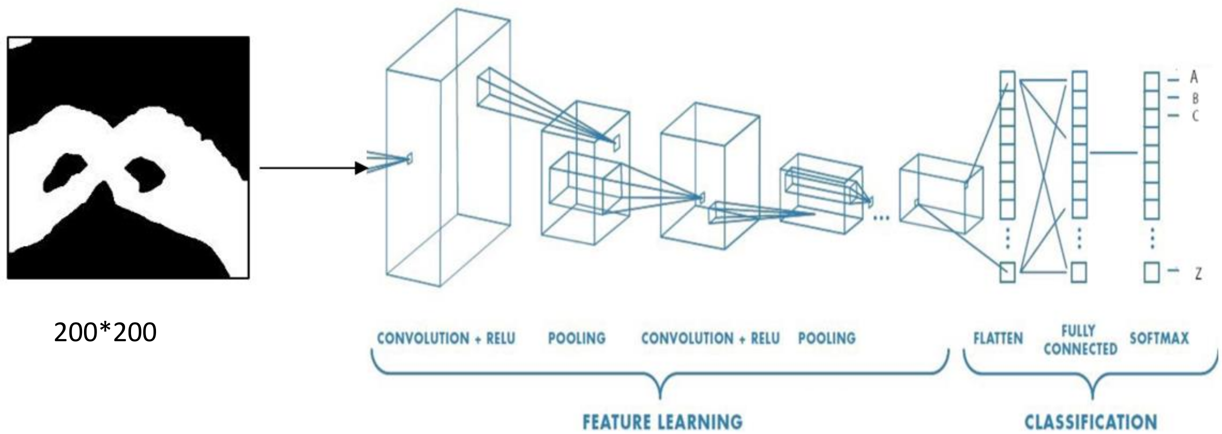


Fig 8: Architecture of our proposed CNN model

A CNN typically has three layers: a convolutional layer, pooling layer, and fully connected layer.

1) Input Layer:

The convolutional layer provides the input to the base CNN model which is an array of (200,200,1) dimension array of pixel values

2) Hidden Layers:

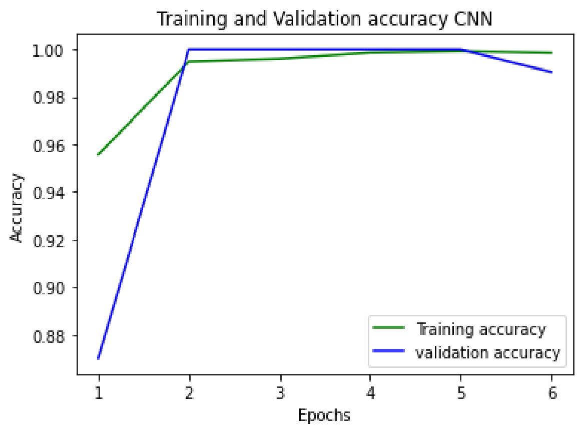
The hidden layer presents in-between input and output layers. It performs all the calculations to find hidden features and patterns.

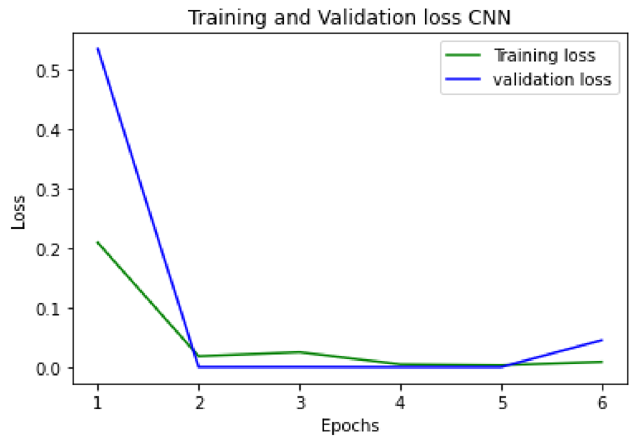
- 2.1) Convolutional layer- This layer generates feature maps
- 2.2) Maxpooling layer- This layer selects the maximum element from the region of the feature map covered by the filter; thus, the output is a feature map containing the most prominent features of the previous feature map
- 2.3) Dense Layer [1]
- 2.4) Dropout Layer [1]

3) Output Layer:

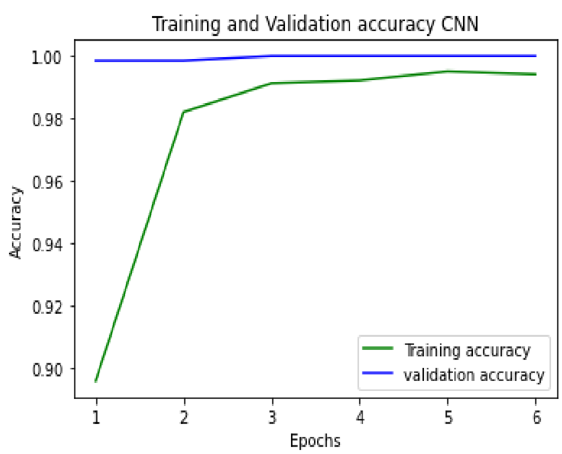
The output layer is a Dense layer with 29 neurons representing the 29 classes.

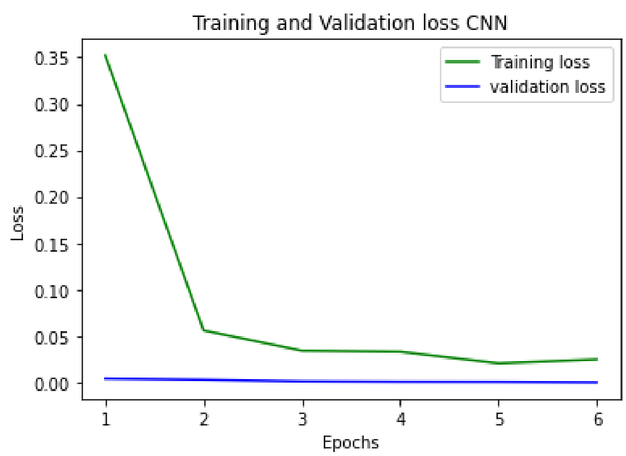
Results of one layered CNN





Results of two layered CNN





5.4. Implementation of Transfer Learning

Transfer learning is the reuse of a pre-trained model for solving on a new problem. In transfer learning, the knowledge of an already trained machine learning model is applied to a different but related problem.
Transfer learning has several benefits, saving training time, better performance of neural networks (in most cases), and there is no need of huge data.

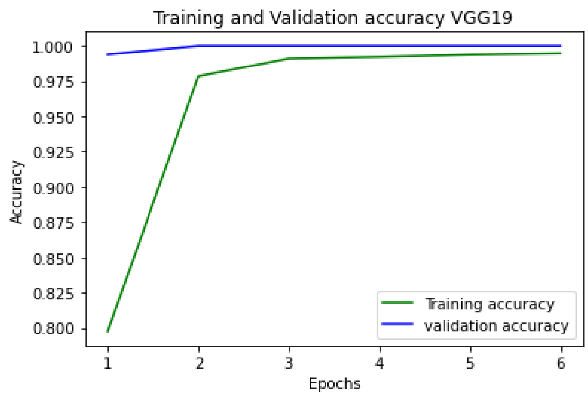
Transfer Learning using pre-trained CNN models:

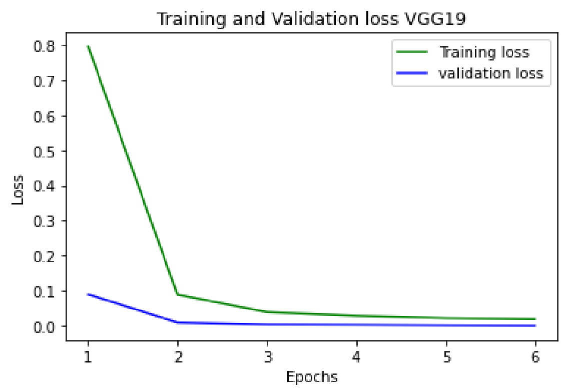
A) VGG 19

- VGG-19 is a convolutional neural network that is 19 layers deep.
- The network has a dense layer with 29 neurons as it's last layer.

The 18 layers were pre-trained CNN layers and in the last 19th layer of VGG19 transfer learning was applied

Results

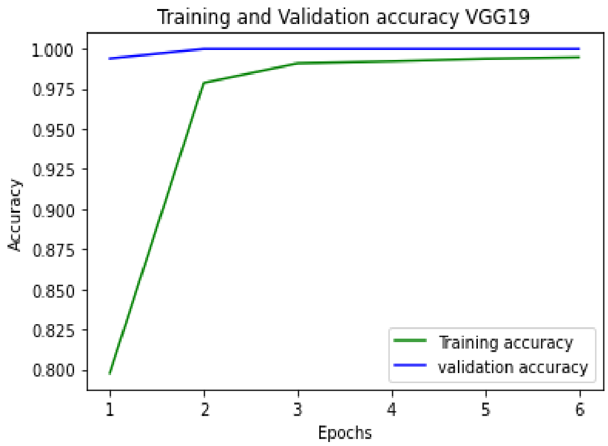


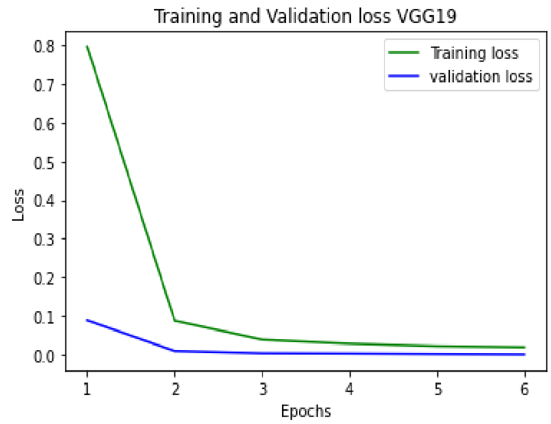


B) Inception V3

- Inception-v3 is a convolutional neural network that is 48 layers deep.
- The network has a dense layer with 29 neurons as it's last layer.
- The 47 layers were pre-trained CNN layers and in the last 48th layer of inception v3 transfer learning was applied.

Results





6. RESULTS

6.1 Accuracy and Loss

S. No	Models	Accuracy	Loss	Val-Accuracy	Val-Loss
1	ANN	94.6	0.28	97	0.06
2	CNN-1	99.4	0.01	99	0.05
3	CNN-2	99.2	0.025	100	0
5	VGG19	99.8	0.03	100	0
6	InceptionV3	99.7	0.024	99.7	0.022

6.2 Number of Layers and Time Elapsed in Laptop

Laptop/PC Configuration: -

OS Windows 11, 21H2
CPU intel Core I i5-8250U CPU 1.60GHz, 1.80 GHz
RAM 8.00 GB
HDD 2TB

S. No	Models	LAYERS	Training time in (Hr)
1	ANN	Flatten -1, Dropout-1, Dense-3	0.47
2	CNN-1	Convolution-1, Maxpool-1, Flatten-1, Dense-2, Dropout-2	2.54
3	CNN-2	Convolution-2, Maxpool-2, Flatten-1, Dense-3, Dropout-3	2.40
4	VGG19	Hidden layers total- 19, Dense Layer (Final O/p layer)-1	1.34
5	InceptionV3	Hidden layers total- 47, Dense Layer (Final O/p layer)-1	1.57

6.3 Performance Evaluation of Param Shavak System

Model	Epochs	Steps/epoch	Batch size	Laptop training time in (Hr, min)	HPC training time in (Hr,min)	Improvement
ANN	6	4000	3	0.47(Hr), 28 min	0.16(Hr), 10 mins	64.28%
CNN	6	4000	3	2.54(Hr), 152.4 min	0.7(Hr), 42 mins	72.44%

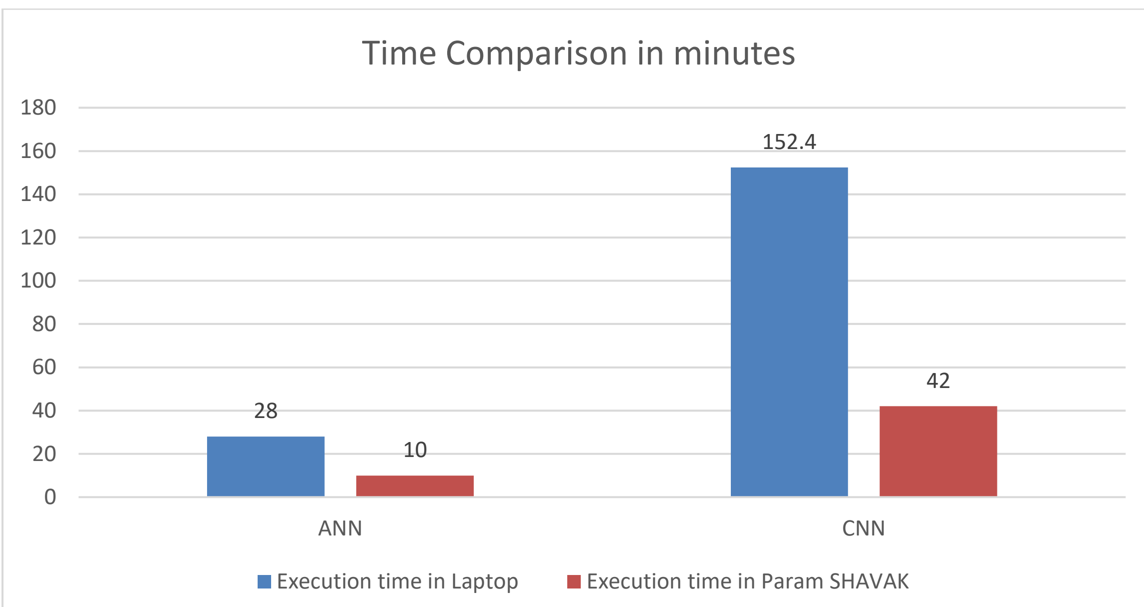


Fig 9: Time comparison Graph

7. Conclusion:

1. A systematic investigation of AI-based approaches is presented to recognize ISL.

2. The ISL dataset is pre-processed through image blurring and thresholding using OTSU's algorithm.

3. The filtered binary images are used as inputs for ANN, CNN, and pre-trained models (VGG19 and InceptionV3) for training and validation.

4. The performance time of the models is further improved using PARAM SHAVAK HPC system.

8. Future Scope

Currently, only ISL hand gestures were trained and detected, with the use of Natural Language Processing algorithms, this system can be extended to recognize sentences in ISL, by recognizing multiple gestures in the same video capture.

This approach could be further extended using Object Detection techniques to extract the hand region from the image. The only limitation in the implementation of Object Detection techniques is a requirement of a very wide variety of annotated hand samples so that it could detect hands in almost any position, orientation, and background.

The current approach also requires that the lighting conditions should be optimal – neither too dark nor too bright. The use of even better skin color segmentation techniques that can perform well under a wider variety of lighting conditions can lead to better segmentation results and in turn aid in feature extraction.

9. References

[1] Jing-Hao Sun, Ting-Ting Ji, Shu-Bin Zhang, Jia-Kui Yang, Guang-Rong Ji "Research on the Hand Gesture Recognition Based on Deep Learning",07 February 2019.

[2] Mokhtar M. Hasan, Pramoud K. Misra, (2011).“Brightness Factor Matching For Gesture Recognition System Using Scaled Normalization”, InternationalJournal of Computer Science Information Technology
(IJCSIT), Vol. 3(2).

[3] Simei G. Wysoski, Marcus V. Lamar, Susumu Kuroyanagi, Akira Iwata, (2002). “A Rotation Invariant Approach On Static-Gesture Recognition Using BoundaryHistograms And Neural International Journal of Artificial Intelligence Applications (IJAIA), Vol.3, No.4,July 2012.

[4] Stergiopoulou, N. Papamarkos. (2009). “Hand gesture recognition using a neural network shape fitting technique,” Elsevier Engineering Applications of Artificial Intelligence, vol. 22(8), pp. 1141– 1158, DOI: 10.1016/j.engappai.2009.03.008

[5] V. S. Kulkarni, S.D.Lokhande, (2010) “Appearance Based Recognition of American Sign Language UsingGesture Segmentation”, International Journal on Computer Science and Engineering (IJCSE), Vol. 2(3), pp. 560-565.

[6] Geethu G Nath and Arun C S, "Real Time Sign Language Interpreter," 2017 International Conference on Electrical, Instrumentation, and Communication Engineering (ICEICE2017)

[7] Kumud Tripathi, Neha Baranwal and G. C. Nandi,"Continuous Indian Sign Language Gesture Recognition and Sentence Formation", Eleventh International Multi-Conference on Information Processing-2015 (IMCIP-2015), Procedia Computer Science 54(2015) 523 – 531

[8] Noor Tubaiz, Tamer Shanableh, and Khaled Assaleh, "Glove-Based Continuous Arabic Sign Language Recognition in User-Dependent Mode," IEEE Transactions on Human-Machine Systems, Vol. 45, NO. 4, August 2015.

[9] B. Bauer,H. Hienz "Relevant features for video-based continuous sign language recognition", IEEE International Conference on Automatic Face and Gesture Recognition, 2002.

[10] Pigou, L., Dieleman, S., Kindermans, P.-J., Schrauwen, B. (2014). Sign Language Recognition UsingConvolutional Neural Networks. ITM Web of Conferences 40, 03004 (2021)
ICACC-2021