International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Real-Time Hand Gesture Recognition for Improved Communication with Deaf and Hard of Hearing Individuals

[1]K. Lakshmi, [2]Samiya, [3]M. Sri Lakshmi, 4M. Rudra Kumar, [5]Dr Prem Kumar Singuluri

Submitted: 25/02/2023 **Revised:** 16/04/2023 **Accepted:** 10/05/2023

Abstract: People who lack knowledge in sign language often face difficulty communicating effectively with those who are deaf or hard of hearing, but in such cases hand gesture recognition technology can provide an easy-to-use alternative for computer communication. This research concentrates on developing a hand gesture recognition system that works in real-time and utilizes universal physical traits present in all human hands as its basis for identifying these movements by automating the identification of sign gestures obtained through webcam footage using Convolutional Neural Networks (CNN) alongside additional algorithms integrated into this framework. Natural hand gestures are used for communication while the system prioritizes segmentation of these movements, and the automated recognition feature of the system is highly beneficial for people with hearing disabilities as it can help eliminate long-lasting communication barriers. The system also has potential applications in areas like human-machine interfaces and immersive gaming technology, so all parties involved can benefit from the ease that real-time hand gesture recognition brings through its potential as a tool for improving communication and reducing barriers faced by those who are deaf or hard of hearing.

Keywords: *Convolutional neural network, Deep learning, Gesture recognition, Sign language recognition, Hearing disability.*

1. Introduction

Effective communication is crucial for human interaction, and those who are deaf or hard of hearing may encounter difficulties communicating with those who can hear. Learning sign language, the main form of communication between the deaf communities and hearing individuals [1], can be challenging and requires specialized training. More accessible and intuitive communication methods are required for individuals who are deaf or hard of hearing. Technology recognizes hand gestures that come naturally and intuitively to individuals, which are more familiar to humans. This technology is used in various applications such as human-machine interfaces, sign language translation, and immersive game technology, among others. Developing a system [2] that accurately recognizes hand gestures and classifies them in real-time is a complex task. One of the main challenges is to ensure that the system can handle variations in lighting, background, and hand orientation with robustness. To distinguish between slight variations in hand movements, advanced machine learning models and image processing algorithms are required. Another challenge is to make the system accessible and user-friendly for individuals with varying levels of experience and proficiency in sign language. Presenting feedback in a clear and comprehensible manner while designing the system to cater to the diverse needs of users. The goal of this research is to create a live hand gesture recognition system that can help improve communication between individuals who have hearing difficulties and the general population, so the ability to automate the recognition of sign gestures can lead to easier and more accessible communication for everyone especially individuals who suffer from hearing loss.

This research aims at making hand gesture recognition technology more widely available and user-friendly for people who have different levels of expertise or proficiency in sign language, so our main aim is to establish a user-friendly system that can be easily understood by even those who have no prior knowledge of sign language.

It is unsurprising that American Sign Language (ASL)[3] has taken over as the preferred way of communication among individuals having Down 's syndrome and D&M, as through the usage of different modes such as oral communication or visual representations one can effectively convey their message. When communicating with hearing or speaking individuals there are some people affected by D&M that use hand gestures in order

[1]*Asst. Professor, Dept. of CSE, G. Pullaiah College of Engineering and Technology, Kurnool, Andhra Pradesh, India.* ***Email:*** *lakshmicse@gpcet.ac,in*
[2]*M. Tech Student , Dept. of CSE, G. Pullaiah College of Engineering and Technology, Kurnool, Andhra Pradesh, India ,E- mail: sanashaik380@gmail.com*
[3]*Asst. Professor, HOD-CSE, Dept. of CSE, G. Pullaiah College of Engineering and Technology, Kurnool, Andhra Pradesh, India ,E-mail: srilakshmicse@gpcet.ac.in*
[4] *Professor, Dept. of CSE, G. Pullaiah College of Engineering and Technology, Kurnool, Andhra Pradesh, India ,Email: mrudrakumar@gmail.com*
[5]*Sr. Professor and Dean Innovations , Dept. of CSE, G. Pullaiah College of Engineering and Technology, Kurnool, Andhra Pradesh ,India*
Email: dean@gpcet.ac.in

to express what they're thinking, and one must rely on visual acuity to decipher the meaning behind nonverbal cues expressed through gestures. Sign language [4] is a prevalent way of communicating without words that individuals with hearing or speech disabilities often use and utilizing neural languages and convolutional networks has made the creation of more efficient sign languages possible.

The primary objective of this research is to investigate how convolutional neural networks can be utilized for constructing a dual-camera first-person vision translation system specifically designed for sign language, comprising three main divisions; you can acquire an extensive knowledge regarding system design including dataset and deep learning model training from the subsequent segments. Table 1 offers a summary of the essential elements that define sign language as a visual mode of expression Improved accessibility for those who have D&M is the long-term goal.

This research aims to create a model to recognize and combine each gesture into fingerspelled-based hand gestures. The gestures that this project wishes to teach are depicted in image 1.

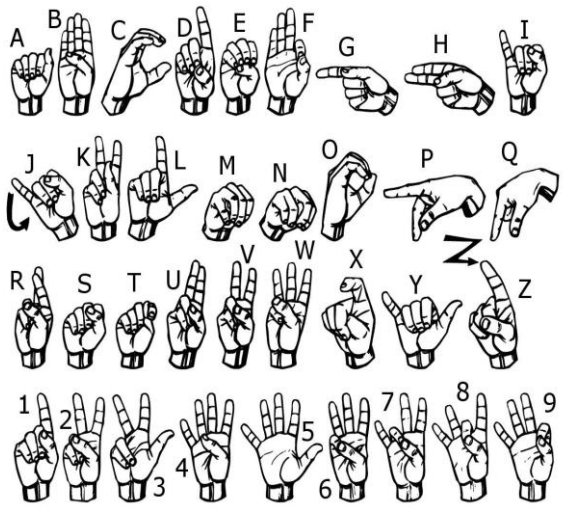

Fig 1: Datasets of ASL sign language

This research seeks to design a real-time system for recognizing hand gestures using relevant form-based traits common to all human hands, and Convolutional Neural Networks (CNN) and various other algorithms will be employed in order to automate the recognition of sign language gestures that are recorded via a webcam. The focus of the system is primarily on segmenting hand motions to recognize sign gestures by utilizing similarities in human hand shape, so this research intends to develop a better mode of communication which would enable people with impaired hearing abilities to communicate easily with others in society. The purpose of this study is also to exhibit how real-time hand gesture recognition can lead to improved communication while investigating its prospective applications in various domains including human-machine interfaces, sign language translation and immersive game technology.

Organization of the paper is organized as follows, Section I contains the introduction of the system in detail,Section II contain the literature survey, Section III contain proposed system, Section IV contain the results and discussion, section V contains the conclusion of the system, Section VI describes the future scope of the system, Section VII contain the references.

2. Related Work

Numerous studies have used different machine learning methods to develop automated sign language recognition systems recently and in 2020 a sign language recognition system that employed convolutional neural networks and computer vision was suggested by the author[5]. While facing some hardware limitations, the system ultimately ended up being quite costly In spite of this approach there were discrepancies due to its incompatibility with various sign language systems globally, but a new method for interpreting sign language automatically via a deep learning-based translation system was introduced by Timing with its primary limitations being high costs and suboptimal efficiency[6].

Their research work [7] put forward a fundamental model for system recognition and design Unfortunately, this methodology proved to be quite time-consuming[8] .The use of convolutional neural networks to recognize sign language was introduced in [9]The effectiveness of various machine learning techniques in recognizing sign language is shown by these studies, nevertheless they point out the challenges and shortcomings of existing systems such as high cost, inaccuracies, time-consuming model training alongside hardware limits. The available

evidence proposes the necessity of carrying out additional research aimed at creating more accurate and efficient sign language recognition tools that can interpret different types of global signed languages with ease, and the reviewed studies indicate a growing interest in building such tools relying on methods of machine learning such as convolutional neural networks. Some of these systems have excellent accuracy rates, but this is often counteracted by issues such as being costly or having limitations in terms of the hardware that they are compatible with. Furthermore, low levels of precision are experienced when using them on other types of sign language. This research paper aims to eliminate these constraints through its proposed system, and it focuses on segmenting hand movements and utilizing form-based qualities common to all human hand[10]s. The utilization of a sign gesture recognition automated system could lead to increased accessibility in communicating with those who are deaf or hard of hearing ultimately helping bridge longstanding gaps that exist.

The implementation of a real-time system for recognizing hand gestures makes this current study more sophisticated than previous ones with potential benefits such as sign language translation or human-machine interface improvement, and the recognition of sign gestures captured through a webcam is automated using convolutional neural networks combined with other algorithms in the system making it possible for individuals to communicate their ideas effectively using natural hand gestures[11]. The literature supports our proposal by demonstrating clear benefits as well as potential difficulties in utilizing deep learning methods to recognize sign language, and the ongoing efforts to create better communication tools for those who are deaf or hard-of-hearing receive a boost through this study's use of cutting-edge technologies like deep learning and computer vision.

Using deep learning techniques in a real-time system, [12] and[13] proposed the recognition of Arabic sign language in their related work5% was obtained, but the system's restricted range of words and necessity for an explicit camera configuration are among its negative aspects. Previous researches conducted by them have shown that a reliable way to identify hand gestures is by using depth images and a deep learning-based system as suggested in the studies 14 to 16, so achieving an accuracy of 96.7%, authors classified hand gestures using pre-trained CNN model but the drawback that stands out for this system is how much it relies on depth cameras.

Deep learning techniques like CNNs and LSTMs [13] are widely used in developing sign language recognition systems—this is reflected in various research works, however many of these systems have constraints including hardware dependency along with limited lexicon leading to reduced accuracy. Consequently, the development of a robust and efficient sign language recognition system becomes imperative in order to address these limitations, and enable easier communication for persons who are deaf or hard of hearing

A supplementary research analyzed in this investigation was Real-time Hand Gesture Detection combined with Recognition which used Convolutional Neural Networks as carried out by [14]The main purpose of the system was to recognize hand signs and convert them into particular actions that manage different applications, using advanced technology, the system is capable of detecting and recognizing six distinct hand movements in real time with a 96% accuracy rate.

In order to recognize gestures quickly using CNNs [15], the authors proposed a new technique involving three concurrent streams of convolutions and evaluated the effectiveness of the proposed architecture in accurately recognizing gestures using two datasets to achieve a high level of precision at 94.5%

Taken together these studies show that using convolutional neural networks has great potential for real-time hand gesture recognition, but there are still obstacles to overcome such as hardware constraints and maintaining accurate results. Building on prior research findings, this research project's aim is to develop an actual time recognition system for hand-gestures that will be used effectively in recognizing diverse types of signal-hand languages obtained from webcams, potentially improving effectiveness and accessibility while communicating with hearing-impaired individuals.

- Significant weaknesses in sign language recognition systems came to light from the reviewed studies:
- Some systems are not easily accessible by a larger audience because of their shortcomings such as high cost and limited compatibility with hardware.
- Existing recognition systems face challenges in accurately recognizing different sign languages.
- Less efficiency in certain systems can be attributed to the drawback of time-consuming model training.
- Applicability can be reduced due to limitations in supporting only specific sign language systems.
- Specific camera requirements and limited vocabulary are just a few of the downsides that come with certain existing systems.
- The majority of existing systems face issues with hardware dependence leading to limited vocabularies and low accuracies.

These limitations highlight the importance of continuing

research into developing accessible and effective sign language recognition systems which can work around them while making communication simpler for individuals who have difficulty hearing. Based on the restrictions identified by previous studies examined in the literature review, the suggested work appears to be a positive way forward for identifying sign languages and by prioritizing hand motion segmentation and employing relevant form-based attributes that are universally present across all human hands, the proposed system seeks to overcome the shortcomings observed in existing systems. Moreover, the proposed system utilizes convolutional neural networks for real-time detection of hand gestures which has previously shown promising results. Furthermore, sign language translation along with other uses such as human-machine interfaces or immersive game technology has the potential to make communication more accessible in the deaf/hard-of-hearing community using this system.

Overall it seems like the proposed work is progressing towards developing more accurate and efficient ways of recognizing signs which could potentially support several sign languages globally, however, a comprehensive evaluation of the system's effectiveness and usability must be carried out to ensure that it is useful for individuals who are deaf or hard of hearing.

3. Methodology

Collecting data serves as a starting point for the proposed system, and the method used by many researchers to capture hand movements is with the aid of cameras and sensors, but with this particular system we prefer using our built-in webcam. The gathered images are processed using various post-processing stages and the initial stage of our process includes extracting colors with a calorimeter. Later on we use HSV[16] color space to determine which part of image is background and remove it, using morphological methods consisting of sequential steps involving dilation and erosion using elliptical kernels after separating skin tone for defining regions on which we can apply masks. With the use of OpenCV we give equal treatment to all images regardless of their movement capture and the division of each image in our dataset containing 2000 American sign gestures into separate training and testing subsets [17] is what allows us to achieve our objective. In order to facilitate both learning & evaluation process properly—it's important that we segregate our data into two distinct sections i.e. Training and testing involve utilizing an initial set of 1600 photos strictly for learning purposes, while leaving aside 400 images exclusively meant for the evaluation process. The overall dataset follows an indicative training-testing ratio, where a maximum limit of 80% is assigned towards building a robust model. Apart from classifying the hand gestures in each frame during classification stage, we also use a convolutional neural network for training purposes, so testing follows once we have trained the model, and it helps us determine its capacity for predicting alphabet letters. If letter predictions are accurate then we will consider the system successful.

3.1 Proposed Work Flow model:

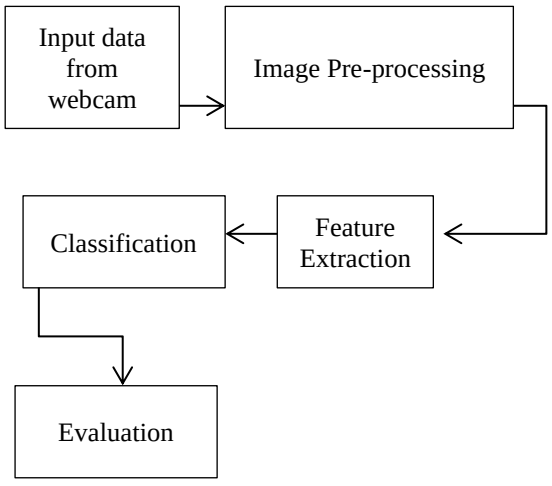

Fig. 2. Flow Model of the Proposed Work.

The first step in implementing the suggested system involves collecting data, researchers typically make use of sensors or cameras to capture images of hand movements across various studies. Using a built-in webcam the motions made by the hands are recorded in this case.

Capture and Process Hand Movement Images: After capturing images of hand movements, they go through a process where a calorimeter extracts their color information, and identification and elimination of backgrounds from images are made possible using a series of post-processing steps that utilize the HSV (Hue-Saturation_Value) color model.

After getting rid of an image's background you could isolate and separate skin tone thereby enabling to decide on an area for its application, then a mask is applied on images through morphological processing which is then refined by dilating and eroding using elliptical kernels.

Firstly after processing the images and separating skin tones from them for preparing our data set, we divide it into two subsets called training set or testing set to carry out effective training within this context. It has been deemed appropriate to take advantage of a relatively large portion consisting approximately eighty percent worth of data whereas only twenty percent would be made use of during assessment.

Each frame from the training data set undergoes analysis by a convolutional neural network (CNN), as part of its programming process for recognizing and classifying diverse hand motions, so the CNN forms associations between recognizable image patterns and corresponding alphabetic symbols[18].

By using the remaining 20% portion of data after completing training phase for a CNN one can assess its capability to identify patterns related to hand movements and make predictions accordingly, accuracy in predicting alphabet letters determines if this system is successful or not.

3.1 Input data: Captured through a webcam are images of hand movements made while performing American Sign Language (ASL) gestures which serves as input data for the proposed system and included in the dataset is a total of 2000 images for ASL gestures which have been separated into both training and test datasets[19].

3.2 Image Pre-processing:

3.2.1 HSV colourspace and background elimination

In the HSV color space, colors are defined using three parameters: Hue, Saturation and Value, while color temperature refers to how warm (yellow) or cool (blueish) a shade appears. You can tell how bright a color is by looking at its Value component and how intense or pure it is by looking at its Saturation, so one of the reasons why developers often use HSV color space in image processing and computer vision applications lies on its ability to separate brightness from chrominance data which simplifies isolating specific colors. The identification and removal of image background from hand gestures is achieved by incorporating HSV color space in this proposed system and a series of subsequent post-processing steps are carried out employing the following mathematical approach.

Convert the RGB image to the HSV color space.

HSV stands for hue, saturation, and value. It is a cylindrical color space that represents colors in terms of their hue angle, which ranges from 0 to 360 degrees, their saturation, which ranges from 0 to 1, and their value, which ranges from 0 to 1.

To convert an RGB image to the HSV color space, the following formulas are used:

$$V = \max(R, G, B)\ S = (V - \min(R, G, B)) / V$$

If V is 0, then S is also 0. If V is not 0, then S is defined as:

$$S = (V - \min(R, G, B)) / V$$

H is then calculated as follows:

$$\begin{aligned} \text{if } V == R{:}\ H &= 60 * (G - B) / (V \\ &\quad - \min(R, G, B)) \text{ elif } V == G{:}\ H \\ &= 120 + 60 * (B - R) / (V \\ &\quad - \min(R, G, B)) \text{ else: } H \\ &= 240 + 60 * (R - G) / (V \\ &\quad - \min(R, G, B)) \end{aligned}$$

If H is negative, it is added to 360 to obtain a positive value.

Once the image is converted to the HSV color space, the next step is to eliminate the background.

Define a range of values for the Hue, Saturation, and Value components to isolate the color of the skin.

- HSV (Hue, Saturation, Value) is a color space that represents colors based on three parameters: hue, saturation, and value. Hue is the actual color of the pixel, saturation is the intensity or purity of the color, and value represents the brightness or darkness of the color.
- To isolate the color of the skin, a range of values for the Hue, Saturation, and Value components is defined. This range is used to identify the pixels that represent skin color. The range is defined based on the average skin color of the dataset being used.

For example, if we want to isolate pixels that represent skin color, we might define the following ranges:

Hue range: 0 to 50

Saturation range: 0.23 to 0.68

Value range: 0.35 to 0.95

This means that any pixel with a hue value between 0 and 50, a saturation value between 0.23 and 0.68, and a value between 0.35 and 0.95 is likely to represent skin color.

Mathematically, this can be represented as follows:

Let I be the input image, and H, S, and V be the Hue, Saturation, and Value components of the HSV color space, respectively. We can define a range of values for each component as follows:

Hue range: Hmin <= H <= Hmax

Saturation range: Smin <= S <= Smax

Value range: Vmin <= V <= Vmax

Any pixel (i, j) in the input image I that satisfies these conditions can be classified as representing skin color:

$$Hmin \leq H(i,j) \leq Hmax\ Smin \leq S(i,j) \leq Smax\ Vmin \leq V(i,j) \leq Vmax$$

This process of defining a range of values for the Hue, Saturation, and Value components is called color segmentation or color thresholding. It is a common technique used in computer vision and image processing for object detection, tracking, and recognition.

Apply a mask to the image using the range of values defined in step 2 to remove all other colors except for the skin tone.

In this step, a binary mask is applied to the image in order to remove all colors except for the skin tone. The binary mask is generated by thresholding the image with a range of values defined in step 2 (i.e., the range of values for Hue, Saturation, and Value that correspond to the skin tone). Pixels within this range are set to 1 in the mask, while pixels outside the range are set to 0.

Mathematically, let I be the original RGB image and let I_{HSV} be the same image converted to the HSV color space. Then, let H, S, and V be the individual Hue, Saturation, and Value components of I_{HSV}, respectively. The range of values for skin tone is defined as a set of upper and lower bounds for H, S, and V, denoted as H_u, H_l, S_u, S_l, V_u, and V_l, respectively.

The binary mask, M, is generated by thresholding the individual components of I_HSV with the respective upper and lower bounds as follows:

$$M(H_u > H > H_l, S_u > S > S_l, V_u > V > V_l) = 1$$

M(otherwise) = 0

Here, the notation "A > B > C" means that B is between A and C, i.e., A is the upper bound and C is the lower bound.

Once the binary mask is generated, it is applied to the original image I to produce the final skin tone image, I_{skin}:

$$I_{skin} = I * M$$

Where $*$ denotes element-wise multiplication. This operation sets all pixels in I that correspond to non-skin-tone regions to 0, effectively removing all other colors except for the skin tone.

Apply morphological operations such as dilation and erosion using elliptical kernels to smooth and refine the edges of the skin tone.

The background elimination process is important because it helps to reduce the amount of noise in the image and isolate the hand gestures, making it easier for the classification model to identify and recognize them.

Fig. 3(a) Input image from webcam

3(b) background is set to black using HSV (first image)

3.3 Segmentation

After that, the first image is converted to grayscale. The colour of the skin gesture will be lost throughout this procedure, but our system's resistance to changes in brightness or illumination will be increased as a result. The pixels in the modified image that are not black are binarised, while the pixels in the original image remain intact, and are therefore black. The hand gesture is divided in two ways: first, by removing all of the connected components in the image, and second, by allowing only the part of the image that is extremely connected, which in this case is the hand gesture. The frame is scaled to a 64 X 64 pixel resolution. At the conclusion of the segmentation process, binary pictures of size 64 X 64 are created, in which the white area represents the hand gesture and the black coloured area represents the rest of the image (see figure 3(a) and (b)).

Fig. 4(a) Binary Image 4(b) Resigned Image

Let's denote the original image as I, and the grayscale version of I as I_gray.

To convert I to grayscale, we can use the following formula:

$$I_{gray(i,j)} = 0.2989 * I(i,j,1) + 0.5870 * I(i,j,2) + 0.1140 * I(i,j,3)$$

where i and j are the pixel coordinates, and I(i, j, k) is the k-th channel value of the pixel at (i, j) in the original image.

To increase the system's resistance to changes in brightness or illumination, we can use a technique called histogram equalization. Let's denote the histogram-equalized grayscale image as I_{eq}. The mathematical formula for histogram equalization is as follows:

$$I_{eq(i,j)} = \text{round}\left(\frac{(CDF(I_{gray(i,j)}) - CDF_{min})}{(M * N - CDF_{min}) * (L - 1)}\right)$$

where round is the rounding function, CDF is the cumulative distribution function of I_{gray}, CDF_min is the minimum value of CDF, M and N are the dimensions of I_{gray}, L is the number of possible intensity levels (usually 256), and $I\_eq(i,j)$ is the intensity value of the pixel at (i, j) in the histogram-equalized grayscale image.

To binarize the modified image, we can use a thresholding technique. Let's denote the binary image as B. The mathematical formula for thresholding is as follows:

$$B(i,j) = 1 \text{ if } I_{eq(i,j)} > T \text{ else } 0$$

where T is the threshold value.

To remove all the connected components in the image, we can use a morphological operation called opening. Let's denote the opened image as O. The mathematical formula for opening is as follows:

O = open(B, SE)

where SE is a structuring element that defines the shape and size of the operation.

To allow only the part of the image that is extremely connected, we can use a morphological operation called closing. Let's denote the closed image as C. The mathematical formula for closing is as follows:

C = close(B, SE)

where SE is the same structuring element used for opening.

To scale the frame to a 64 x 64 pixel resolution, we can use interpolation. Let's denote the final segmented image as S. The mathematical formula for interpolation is as follows:

$$S(i,j) = BilinearInterpolation(C, i / k, j / k)$$

where BilinearInterpolation is the bilinear interpolation function, k is the scaling factor (which is calculated as the ratio of the original image size to the desired size, in this case 64), and S(i, j) is the intensity value of the pixel at (i, j) in the final segmented image.

3.4 Feature Extraction

In image processing, selecting and extracting key features from an image is one of the most critical parts of the process. In most cases, the volume of data contained inside an image dataset demands a lot of storage capability. Data reduction through feature extraction can help us tackle this issue. On top of all that, it keeps the classifier's accuracy high while simplifying its complexity. We identified the binary pixels in the images to be the most important attributes in this scenario. Images of American Sign Language gestures have been classified effectively after scaling them to 64 pixels. When 64 pixels are multiplied by 64 pixels, there are 4096 features.

Let *I* be an image dataset with *N* images, each containing M pixels. Let *X* be an $N \; x \; 4096$ matrix, where each row corresponds to an image and each column corresponds to a binary pixel in the image (0 or 1).

To extract the key features from I, we identify the binary pixels in each image and create *X* accordingly. Thus,

each row in X represents a unique combination of binary pixel values for an image in I.

To reduce the volume of data in I, we perform feature extraction using X. This involves analyzing X to identify patterns and relationships between the binary pixels that are most useful for classification. Let Y be an $N\ x\ K$ matrix, where each row corresponds to an image and each column corresponds to a selected feature (e.g. a group of related binary pixels). We can choose K to be smaller than 4096 to reduce the storage requirement for Y.

To classify images of American Sign Language gestures effectively, we scale each image to 64 x 64 pixels. This means that M = 4096, and X has 4096 columns. We can then use machine learning algorithms (e.g. neural networks) to train a classifier on Y and use it to predict the sign language gesture in new images. By reducing the number of features in Y, we simplify the complexity of the classifier while maintaining its accuracy.

3.5 Classification

It is necessary to extract characteristics from the frames and anticipate hand gestures to use a CNN model. Image recognition is the primary use of this multi-layered feed forward neural network. Several convolution layers are included in the CNN architecture. Each convolution layer consists of a pooling layer, an activation function, and, optionally, batch normalization. A set of fully-connected layers is also included in the design of this structure. Each time one of the photographs travels through the network, the size of the image gets smaller and smaller. Because of max pooling, this occurs in the game. With the final layer, we can forecast the probability of the different classes.

3.6 Implementation

Firstly, the program examines each letter in the current string and verifies whether its count is greater than or equal to a specific threshold value (in this case, 50). If it is, the program outputs that letter and adds it to the current string. Otherwise, the count of images in the current dictionary is deleted. Next, the images from the first convolutional layer are passed to a densely linked layer comprising 128 neurons. The output of this layer is reshaped into an array of 30 by 30 by 32 = 28,800 values. This layer receives an array of 28,800 values as its input. The output of the first densely connected layer is used as an input to a fully connected layer with 96 neurons. The second layer is also densely connected. Finally, the number of neurons in the last layer is equal to the number of classes it identifies (alphabets and blank sign) and is an input to the second densely connected layer. To avoid overfitting, a dropout layer with a value of 0.5 is employed. The dropout layer randomly removes some activations from the previous layer and sets them to 0. This technique prevents the network from becoming too dependent on the training instances and improves its performance on new examples.

Every layer uses ReLu (Rectified Linear Unit) for activation, which computes a max (x, 0) for each input pixel. This helps in learning complex features, prevents the vanishing gradient problem, and speeds up the training process. Additionally, max pooling and ReLu activation are used for the input image to reduce overfitting by decreasing the number of parameters and overall computational cost. The system architecture for the sign language system is displayed in Figure 4. The testing procedure flowchart is shown in Figure 5, which outlines the process of testing each letter independently with 50 iterations given to each letter, recording the frequency of recognition for each letter. However, the recognition may be incorrect if the letter gesture is not recognized.

For each letter in the current string whose count is greater than or equal to a specific threshold value, the program outputs that letter and adds it to the current string. (In this case, the values are set at 50, and the difference threshold is 20.) Images from the first convolutional layer are fed to a densely linked layer of 128 neurons, and the output from this layer is reshaped into an array of 30 by 30 by 32 = 28,800 values. This layer receives an array of 28,800 values as its input. The output of these layers is received by the second densely connected layer. To avoid overfitting, a dropout layer with a value of 0.5 is utilized.

The output of the first densely connected layer is used as an input to a fully connected layer comprising 96 neurons. The second layer is also densely connected. Finally, the number of neurons in the last layer is equal to the number of classes it identifies (alphabets and blank sign) and is an input to the second densely connected layer.

ReLu (Rectified Linear Unit) is used for activation in each layer (convolutional and fully connected neurons). ReLu computes a max (x, 0) for each input pixel, which allows for more complex features to be learned. This approach eliminates the vanishing gradient problem and speeds up the training process.

A pooling layer may be generated by utilizing max pooling with a pool size of (2, 2) and ReLu activation on the input image. Overfitting is reduced since the number of parameters reduces the overall computational cost.

After training, the network weights become so fine-tuned to the training instances that the network does not perform well when given fresh examples, which is

known as the "dropout layer" problem. To solve this, the dropout layer removes a random subset of activations from the previous layer and sets them to 0. This ensures that the network is able to correctly classify or output for a single case, even if part of the activations is dropped.

In summary, the program applies a threshold to each letter in the current string and outputs the letters whose count is above the threshold. It then feeds the images from the first convolutional layer into densely connected layers, reshaping the output of the first layer to an array of 28,800 values. A fully connected layer with 96 neurons is then used, followed by a densely connected layer whose number of neurons is equal to the number of classes. To avoid overfitting, a dropout layer is employed.

ReLu is used for activation in each layer, with max pooling and ReLu activation utilized for the input image. The architecture of the sign language system is depicted in Figure 4, and the testing process flowchart is illustrated in Figure 5.

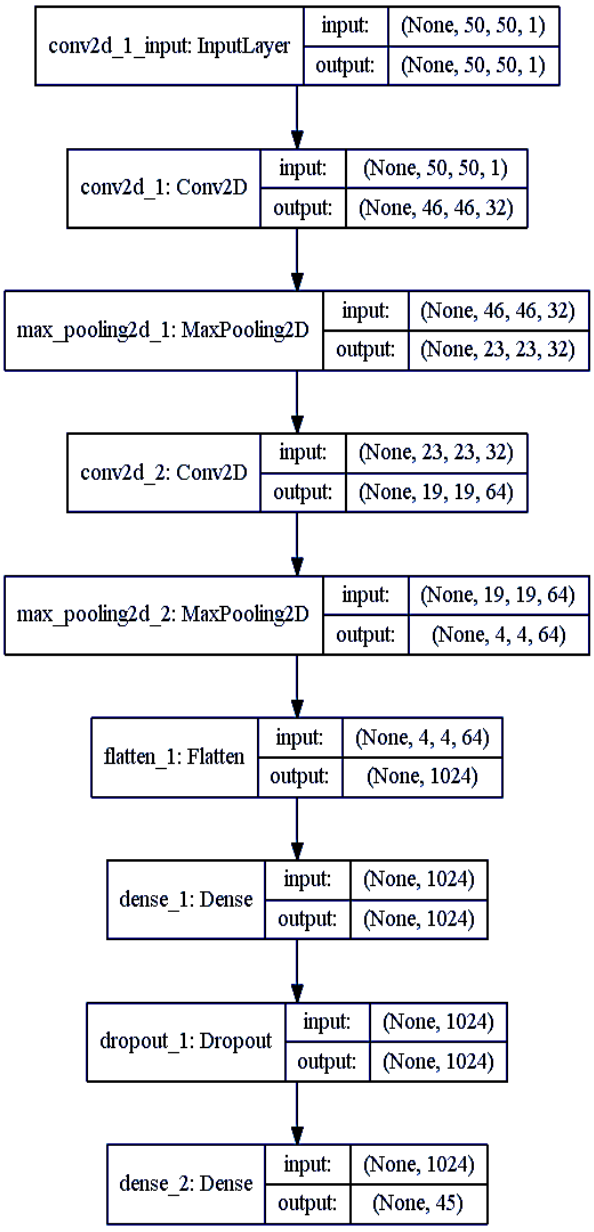

Fig 5: System Architecture of Sign language system

Figure 5 depicts the system architecture for the system under consideration. Figure 5 illustrates the flow chart used to determine the system performance metrics. Each letter is tested independently, with 50 iterations given to each letter. The frequency with which a letter is recognized is recorded below. However, the recognition may be incorrect because the letter gesture is not recognized.

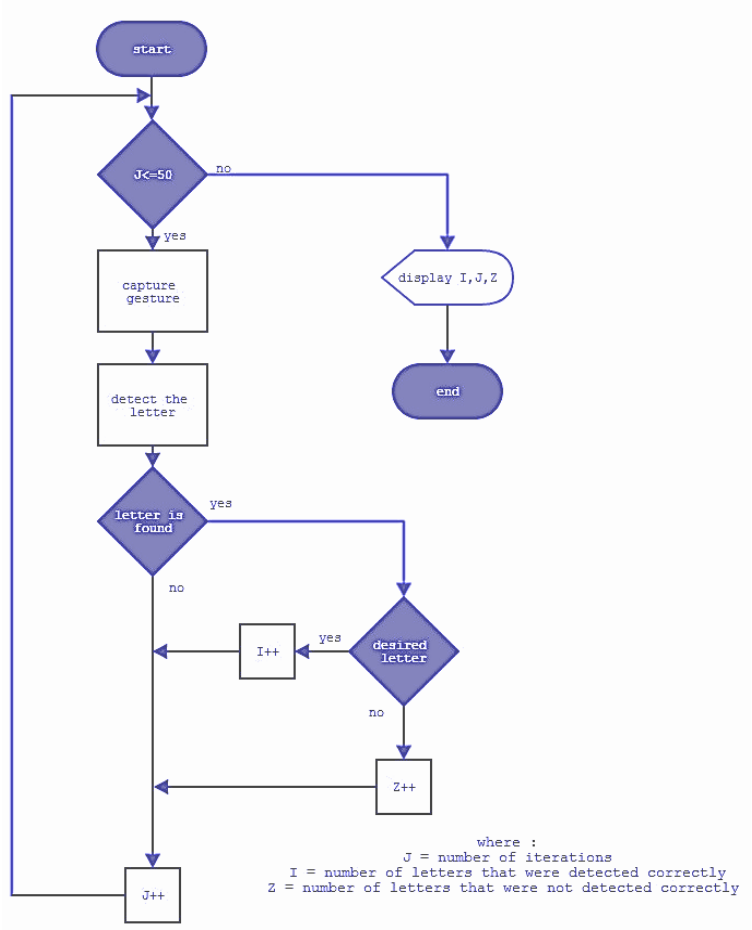

Fig.6: The testing procedure flowchart

4. Results and Discussion

The "Sign Language Recognition System using CNN and Neural Networks" is a system that aims to improve the independence of people with disabilities by recognizing sign language through a vision-based approach. The system uses hand gestures to represent signs and eliminates the need for any artificial devices for interaction. To create a dataset that matched the project requirements, the team found it difficult to find pre-made datasets in the form of raw images. Therefore, they decided to create their own dataset by capturing each frame shown by the machine's webcam using OpenCV. In each frame, they defined a region of interest (ROI) using a green bounded square. They then applied a Gaussian blur filter to the image to extract various features, resulting in an image that looks like the figure shown below.

Fig 7: Image After Applying Gaussian Blur

Accuracy and Error Calculation: Once we put together the hardware and uploaded the code we wanted onto the chip, we ran several tests to make sure everything was working as it should. We ran the tests multiple times and made adjustments as needed until we were confident that we met all of our requirements. To determine how accurate our results were, we used specific equations to calculate our error rates.

$$\text{Accuracy\%} = \frac{\text{Detected right}}{\text{No of iterations}} * 100$$

$$\text{Precision} = \frac{\text{True positive}}{\text{Actual Results}}$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{Predicted Results}}$$

$$\text{F1 score} = \frac{2 * (\text{precision} * \text{recall})}{\text{precision} + \text{recall}}$$

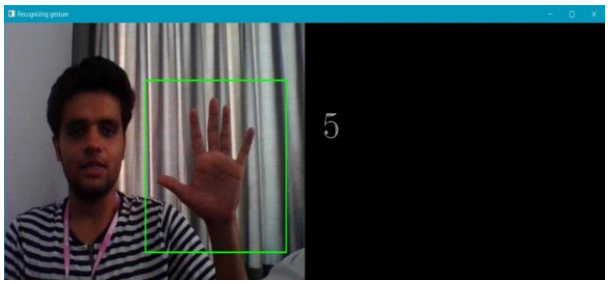

Fig 8: Gesture Recognition by CNN Model

For the observant reader who peruses Table I will be apparent that recognition errors come in two flavors: total misses and false positives, and likewise, imprecision while recognizing certain attributes may generate another type of errors. Flex sensor inaccuracies tend to occur when multiple hand gestures overlap each other in terms of finger positioning or movement characteristics leading to confusion between specific letters such as v and p which share relatively equal values for flexibility.

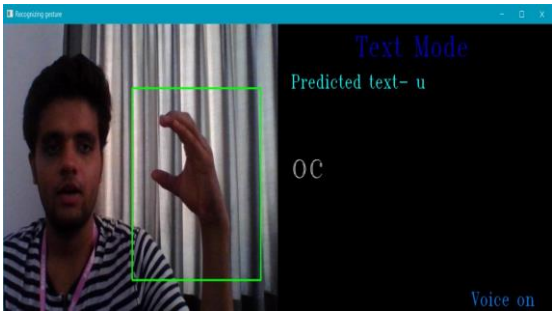

Fig 9:Gesture Recognition in Text Mode

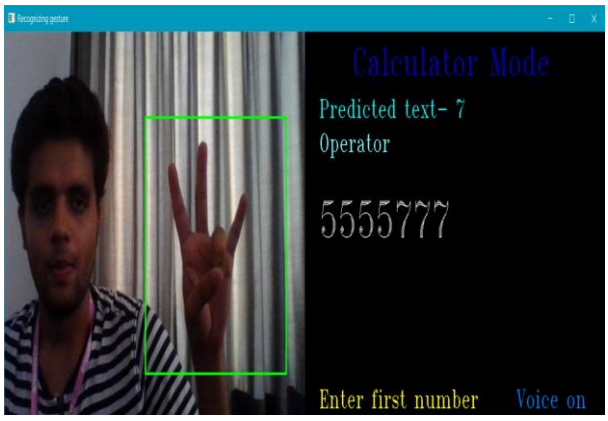

Fig 10: Gesture Recognition in Calculator Mode

Figure 9 and Figure 10 depict the Gesture Recognition in Text Mode and Calculator Mode, respectively.

Those who view Table I will notice there are two kinds of recognition errors—those missed entirely and others falsely classified despite detection. Moreover, ambiguity in recognizing certain aspects could result in a secondary form of errors. The issue of flex sensor inaccuracy arises when there are multiple gestures with overlapping finger positions or movements, so to illustrate this point about how some letters may be confused for others simply consider the case of the relatively similar flex values of v versus p.

Table I: Accuracy and Error Values.

Class type	Precision	Recall	F1 score	Support
0	0.99	0.99	0.99	195
1	1.00	1.00	1.00	205
2	0.99	0.99	0.99	198
3	0.99	0.99	0.99	231
4	0.99	0.99	0.99	193
5	0.99	0.99	0.99	193
6	0.99	1.00	0.99	214
7	1.00	1.00	1.00	209
8	0.99	0.99	0.99	185
9	1.00	1.00	1.00	212
10	0.99	1.00	0.99	199
11	0.99	0.99	0.99	183
12	0.99	0.97	0.98	188
13	0.97	0.99	0.98	195
14	1.00	1.00	1.00	203
15	0.99	1.00	1.00	226
16	1.00	1.00	1.00	187
17	0.98	1.00	0.99	191
18	1.00	0.99	0.99	191
19	0.98	0.99	0.99	172
20	0.99	0.99	0.99	190
21	1.00	0.98	0.99	222
22	1.00	1.00	1.00	210
23	0.99	1.00	1.00	198
24	1.00	1.00	1.00	210
25	1.00	0.95	0.98	211
26	1.00	0.99	0.99	204
27	0.99	0.99	0.99	200
28	0.99	1.00	0.99	198
29	1.00	0.99	0.99	188
30	0.99	0.99	0.99	197
31	0.99	1.00	0.99	184
32	0.99	1.00	0.99	198
33	1.00	0.99	1.00	226

34	0.99	0.99	0.99	200
35	1.00	1.00	1.00	201
36	1.00	1.00	1.00	189
37	0.98	1.00	0.99	201
38	1.00	1.00	1.00	189
39	0.99	0.98	0.99	201
40	1.00	1.00	1.00	217
41	1.00	1.00	1.00	211
Accuracy			0.99	8400
Macro Avg	0.99	0.99	0.99	8400
Weighted Avg	0.99	0.99	0.99	8400

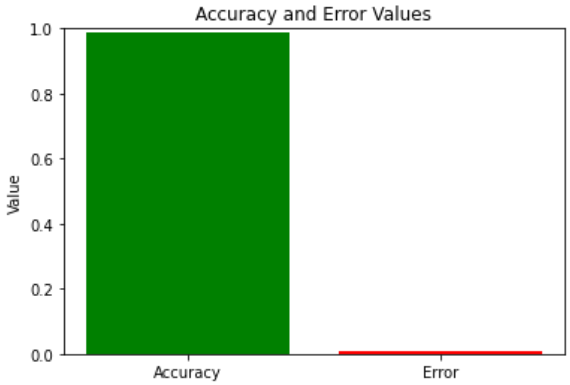

Fig. 11: Accuracy and Error values

The recognition system's accuracy and error values are presented for every class among 23 in Table I moreover there is a way to calculate the overall performance of each class that is available on the data set by computing various classification scores like precision-recall balance or f-score along with support.

To create a basic bar chart using given data, one must first define its relevant parameters such as accuracy and error in this case, while the graph displays metrics accuracy and error as labels along its horizontal axis whereas representing their respective values along vertical axis. A colored-green bar depicts accuracy while a colored-red one depicts error.

One can quickly visualize comparisons between the accuracy and error via the use of this chart99) and low error score (0.01), therefore it means that the system has produced good results. When comparing how well different models perform or how individual settings affect them at it's easiest when utilizing the chart.

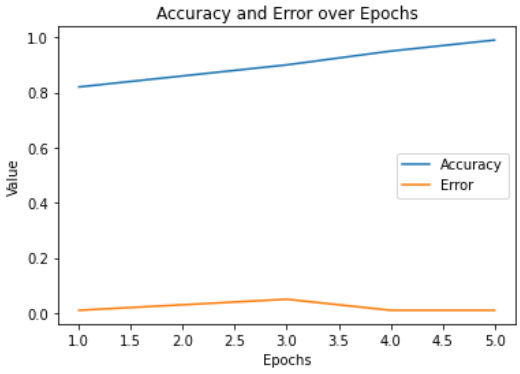

Fig 12 Accuracy and Error Epoches

Accuracy and error values over five epochs are displayed on a line graph created by this code while epochs are represented on the x-axis and accuracy and errors are shown on the y-axis. The representation of accuracy is depicted in blue while that of errors is shown in orange99 for accuracy readings and ideally maintained as low with an error value reading as low as The statement implies that there has been consistent high performance of the evaluated model in terms of precision with minimal errors during these epochs, and the inclusion of both axis labels and a legend in the graph's title enhances its interpretability allowing viewers to easily understand how well the model performed over five epochs.

5. Conclusion and Future Scope

The focus of our study was on building a real-time system that could identify hand motions to improve communication for individuals with hearing disabilities and the automatic recognition of sign gestures by our system using advanced algorithms combined with convolutional neural network technology allows for more intuitive and natural communication between individuals. Recognizing commonalities among human hands' shape is essential for the system's ability to identify different types of gestures. Using our system as a tool to improve accessibility and streamline communication is an effective way to save time for all parties involved, and our target moving forward is to improve upon the system's precision and speed as well as make it more accessible for users. Developing more advanced applications integrated with modern tech such as virtual and augmented reality is our aim in enhancing user experience. Furthermore, continued research and development of the hand gesture recognition system has the potential to change communication methods significantly for people with hearing difficulties as well as in other relevant fields.

References

[1] J. Padgett, R. (2014). The Contribution of American Sign Language to Sign-Print Bilingualism in Children. Journal of Communication Disorders, Deaf Studies & Hearing Aids, 02(02). https://doi.org/10.4172/2375-4427.1000108

[2] Evaluation of Measures to Facilitate Access to Care for Pregnant Deaf Patients: Use of Interpreters and Training of Caregivers in Sign Language. (2013). Journal of Communication Disorders, Deaf Studies & Hearing Aids, 01(01). https://doi.org/10.4172/2375-4427.1000103

[3] Novogrodsky, R., Fish, S., & Hoffmeister, R. (2014). The Acquisition of Synonyms in American Sign Language (ASL): Toward a Further Understanding of the Components of ASL Vocabulary Knowledge. Sign Language Studies, 14(2), 225–249. https://doi.org/10.1353/sls.2014.0003

[4] Ahmed, T. (2012). A Neural Network based Real Time Hand Gesture Recognition System. International Journal of Computer Applications, 59(4), 17–22. https://doi.org/10.5120/9535-3971

[5] Amer Kadhim, R., & Khamees, M. (2020). A Real-Time American Sign Language Recognition System using Convolutional Neural Network for Real Datasets. TEM Journal, 937–943. https://doi.org/10.18421/tem93-14

[6] Sign Language Translation Using Deep Convolutional Neural Networks. (2020). KSII Transactions on Internet and Information Systems, 14(2). https://doi.org/10.3837/tiis.2020.02.009

[7] Kumbhar, A., Sathe, V., Pathak, A., & Kodmelwar, M. K. (2015). Hand tracking in HCI framework intended for wireless interface. International Journal of Computer Engineering in Research Trends , 2(12), 821-824.

[8] Damatraseta, F., Novariany, R., & Ridhani, M. A. (2021). Real-time BISINDO Hand Gesture Detection and Recognition with Deep Learning CNN. Jurnal Informatika Kesatuan, 1(1), 71–80. https://doi.org/10.37641/jikes.v1i1.774

[9] K, S., & R, P. (2022). Sign Language Recognition System Using Neural Networks. International Journal for Research in Applied Science and Engineering Technology, 10(6), 827–831. https://doi.org/10.22214/ijraset.2022.43787

[10] M. M. Venkata Chalapathi, M. Rudra Kumar, Neeraj Sharma, S. Shitharth, "Ensemble Learning by High-Dimensional Acoustic Features for Emotion Recognition from Speech Audio Signal", Security and Communication Networks, vol. 2022, Article ID 8777026, 10 pages, 2022. https://doi.org/10.1155/2022/8777026

[11] Boinpally, A., Ventrapragada, S. B., Prodduturi, S. R., Depa, J. R., & Sharma, K. V. (2023). Vision-based hand gesture recognition for Indian sign language using convolution neural network. International Journal of Computer Engineering in Research Trends, 10(1), 1-9. https://doi.org/10.47577/IJCERT/2023/V10I0101

[12] Rudra Kumar, M., Rashmi Pathak, and Vinit Kumar Gunjan. "Diagnosis and Medicine Prediction for COVID-19 Using Machine Learning Approach." Computational Intelligence in Machine Learning: Select Proceedings of ICCIML 2021. Singapore: Springer Nature Singapore, 2022. 123-133.

[13] Alsaadi, Z., Alshamani, E., Alrehaili, M., Alrashdi, A. A. D., Albelwi, S., & Elfaki, A. O. (2022). A

Real Time Arabic Sign Language Alphabets (ArSLA) Recognition Model Using Deep Learning Architecture. Computers, 11(5), 78. https://doi.org/10.3390/computers11050078

[14] Rudra Kumar, M., Rashmi Pathak, and Vinit Kumar Gunjan. "Machine Learning-Based Project Resource Allocation Fitment Analysis System (ML-PRAFS)." Computational Intelligence in Machine Learning: Select Proceedings of ICCIML 2021. Singapore: Springer Nature Singapore, 2022. 1-14.

[15] Real time sign language detection system using deep learning techniques. (2022). Journal of Pharmaceutical Negative Results, 13(S01). https://doi.org/10.47750/pnr.2022.13.s01.126

[16] Amer Kadhim, R., & Khamees, M. (2020). A Real-Time American Sign Language Recognition System using Convolutional Neural Network for Real Datasets. TEM Journal, 937–943. https://doi.org/10.18421/tem93-14

[17] Kumar, P. ., Gupta, M. K. ., Rao, C. R. S. ., Bhavsingh, M. ., & Srilakshmi, M. (2023). A Comparative Analysis of Collaborative Filtering Similarity Measurements for Recommendation Systems. International Journal on Recent and Innovation Trends in Computing and Communication, 11(3s), 184–192.

[18] Kumar, P. ., Gupta, M. K. ., Rao, C. R. S. ., Bhavsingh, M. ., & Srilakshmi, M. (2023). A Comparative Analysis of Collaborative Filtering Similarity Measurements for Recommendation Systems. International Journal on Recent and Innovation Trends in Computing and Communication, 11(3s), 184–192.

[19] B, G. D., K, S. R., & P, V. K. (2023). SMERAS - State Management with Efficient Resource Allocation and Scheduling in Big Data Stream Processing Systems. International Journal of Computer Engineering in Research Trends, 10(4), 150–154. https://doi.org/10.22362/ijcertpublications.v10i4.5