Sign Language Recognition Using Custom CNN model

A Minor Project Report

in partial fulfillment of the degree

Bachelor of Technology
in
Computer Science & Engineering

By

19K41A0520	P. Shiva Sowmya
19K41A0532	B. Anu
19K41A0539	G. Anjali Kiran
19K41A0540	G. Sai Srujan

Under the Guidance of

Dr P Praveen

Submitted to

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
S.R.ENGINEERING COLLEGE (A),ANANTHASAGAR, WARANGAL
(Affiliated to JNTUH, Accredited by NBA)

May, 2022.

SR Engineering College
Innovation . Creativity . Entrepreneurship

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the Minor Project Report entitled "Sign Language Recognition Using Custom CNN model" is a record of bonafide work carried out by the student(s) Shiva Sowmya, Anu, Anjali Kiran, Sai Srujan bearing Roll No(s) 19K41A0520, 19K41A0532, 19K41A0539, 19K41A0540 during the academic year 2021-2022 in partial fulfillment of the award of the degree of ***Bachelor of Technology*** in **Computer Science & Engineering** by the Jawaharlal Nehru Technological University, Hyderabad.

Supervisor **Head of the Department**

External Examiner

ABSTRACT

Normal humans can easily interact and communicate with one another, but the person with hearing and speaking disabilities face problems in communicating with other hearing people without a translator. Sign Language Recognition has emerged as one of the important area of research in Computer Vision. Hand signs are an effective form of human-to-human communication that has a number of possible applications. Being a natural means of interaction, they are commonly used for communication purposes by speech impaired people worldwide. In fact, about one percent of the Indian population belongs to this category. This is the key reason why it would have a huge beneficial effect on these individuals to incorporate a framework that would understand Indian Sign Language. The Sign Language is a barrier of communication for deaf and dumb people. This is the reason that the implementation of a system that recognize the sign language would have a significant benefit impact on dumb - deaf people. In this paper, a method is proposed for the automatic recognition of the alphabet. Here, the sign in the form of gestures is given as an input to the system. Further various steps are performed on the input sign image. People with hearing and speaking disability are highly dependent on non-verbal form of communication that involves hand gesture. One of the vital challenges that deaf people face in the present society is communication. The use of sign language translators is not a viable solution. Recently Deep learning models showcased outstanding results in classification and recognition systems. So, researchers are focusing more on developing deep learning models with at most accuracy. The proposed system comprises of three stages: Preprocessing, Feature Extraction and Classification. The study aims to represent a Convolutional neural network-based Indian Sign Language identification model for deaf persons and further convert it into the text, which is more accurate than prior studies. In this study, a CNN-based, ISL hand- gesture recognition model is developed, which outperforms many other existing models. A custom dataset was prepared, to train the model. The developed model can recognize 26 alphabets represented through hand gestures in ISL. We found that the model produced an accuracy of 98%.

CONTENTS

ABSTRACT		*iii*
Chapter No.	**Title**	**Page No.**
1	**INTRODUCTION**	
	1.1 INTRODUCTION	06
	1.2 EXISTING SYSTEMS	08
	1.3 PROPOSED SYSTEM	10
2	**LITERATURE SURVEY**	
	2.1 LITERATURE SURVERY	11
	2.2 SYSTEM STUDY	15
3	**DESIGN**	
	3.1 REQUIREMENT SPECIFICATION	
	3.1.1 Functional Requirements	16
	3.1.2 Non Functional Requirements	16
	3.1.3 Hardware and Software Requirements	16
	3.2 UML DIAGRAMS	
	3.2.1 Dataflow diagram	17
	3.2.2 Use Case diagram	17
	3.2.3 Sequence diagram	19
4	**IMPLEMENTATION**	
	4.1 MODEULES	
	4.1.1 Data Collection	20
	4.1.2 Data Preprocessing	20
	4.1.3 CNN Model	21
	4.2 OVERVIEW TECHNOLOGY	
5	**TESTING**	24
6	**RESULTS**	
	6.1 DATASET	26
	6.2 PROCEDURE	28
7	**CONCLUSION**	30
8	**FUTURE SCOPE**	30
9	**BIBLIOGRAPHY**	31

1. INTRODUCTION

1.1 Introduction

Communication is the activity of conveying information. But from survey it's known that nearly 6.3% of the population in India, are not blessed with the power of communication. Most of them neither can speak nor are able to hear which limits their interactions. This special type of people express their feelings through Sign Language. Indian Sign Language is used by the people of India and it is standardized so as to be common all over the nation. Indian Sign Language is communicated using hand gestures made by Single hand and Double hands. In this, grammar usage is not considered and only main words are taken into consideration. Indian Sign Language uses Single hand and Double hand gestures for communication.

To aid in the teaching and interpretation of sign language, a variety of programs have been developed. However, progress in recognizing sign languages with modern technology has been promising but very limited. Software that can recognize and interpret sign language is in high demand. It can also act as the bridge between sign language users and non-sign language users. As a result, it is critical to establish the link between the deaf and the spoken by creating a deep learning model to translate sign language to text. Humans can communicate with others in different ways for example, speaking in some languages like Telugu, Hindi and many more. Sign language Recognition will reduce the communication gap between normal people and deaf or muted people.

Sign language is the combination of static and dynamic gestures. Static hand gestures don't require any hand movement. The 26 alphabets, ten digits, and a few static word signs make up sign language. Each country employs its own as there is no universally acknowledged sing language. Sign languages include American Sign Language, British Sign Language, Indian Sign Language, and many more. Even though Indian sign language is used by the majority of deaf people in India, it is not widely employed in schools. Many organizations that work with deaf people have made efforts to promote Indian Sign Language, but most people are unaware of these signals due to their complexity, making communication between the deaf and the spoken difficult. Throughout the previous few decades, studies have predominantly focused on the identification of American Sign Language. American Sign Language represents the alphabet with a single hand, whereas Indian Sign Language (ISL) represents the alphabet with both hands.

The sign language is used widely by people who are deaf-dumb; these are used as a medium for communication. A sign language is nothing but composed of various gestures formed by different shapes of hand, its movements, orientations as well as the facial expressions. These gestures are generally used by deafdumb people in order to express their thought. Dumb-deaf persons faces communication barrier in public places while interacting with normal person, such as in bank, hospital and post offices. Sometimes the deaf needs to seek the help of the sign language interpreter so as to translate their thoughts to normal people and vice versa. However, this way turns out to be very costly and does not work throughout the life period of a deaf person. So a system which can automatically recognize the sign language gestures becomes a necessity. Introducing such a system would lead to minimize the gap between deaf and normal people in the society. The sign language in use at a particular place depends on the culture and spoken language at that place. Indian sign language (ISL) is used by the deaf community in India. ISL is a standard and well-developed way of communication for hearing impaired people in India and speaking in English. Different symbols are involved for different alphabets for Indian Sign Language. It consists of both word level gestures and finger spelling. This paper presents a method for the automatic recognition of the static gestures in the Indian sign language alphabet.

The hearing impaired people becomes neglected from the society because the normal people never try to learn ISL nor try to interact with the hearing impaired people. This becomes a curse for them and so they mostly remain uneducated and isolated. Thus recognition of sign language was introduced which has not only been important from English.Sign Languages are a set of languages that use predefined actions and movements to convey a message. These languages are primarily developed to aid deaf and other verbally challenged people. They use a simultaneous and precise combination of movement of hands, orientation of hands, hand shapes etc. Different regions have different sign languages like American Sign Language, Indian Sign Language etc. We focus on Indian Sign language in this project.

Indian Sign Language (ISL) is a sign language that is predominantly used in South Asian countries. It is sometimes referred to as Indo-Pakistani Sign Language (IPSL). There are many special features present in ISL that distinguish it from other Sign Languages. Features like Number Signs, Family Relationship, use of space etc. are crucial features of ISL. Also, ISL does not have any temporal inflection.

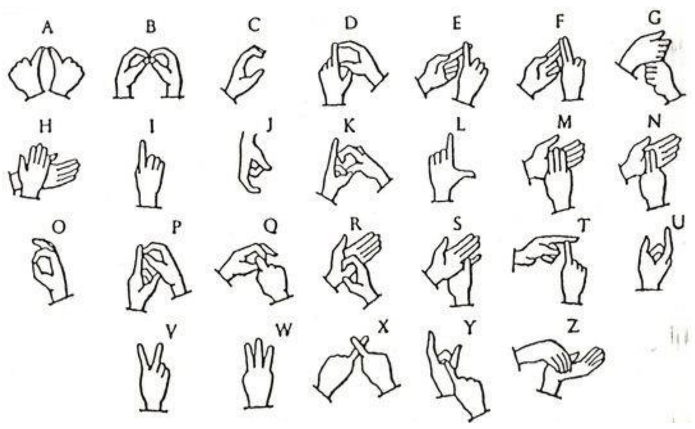


Figure 1. Alphabets in Indian Sign Language

In this project, we aim towards analyzing and recognizing various alphabets from a dataset of sign images. Database consists of various images with each image clicked in different light condition with different hand orientation. With such a divergent data set, we are able to train our system to good levels and thus obtain good results. It can be observed that the double handed Indian sign language patterns are very similar to each other, as an eg., the patterns of letter E, I & O are very similar. Therefore, classification of these patterns is little challenging.

1.2 Existing System

Primarily we have two existing methodologies. They are Glove Based System Image Processing and Recognition. Many early Sign Language Recognition systems used datagloves and accelerometers to acquire details of the hands. The measurements were measured directly using Data Glove. In glove-based system, person has to wear a glove which consists of sensors like flex sensor, it tracks the motion of the hand. It compares with the data, which is received directly from sensors based on our finger movement. It consists of 5 Flex sensors and accelerometer to recognize the hand and palm movements at the input. The accelerometer detects the rotation of palm which selects the language of the communication forming the 1st bit of binary number to be compared in the lookup table. The flex sensor has a Resistive strip which can be folded. The resistance varies as the flex strip is folded to indicate either logical 1 or logical 0 by giving the variable analog resistive Input to inbuilt ADC of ARM7 microcontroller. In this way, 6 bit binary digit will be formed for each gesture. In totality 32 such gestures are possible due to 5 flex sensors. These digits are stored in the external memory of ARM7. The microcontroller will then compare these readings to the look up table stored in the internal program memory. Whichever reading is closest to the lookup table ARM7 will then select If the user hand is showing the following gesture, then according to the flex sensors reading, the ARM7 will search the binary digit for the word. After this the ARM7 will search the audio file with the name good. If the file exists in the external memory, then µC will play the corresponding audio file with the word pronunciation. Here we are using two keys i.e. English and Hindi/Marathi, if user wants to play English/Hindi /Marathi words then select those keys. While it gave the advantage of precise positions, they didn't permit full normal development and contracted the versatility of the signer, adjusting the signs performed. Preliminaries with an altered glove-like device, which was less tightening, attempted to resolve this issue. However, because of the restrictive expenses of such methodologies, the utilization of vision has become more famous.

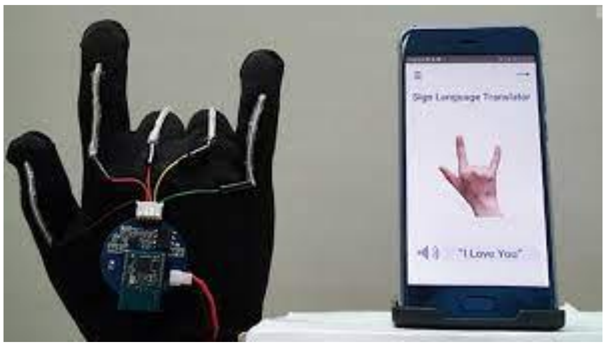


Figure 2.Example of Data Glove

Image processing and recognition model uses camera to capture the images and it compares the given data with images existing and recognizes the sign. In this system, still hand image frame is captured using a webcam. These frames are processed to get enhanced features. Then feature extraction and classification algorithms are used to translate the sign language into English text. This translation is converted to speech using text to speech API.

Before performing feature extraction, the images must be processed in such a way that only the useful information is considered and the redundant, distracting noise and superficial data are neglected. The images are first converted to 100 x 100 pixel size for faster computations. The image is then converted to grayscale and finally transform into a binary image. Simultaneously, skin color is detected using YCbCr model. Finally, edge detection is performed using Canny edge detector.

Recently Microsoft Kinect has offered a reasonable depth camera. Because of its capabilities, Kinect is now widely employed by scientists. Kinect is capable of delivering both colour and depth video streams at the same time. Background segmentation is simple to accomplish with depth data. Kinect is used to recognise signs. At present there are no data sets accessible and as such the outcomes are limited.

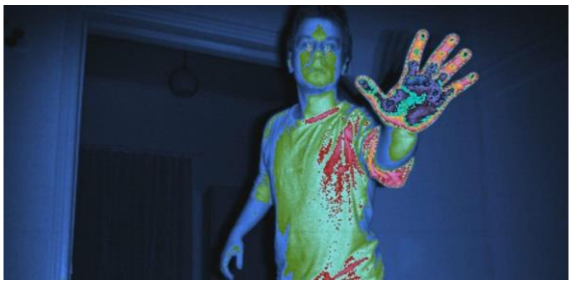


Figure 3.Example of Kinect color and depth video stream

Recently Microsoft Kinect has offered a reasonable depth camera. Because of its capabilities, Kinect is now widely employed by scientists. Kinect is capable of delivering both color and depth video streams at the same time, and background segmentation is simple to accomplish with depth data. Kinect is used to recognize signs. Recently there are no data sets accessible as the outcomes are limited.

1.1 Proposed System

For Indian Sign Language recognition, to obtain good accuracy we have collected our own dataset through webcam using open CV. To collect the dataset, we have created two folders, train and test folders. The images present in train and test folder are used for training and testing the built model.

The train and test folders contain the sub folders, named as A, B, C…. Z. These are the classes we have to predict. Any image which is captured belongs to one of these classes. Folder A consists of images in which, alphabet A is represented in Indian Sign Language. And all other folders also consist of their respective images.

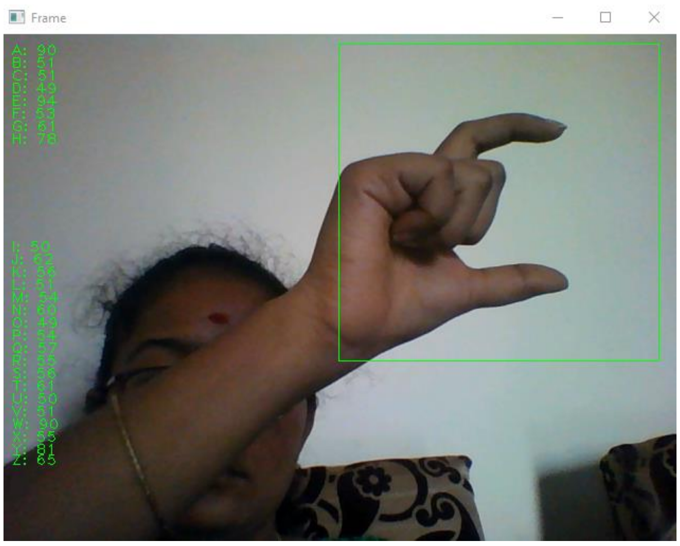


Figure 4.Dataset collection

Fig.4 is the output display screen of data collection. The green square shown in the figure is region of interest, the sign which we represent with our hands should be placed in that region itself. Only the image covered in that region will be saved. While capturing the image, to save the image into its corresponding folder we have to press the alphabet to which it belongs, through keyboard. In the left side of figure 3 we have labels like A: 90, B: 51 and so on, these are the count of images present in each folder. According to figure 3, we have 90 images in folder A, 51 images in folder B, etc. In the same way data is collected for both testing and training.

We developed a custom CNN model, if we divide the whole model into two layers: top layer and bottom layer. Bottom layer consists of convolutional layers and top layer is fully connected network. Bottom layer consists of pair of three convolutional and pooling layers. And top layer consists of 5 hidden layers and one output layer with 26 neurons (because the model has to classify 26 alphabets).

For our proposed model we have created custom dataset of 26 alphabets which are signed in Indian sign language. 250 images are collected for each alphabet.

2. LITERATURE SURVEY

Over past few years, many studies have been taken place on Indian sign language recognition. Research in sign language primarily has two approaches one is image recognition and the other is Glove based System. Image recognition uses a Camera to capture the images or video and compare the data with images and recognize the image. Another approach is a Glove based System in which the user needs to wear a glove that consists of a sensor and motion tracker. It compares with the data directly from the sensors based on our finger movements.

- Sirshendu Hore, Sankhadeep Chatterjee, V. Santhi, Nilanjan Dey, Amira S. Ashour, Valentina Emilia Balas and Fuqian Shi, to find a solution for Indian Sign Language Recognition they used 3 novel methods. They combined neural networks(NN) with Generic Algorithm(GA), Evolutionary Algorithm (EA) and Particle Swarm Optimization (PSO) separately to attain minimum error. Among these 3 methods NN-PSO performed well than other approaches with 99.96 accuracy, 99.98 precision, 98.29 recall, 99.63 F-Measure and 0.9956 Kappa Statistic.

- Asish Sharma and Nikitha Sharma inorder to proporse a solution for Indian Sign Language Recogniton, initially they considered 3 models. VGG 16 with finetuning, VGG 16 with transfer learning and hierarchical neural network, these 3 models were considered. They trained the self developed dataset consisting images of Indian Sign Language (ISL) representation of all 26 English alphabets with these three models. Among the 3, hierarchical model performed well with 98.52% accuracy for one-hand and 97% for two-hand gestures.

- Britta Bauer and Hienz proposed a system that is Video-based continuous Sign Language Recognition Using Statistical Methods. Here they used HMM for recognition. For 97 signs this system accomplishes a precision of 91.7%. But the drawback is the user needs to wear simple coloured cotton gloves.

- Kartik, Sumanth, Sri Ram, Prakash proposed a model based on Convolutional Neural Networks for image depiction and classification. The dataset in the proposed model consists of over 87,000 images. The proposed model was well trained on 78,300 images and the testing was performed on 8700 images in the ratio of 9:1. The model produced a training accuracy of 98.67%.

- Starner proposed an HMM-based recognition system with his vision system for recognition of American Sign Language. It recognizes 40 words with 91.3% accuracy.

- Truong, Kai Yang, Tran proposed a translator that automatically detects signs of alphabets in American Sign Language. They used AdaBoost and Haar Cascades classifiers. The translator was trained using a dataset of over 28000 samples of hand sign images. The translator achieved a precision of 98.7%.

- Lean Karlo S. Tolentino, Ronnie O. Serfa Juan, August C. Thio-ac, Maria Abigail B. Pamahoy, Joni Rose R. Forteza, and Xavier Jet O. Garcia developed a Static Sign Language Recognition system which is based on sign color modeling technique, i.e., explicit skin-color space thresholding. The skin-color range is predetermined that will extract pixels (hand) from non–pixels (background). The images were fed into the model called the Convolutional Neural Network (CNN) for classification of images. Keras was used for training of images. The system acquired a accuracy of 93.67%, of which 90.04% was attributed to ASL alphabet recognition, 93.44% for number recognition and 97.52% for static word recognition.

- Boris Mocialov , Graham Turner , Katrin Lohan , Helen Hastie proposed a system i.e., Continuous sign language recognition. They followed a approach which combines heuristics for segmentation of the video stream by identifying the epenthesis with stacked LSTMs for automatic classification of the derived segments. This approach segments continuous stream of video data with the accuracy of over 80% and reaches accuracies of over 95% on segmented sign recognition.

- Raheja, J. L.; Mishra, A.; Chaudhary developed dynamic hand signal acknowledgment procedures for realtime scenes,i.e.,the caught video was changed over to HSV shading space for preprocessing and afterward division was done in view of skin pixels. All this was done by Support Vector Machine. Results were 97.5% accurate. This approach can also be used for implantation for other regional languages.

- Sharvani Srivastava, Amisha Gangwar, Richa Mishra, Sudhakar Singh proposed a method to create an Indian Sign Language dataset using a webcam and then using transfer learning, train a TensorFlow model to create a real-time Sign Language Recognition system .The system has been trained on the Indian Sign Language alphabet dataset. For information obtaining, pictures have been caught by a webcam utilizing Python and OpenCV which makes less expensive. The created framework is showing certainty of 85.45%.

- M. V. D. Prasad, P. V. V. Kishore, D. Anil Kumar, Ch. Raghava Prasad proposed a work on Fuzzy classifying of continuous sign language videos with simple backgrounds with tracking and shape combined features.Methods/Analysis: Tracking and capturing hand position vectors is the artwork of horn schunck optical flow algorithm. Active contours extract shape features from sign frames in the video sequence. The two most dominant features of sign language are combined to build sign features. This feature matrix is the training vector for Fuzzy Inference Engine (FIS). The target vector for training is a text coded messages for each sign. Windows API transforms text into speech. With continuous testing with different sign videos other than used for training produced an average word matching score of around 92.5%.

- Liang-Guo Zhang,Yiqiang Chen,Gaolin Fang,Xilin Chen presented a vision-based medium vocabulary Chinese sign language recognition (SLR) system .The proposed recognition system consists of two modules. In the first module, techniques of robust hands detection, background subtraction and pupils detection are efficiently combined to precisely extract the feature information with the aid of simple colored gloves in the unconstrained environment. In the second part, a Tied-Mixture Density Hidden Markov Models (TMDHMM) framework for SLR is proposed.Proposed methods can work well for the medium vocabulary SLR in the environment without special constraints and the recognition accuracy is up to 92.5%.

- Heung-Il Suk,Bong-Kee Sin,Seong-Whan Lee proposed a method for recognizing hand gestures in a continuous video stream using a dynamic Bayesian network or DBN model. The proposed method of DBN-based inference is preceded by steps of skin extraction and modelling, and motion tracking.They have also developed a DP-based real-time decoding algorithm for continuous gesture recognition ,obtained a recognition rate upwards of 99.59% with cross validation. In the case of recognizing continuous stream of gestures, it recorded 84% with the precision of 80.77% for the spotted gestures.

- Nikolaos Adaloglou,Theocharis Chatzis,Ilias Papastratis,Andreas Stergioulas developed a comparative experimental assess-ment of computer vision-based methods for sign language recognition. By implementing the most recent deep neuralnetwork methods in this field, a thorough evaluation on multiplepublicly available datasets is performed. The aim of the presentstudy is to provide insights on sign language recognition, focusingon mapping non-segmented video streams to glosses. For thistask, two new sequence training criteria, known from the fields ofspeech and scene text recognition, are introduced. Finally,a new RGB+D dataset for the Greek sign language is created.

- Siming He[4] proposed a system having a dataset of 40 common words and 10,000 sign language images. To locate the hand regions in the video frame, Faster R-CNN with an embedded RPN module is used. It improves performance in terms of accuracy. Detection and template classification can be done at a higher speed as compared to single stage target detection algorithm such as YOLO. The detection accuracy of Faster R-CNN in the paper increases from 89.0% to 91.7% as compared to Fast-RCNN. A 3D CNN is used for feature extraction and a sign-language recognition framework consisting of long and short time memory (LSTM) coding and decoding network are built for the language image sequences.On the problem of RGB sign language image or video recognition in practical problems , the paper merges the hand locating network, 3D CNN feature extraction network and LSTM encoding and decoding to construct the algorithm for extraction. This paper has achieved a recognition of 99% in common vocabulary dataset.

- Let's approach the research done by Rekha, J[5]. which made use of YCbCr skin model to detect and fragment the skin region of the hand gestures. Using Principal Curvature based Region Detector, the image features are extracted and classified with Multi class SVM, DTW and non-linear KNN. A dataset of 23 Indian Sign Language static alphabet signs were used for training and 25 videos for testing. The experimental result obtained were 94.4%for static and 86.4% for dynamic.

- In [6] , a low cost approach has been used for image processing . The capture of images was done with a green background so that during processing, the green colour can be easily subtracted from the RGB colourspace and the image gets converted to black and white. The sign gestures were in Sinhala language. The method that thy have proposed in the study is to map the signs using centroid method. It can map the input gesture with a database irrespective of the hands size and position. The prototype has correctly recognised 92% of the sign gestures.

- The paper by M. Geetha and U. C. Manjusha[7], make use of 50 specimens of every alphabets and digits in a vision based recognition of Indian Sign Language characters and numerals using B-Spine approximations. The region of interest of the sign gesture is analysed and the boundary is removed . The boundary obtained is further transformed to a B-spline curve by using the Maximum Curvature Points(MCPs) as the Control points. The B-spline curve undergoes a series of smoothening process so features can be extracted. Support vector machine is used to classify the images and the accuracy is 90.00%.

- In [8], Pigou used CLAP14 as his dataset [9]. It consists of 20 Italian sign gestures.After preprocessing the images , he used a Convolutional Neural network model having 6 layers for training. It is to be noted that his model is not a 3D CNN and all the kernels are in 2D. He has used Rectified linear Units (ReLU) as activation functions. Feature extraction is performed by the CNN while classification uses ANN or fully connected layer. His work has achieved an accuracy of 91.70% with an error rate of 8.30%.

- A similar work was done by J Huang [10].He created his own dataset using Kinect and got a total a total of 25 vocabularies which are used in everyday lives. He then applied a 3D CNN in which all kernels are also in 3D. The input of his model consisted of 5 important channels which are colour-r, colour-b, colour-g, depth and body skeleton. He got an average accuracy of 94.2%

- Another research paper on Action recognition topic by the author J.Carriera [11] shares some similarities to sign gesture recognition .He used a transfer learning method for his research As his pre-trained dataset, he used both ImageNet[12] and Kinetic Dataset [9] . After training the pertained models using another two datasets namely UCF-101 [13] and HMDB-51 [14], he then merged the RGB model, flow model, pre-trained Kinetic and pre-trained ImageNet.The accuracy he got on UCF-101 dataset is 98.0% and on HMDB-51 is 80.9%

2.1 System Study

In recent advancements in the field of deep learning, neural networks offer a variety of applications for understanding sign languages. Deep Learning models showcased prominent results in recognition Systems. So, researchers are focusing more on developing neural network models. We have created our own dataset, that consist of over 6500 images. We have resized the image 224 x 224 pixels and a RGB image is transformed to grayscale image in pre-processing phase.

Then the images are subjected to CNN algorithm which translates the sign to text. The CNN algorithm consists of several layers and used adam and relu optimizers. We have found that our model gains accuracy over 98%.

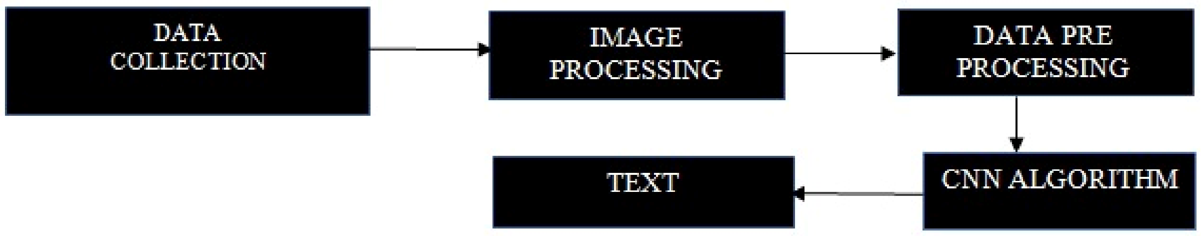


Figure 5. Flow chart

3.DESIGN

3.1 Requirements Specifications

3.1.1 Functional Requirements

Functional requirements are the functions that systems should give to their users. Even though there are two people participating, only one of them interacts with the system directly, hearing-impaired person who signs in front of the system, the system then translates it into text. The other person is a passive user because he does not interact with the system directly, just reads signs. The signs are made by a deaf or dumb individual. Normal user reads the signs. A person who is deaf or dumb should be able to make signs that represent alphabet letters.

3.1.2 Non Functional Requirements

Non-functional requirements are the circumstances in which the system should work, and they are as follows:

Real-time: The system should recognize signs and translate them into text in short period of time for users. The system should be able to recognize objects in real time and with high accuracy in low light.

Accuracy: signs should not be mixed together, and the system should be able to recognize those signs accurately and properly.

Usability: Users should be able to engage with the system in a natural way. The deaf or dumb just needs to perform signs in front of the system.

Reliability: The system is reliable as it experiences very less failure. Rarely the system may face some technical glitches but it can recoverable quickly.

3.1.3 Hardware and Software Requirements

Hardware Requirements

• **System**	**:** Intel Core i3, i5, i7 and 2GHz Minimum
• **Ram**	**:** 4GB or above
• **Hard Disk**	**:** 10GB or above
• **Input**	**:** Camera
• **Output**	**:** Monitor or PC

Software Requirements

• **OS**	**:** Windows 8 or Higher versions
• **Platform**	**:** Jupyter Notebook, Visual Studio
• **Program Language**	**:** Python

3.2 UML DIAGRAMS

3.2.1 DATAFLOW DIAGRAMS

Data flow diagrams are used to graphically represent the flow of data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation. Data flow diagrams can be divided into logical and physical. The logical data flow diagram describes flow of data through a system to perform certain functionality of a business. The physical data flow diagram describes the implementation of the logical data flow. The data flow diagram shows data inputs, outputs, storage sites, and paths between each destination using predetermined symbols and shapes like rectangles, circles, and arrows, as well as short text labels. Data flow diagrams can range from simple, even hand-drawn process overviews to multi-level and in depth that go deeper into how data is processed. The main components or symbols in the data flow diagram include:

Components of Data Flow Diagram:

• **Process**: A circle is a process in data flow diagrams and depicts how the data is handled and processed in the system.

• **Data Flow**: The data flow is the curved line that shows the flow of data in or out of the system.

• **Data Store**: A data store denotes the storage of information that can be retrieved later or by other processes in a different order. A single element or a set of elements can be found in the data storage. The group of parallel lines denotes a location to collect the data items.

• **Entity**: An external entity that serves as a source of system inputs or a sink of system outputs is called a source or sink.

DFD – 0 LEVEL:

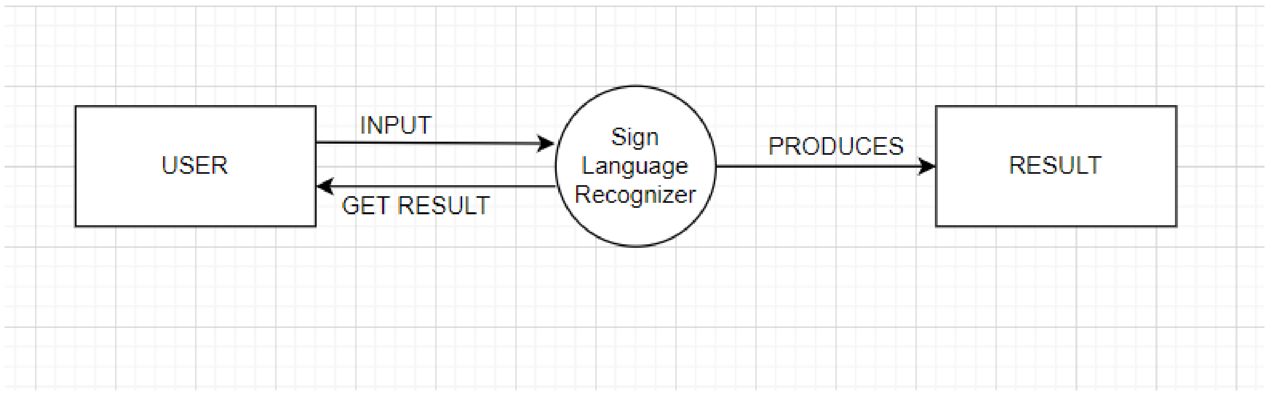


Figure 6.Data Flow diagram for SLR

3.2.2 USE CASE DIAGRAM

In the Unified Modeling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you'll use a set of specialized symbols and connectors. Use case diagrams are a way to capture the system's functionality and requirements in UML diagrams. It captures the dynamic behavior of a live system. A use case diagram consists of a use case and an actor. A use case represents a distinct functionality of a system, a component, a package, or a class. An actor is an entity that initiates the use case from outside the scope of a use case. The name of an actor or a use case must be meaningful and relevant to the system.

A purpose of use case diagram is to capture the core functionalities of a system. UML use case diagrams are ideal for:

▪ Representing goals of system-user interactions

▪ Defining and organizing functional requirements in a system

▪ Specifying the context and requirements of a system

▪ Modelling the basic flow of events in a use case

Components of Use Case Diagram:

• **Actors:** The users that interact with a system. An actor can be a person, an organization, or an outside system that interacts with your application or system. They must be external objects that produce or consume data.

• **System:** A specific sequence of actions and interactions between actors and the system. A system may also be referred to as a scenario.

• **Goals:** The end result of most use cases. A successful diagram should describe the activities and variants used to reach the goal

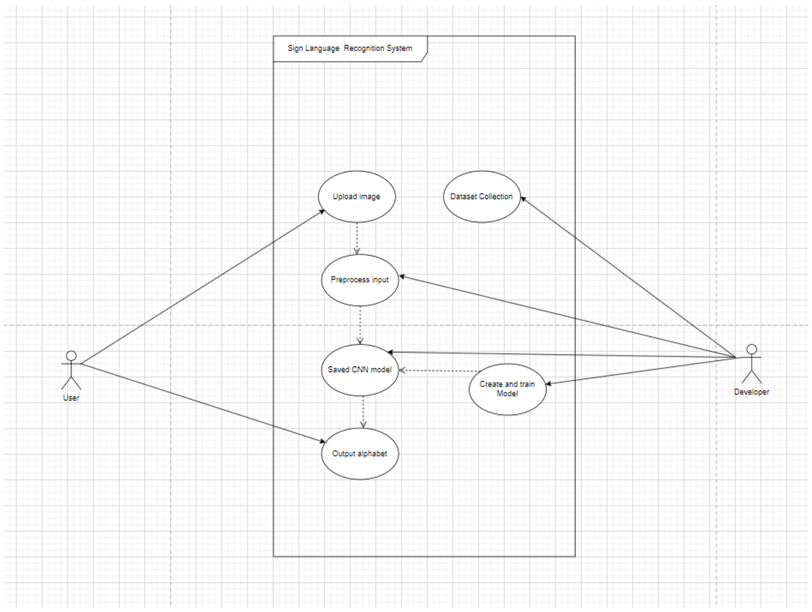


Figure 7.Use case diagram for SLR

3.2.3 SEQUENCE DIAGRAM

A sequence diagram is a type of interaction diagram because it describes how—and in what order—a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process. Sequence diagrams are sometimes known as event diagrams or event scenarios. The sequence diagram represents the flow of messages in the system and is also termed as an event diagram. It portrays the communication between any two lifelines as a time-ordered sequence of events, such that these lifelines took part at the run time. In UML, the lifeline is represented by a vertical bar, whereas the message flow is represented by a vertical dotted line that extends across the bottom of the page. It incorporates the iterations as well as branching.

Components of Sequence Diagram:

• **Object symbol:** Represents a class or object in UML. The object symbol demonstrates how an object will behave in the context of the system. Class attributes should not be listed in this shape.

• **Activation box:** Represents the time needed for an object to complete a task. The longer the task will take, the longer the activation box becomes.

• **Package symbol:** Used in UML notation to contain interactive elements of the diagram. Also known as a frame, this rectangular shape has a small inner rectangle for labeling the diagram.

• **Lifeline symbol:** Represents the passage of time as it extends downward. This dashed vertical line shows the sequential events that occur to an object during the charted process. Lifelines may begin with a labeled rectangle shape or an actor symbol.

• **Option loop symbol:** Used to model if/then scenarios, i.e., a circumstance that will only occur under certain conditions.

• **Alternative symbol:** Symbolizes a choice (that is usually mutually exclusive) between two or more message sequences. To represent alternatives, use the labeled rectangle shape with a dashed line inside.

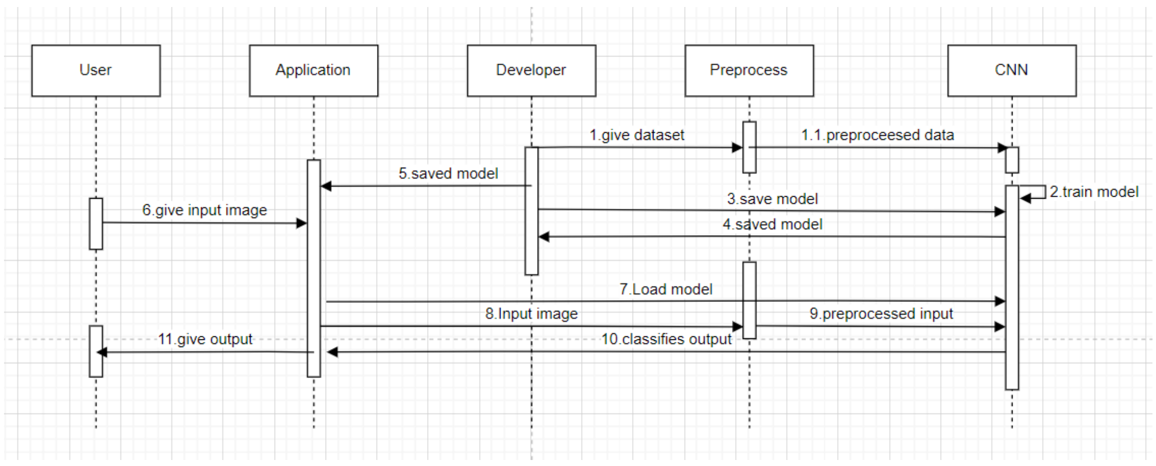


Figure 8.Sequence diagram for SLR

4. IMPLEMENTATION

4.1 MODULES

4.1.1 Data Collection

To achieve high accuracy in Indian Sign Language identification, we have created our own dataset using open CV and a webcam. We generated two folders, one for training and one for testing, to collect the data. The photos in the train and test folders are used to train and test the model that has been built.

The train and test folders have subfolders labelled A, B, C, and so on. These are the classes we must anticipate. Any image captured falls into one of these categories. Folder A contains images that depict the alphabet A in Indian Sign Language. All of the other folders have their images in them as well.

4.1.2 Data Preprocessing

Before storing the image into its respective folder, the image is resized to 224 x 224 pixels and a RGB image (colored image) is transformed to grayscale image.

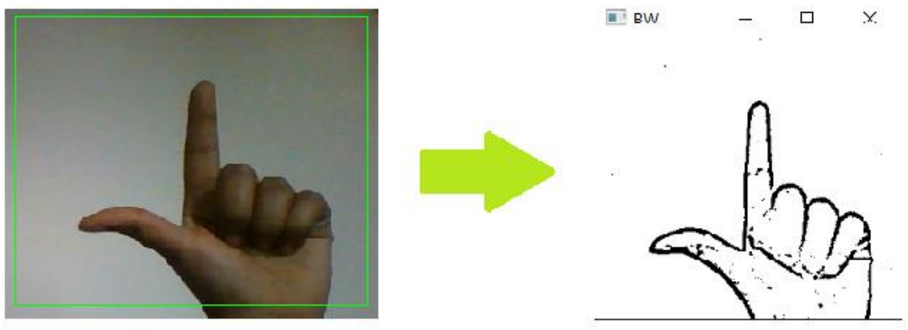


Figure 9. Image Preprocessing

As shown in the figure 9, after capturing the image through webcam, the RGB (colored image) is transformed into a grayscale image. In the grayscale image, the outline of the hand gesture is produced using the Gaussian blur approach and by applying threshold for the image.
Normalization, a preprocessing technique is applied to all the images before training or testing the model. Using this technique all pixel values in an image are ranged between 0 to 1. This is done through rescaling the image. To rescale the image, each pixel value is divided by 225, in an image the maximum pixel value is 225.

4.1.3 Convolutional Neural Network (CNN) Model :

Convolutional Neural Networks also known as CNNs which are most widely used for image classification problems. As compared to Artificial Neural Network(ANN) models, Convolutional Neural Network models are more efficient in feature extraction for images. When it comes to image classification problem, the model parameters in ANN model gets highly increased, as image is composed of a lot many pixels. And each pixel of the image becomes the input neuron as well as complexity of the ANN model also gets increased. To get rid of all these problems CNN models are introduced. Where feature extraction is done through using less number of model parameters as compared to ANN model.

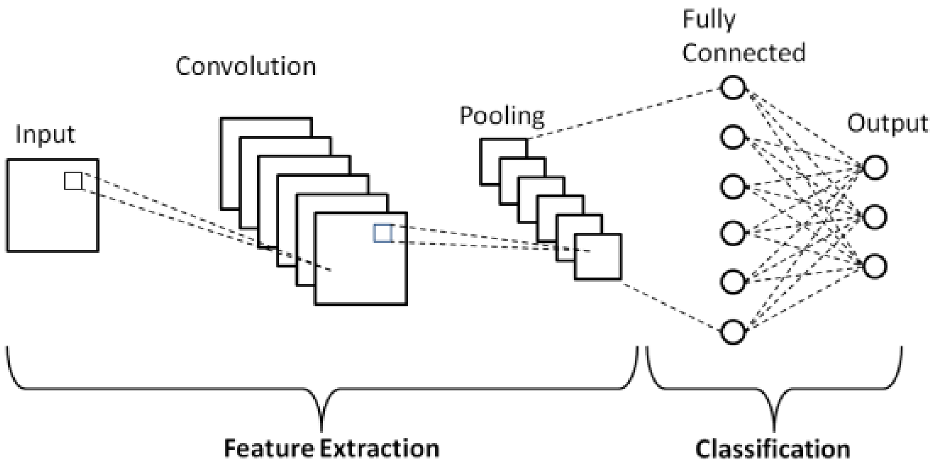


Figure 10. CNN Model

CNN model is composed of Convolutional and Pooling layers. Convolutional layers consists of filters, which are used for feature extraction and it takes parameters like stride, padding, number of filters and size of each filter. An activation function is applied at the end of convolutional layer. And each convolutional layer is followed by a pooling layer , in which a maximum pooling or average pooling or minimum pooling operation is applied. Pooling layer also takes few parametes as inputs like stride, size of each filter. For pooling layer the output image of the convolutional layer is the input.

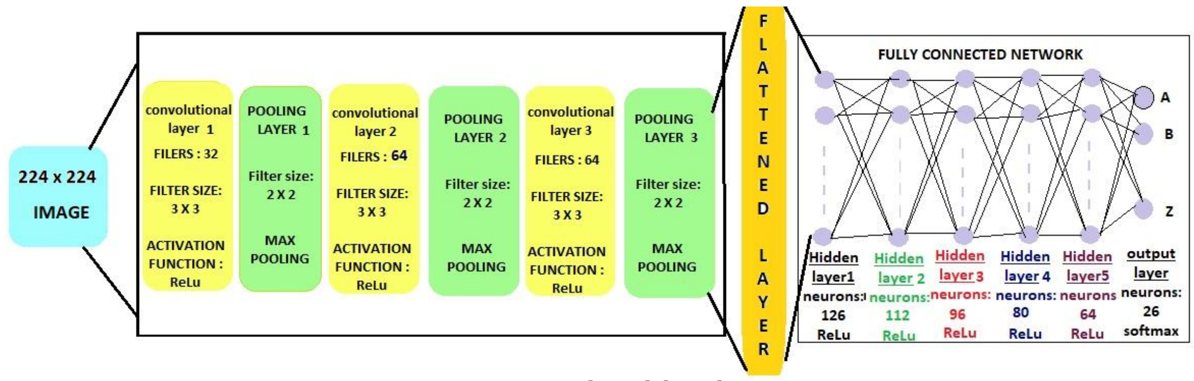
4.2 OVERVIEW TECHNOLOGY

CNN ARCHITECTURE

We developed a CNN model by adding three convolutional layers. The first layer consists of 32 convolutional filters with size 3x3 and with activation function ReLu. This convolutional layer is followed by the pooling layer in which max-pooling operation is used. In pooling layer size of each filter is 2x2. The depth of the input image for the pooling layer determines the number of filters in the pooling layer. The output image of the convolutional layer functions as the pooling layer's input

image. The number of filters employed in the convolutional layer will determine the depth of the output image of the convolutional layer. The number of filters in the pooling layer is indirectly equivalent to the number of filters in the convolutional layer. That is the cause we don't mention the number of filters in the pooling layer.
The architecture of the second and third convolutional layers is the same, with 64 convolutional filters in each. Each filter is 3x3 in size and has the relu activation function. Following each convolutional layer is a pooling layer identical to the first pooling layer.

The last pooling layer's output image is flattened and added to the fully connected network. Five hidden levels and one output layer make up a fully connected network. The first hidden layer contains 128 neurons, the second contains 122 neurons, 96, 80, and the 64 neurons in the remaining hidden layers respectively. The ReLu activation function is employed in all of the hidden layers.


224 x 224 IMAGE
convolutional layer 1 FILERS : 32 FILTER SIZE: 3 X 3 ACTIVATION FUNCTION : ReLu
POOLING LAYER 1 Filter size: 2 X 2 MAX POOLING
convolutional layer 2 FILERS : 64 FILTER SIZE: 3 X 3 ACTIVATION FUNCTION : ReLu
POOLING LAYER 2 Filter size: 2 X 2 MAX POOLING
convolutional layer 3 FILERS : 64 FILTER SIZE: 3 X 3 ACTIVATION FUNCTION : ReLu
POOLING LAYER 3 Filter size: 2 X 2 MAX POOLING
FLATTENED LAYER
FULLY CONNECTED NETWORK
A
B
Z
Hidden layer1 neurons: 126 ReLu
Hidden layer 2 neurons: 112 ReLu
Hidden layer 3 neurons: 96 ReLu
Hidden layer 4 neurons: 80 ReLu
Hidden layer5 neurons 64 ReLu
output layer neurons: 26 softmax
Figure 11. Proposed Model Architecture

26 neurons are present in the output layer because we have 26 alphabets to predict. And we used the Soft max activation function in the output layer. The soft max activation function is widely used for categorical classification problems and we are dealing with a categorical classification problem, we used the softmax activation function.
In the proposed model, we 3 convolutional layers along with pooling layers. In 3 convolutional layers we used activation function as ReLu (Rectified Linear Unit). Convolutional layer produces a output image, and for each pixel of output image ReLu operation is applied.

Max pooling operation is used in pooling layers. This max pooling operation considers the maximum value in the whole filter space as output. The output image size of pooling layer will be

Output image size = [$\frac{n-f}{s} + 1, \frac{n-f}{s} + 1$, d]

In the above equation, n represents the size of input image to the pooling layer, f represents the filter size in pooling layer, s represents the stride value and d represents the depth of the input image to pooling layer.

While training the model we use an optimizer, for reducing the training time we use an optimizer. In our model we used Adam as optimizer, Adam (Adaptive moment estimation) is formed by combining another two optimization algorithms, RMSProp (Root Mean Square Propagation) and Momentum. In Adam optimizer the weights and bias parameters are updated as follows

$$W = \mathrm{W} - \alpha \frac{\mathrm{V_{dw}}}{\sqrt{S_{\mathrm{dw}} + €}} \qquad B = \mathrm{B} - \alpha \frac{\mathrm{V_{dB}}}{\sqrt{S_{\mathrm{dB}} + €}}$$

Where α represents learning rate, W represents weight , B represents bias parameter , $€ = 10^{-8}$ and

$V_{dw} = \beta_1 V_{dw}$(prev) + (1- β_1) dW where β_1= 0.9

$S_{dw} = \beta_2 S_{dw}$(prev) + (1- β_2)$(dW)^2$ where β_2=0.999

$V_{dB} = \beta_1 V_{dB}$ (prev) + (1- β_1) dB where β_1= 0.9

$S_{dB} = \beta_2 S_{dB}$(prev) + (1- β_2)$(dB)^2$ where β_2=0.999

Initially, random weights and bias parameters will be considered, later in each and every iteration the weights and bias parameters will get updated according the mentioned formulas, which helps the model in increasing its accuracy rate. And as the Adam is formed by combination of both the algorithms, Momentum and RMSProp, it has the advantage of both these algorithms. Thus, the Adam optimizer works better than many other optimization techniques.

5. TESTING

Test Cases and Test Results:

Deep learning system requires two types of tests like model evaluation and model testing.

- Model evaluation: covers model metrics that show performance of our model.
- Model Testing: checking the model behavior on predefined scenarios.

1. Model Evaluation Based On Accuracy:

Validation accuracy: the accuracy you calculate on the data set you do not use for training, but you use (during the training process) for validating (or "testing") the generalization ability of the model.

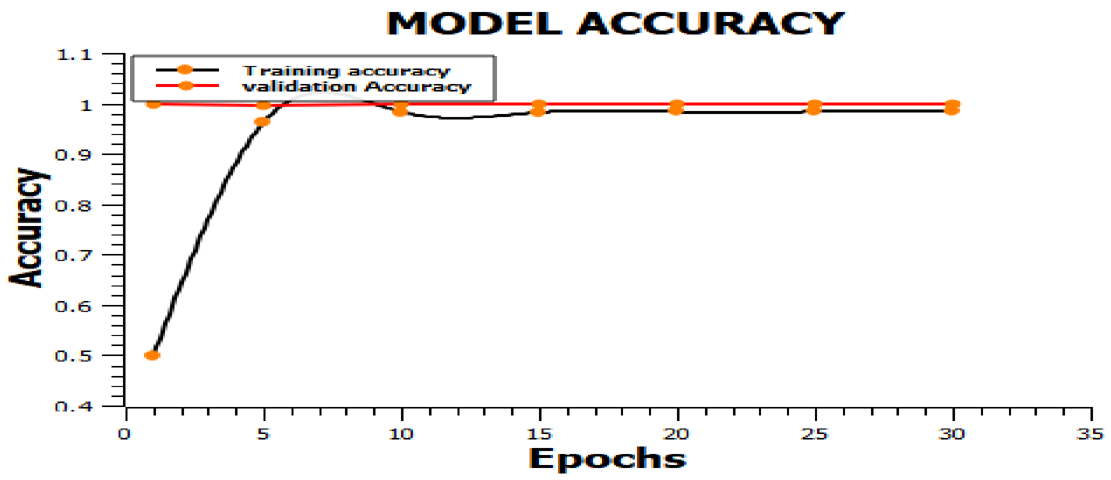


Figure 12.Model accuracy VS Epochs

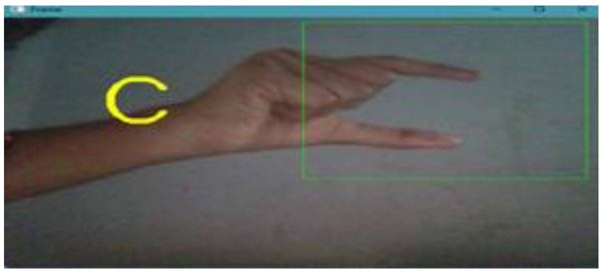
2. Testing the model on different scenarios like lighting conditions, blur images and prediction between two similar letters.

- **Test case 1**: testing the model in low light conditions.

  Test Result 1:

  Model predicted output: C

  Actual output: C

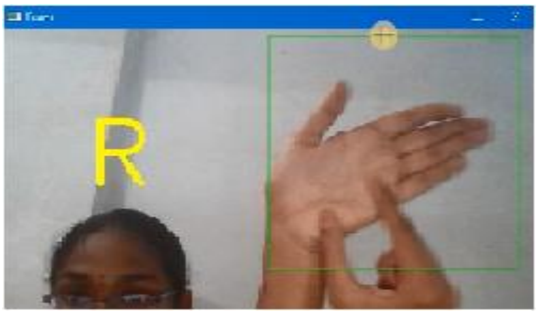


- **Testcase 2**: testing the model in bright light conditions

  Test Result 2:

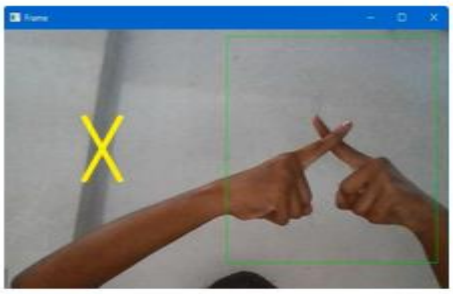
  Model predicted output: R

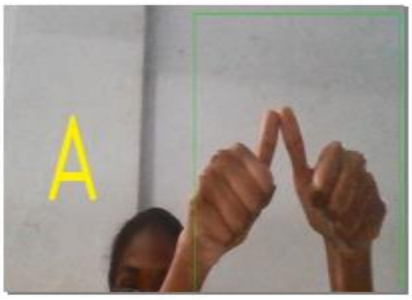
  Actual output: R



- **Test Case 3:** prediction between 2 similar signs A and X.

  Testcase result 3: similar symbols like (V&W, A&X) are misprinted sometimes due to limited training provided to the system.





6. RESULTS

6.1 Dataset

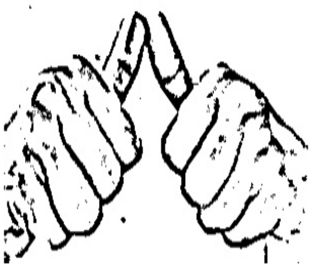
We collected approximately 6500 images with the help of laptop, webcam and open CV library. As there are many datasets available in internet. But they were not reliable. And our dataset is divided into 2 parts in the ratio 70:30. Training dataset contains 5200 images and testing dataset contains 1300 images.

Alphabet	Train Dataset	Test Dataset	Total
A	250	50	300
B	250	50	300
C	250	50	300
D	250	50	300
E	250	50	300
F	250	50	300
G	250	50	300
H	250	50	300
I	250	50	300
J	250	50	300
K	250	50	300
L	250	50	300
M	250	50	300
N	250	50	300
O	250	50	300
P	250	50	300
Q	250	50	300
R	250	50	300
S	250	50	300
T	250	50	300
U	250	50	300
V	250	50	300
W	250	50	300
X	250	50	300
Y	250	50	300
Z	250	50	300

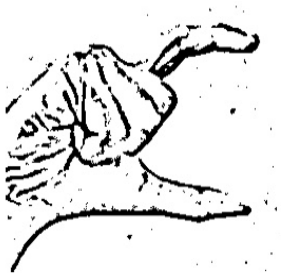
Table.1. Dataset

Table 1: The above table depicts the number of images captured for each alphabet in training and testing datasets. This table gives clear information about the count of images in training and testing datasets which will be very useful in further steps.

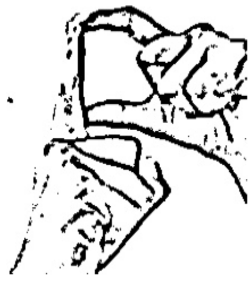
Sample Datasets:



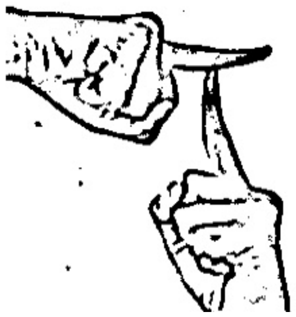
A



C



D



T

Figure.13. Alphabets in sign language

The above figures represent the Indian sign language for the alphabets A, C, D, T. These are the final outputs of the original images, stored in respective folders and given as input to custom CNN model. Actually, the original image is converted into gray scale image where the colored images turn to black and white images which has 2 channels. Then after, we apply Gaussian blur that removes the high frequency components and thresholding is applied on the produced output.

6.2 Procedure

Training and Testing accuracy are the main factors for evaluating the performance of our proposed model. In order to get good accuracy, the model has been trained for different epochs. To train the data, we used custom CNN model. Finally, we observed that our model is performing well at batch size=10, epochs=,30. Our model is predicting the sign language with an accuracy of 98%.

BATCH SIZE 10

MODEL ACCURACY in %

Epochs	Train accuracy	validation accuracy
1	49.7	99.7
5	96.1	99.5
10	98.3	100
15	98.2	100
20	98.3	100
25	98.3	100
30	98.4	100

Table.2. Training and validation accuracies

MODEL LOSS

Epochs	Train Loss	validation Loss
1	1.64	0.02
5	0.12	0.02
10	0.06	0.02
15	0.03	0.02
20	0.03	0.02
25	0.01	0.02
30	0.01	0.02

Table.3. Train and validation loss

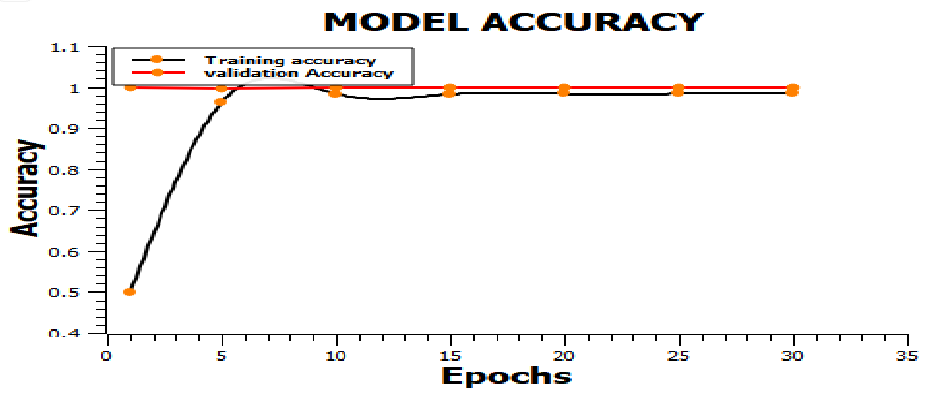


Fig 14: Graph between number of epochs and accuracy

Fig 13 gives information about the model accuracy and epochs. The accuracy value differs from one epoch to another. In the fig 13, as the epoch value increasing, accuracy is also increasing. This is one of the ways to tune our model to get good results.

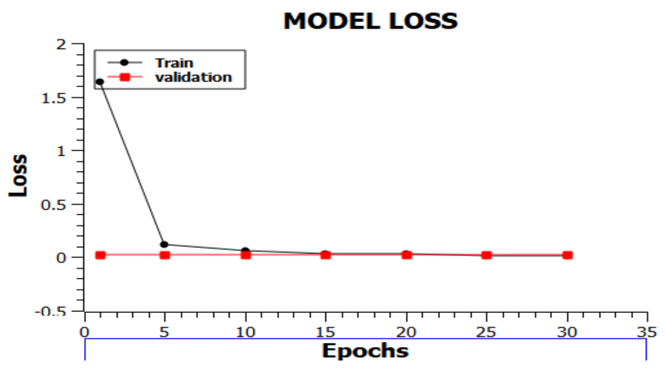


Figure 15: Graph between number of epochs and Loss

Fig 15 gives information about the model loss and epochs. The loss value differs from one epoch to another. In the fig 15, as the epoch value increasing, loss of data is also decreasing. This is one of the ways to tune our model to get good results.

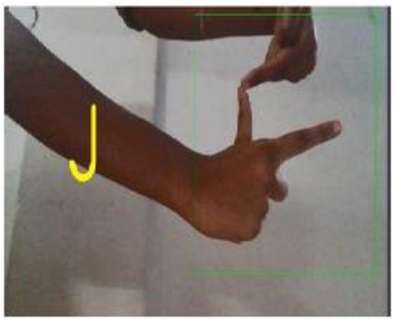


Figure 16.Predicting J

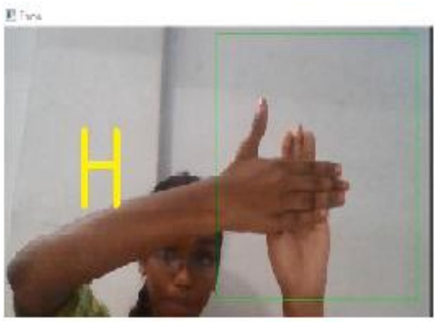


Figure 17.Predicting H

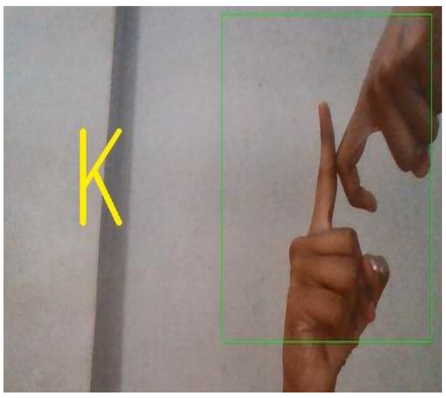


Figure 18.Predicting K

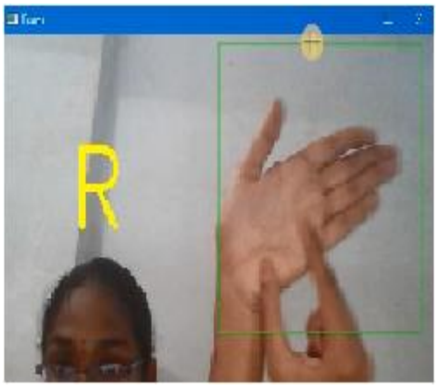


Figure 19.Predicting R

The above figures are the outputs of our proposed model. We can see the alphabet is displaying on the screen for the given input sign. The input sign should be captured in green color box which is of size 224x224.After the completion of image processing the modified image is given as input to CNN model.

7. CONCLUSION

The research presents a technique for classifying and recognizing Indian sign language signs using CNN. Our primary objective is to create a better real-time Sign language recognition system so that the system may be employed everywhere. It is accomplished by creating a custom dataset. We have collected our custom dataset and developed a custom CNN model to predict the images. We can say that Convolutional Neural Network models can be used as classification algorithms to predict the signs. The proposed system was evaluated on a real and challenging sign language dataset. Similar symbols like (V&W, A&X) are misprinted sometimes due to limited training provided to the system.

From the results, it can be inferred that the system presented in this paper is accurately able to track hand signs using some techniques. This model might overcome these limitations if, a more detailed and large dataset in different environment conditions provided for training. The experimental results showed that the proposed system performed well in terms of recognition rate, demonstrating its effectiveness. Our model detects the alphabet, which is signed in the Indian signed language. We found the accuracy of our custom CNN model is about 98%.

8. FUTURE SCOPE

Further, extensions to our project would be converting these predicted alphabets to words and not only detecting static hand gestures, but we can also extend this model to detect dynamic hand gestures. Also, with the use of Natural Language Processing algorithms, this system can be extended
to recognize sentences in ISL, by recognition of multiple gestures in the same video capture. In the future, we can also develop a two-way communication system by converting words to signs which will be very helpful for physically impaired people. A user-friendly web- based application can also be created in future that takes image as input and gives alphabet as output. We can also develop this model by adding speech channel too. After predicting the output, the text will be converted to speech.

9. BIBLIOGRAPHY

[1] K. Bantupalli and Y. Xie, "American Sign Language Recognition using Deep Learning and Computer Vision," 2018 IEEE International Conference on Big Data (BigData),Seattle,WA,USA,2018,pp. 4896-4899,doi:10.1109/BigData.2018.8622141.

[2] M. Geetha and U. C. Manjusha, , "A Vision Based Recognition of Indian Sign Language Alphabets and Numerals Using B-Spline Approximation", International Journal on Computer Science and Engineering (IJCSE), vol. 4, no. 3, pp. 406-415. 2012.

[3]Pigou L., Dieleman S., Kindermans PJ., Schrauwen B. (2015) Sign Language Recognition Using Convolutional Neural Networks. In: Agapito L., Bronstein M., Rother C. (eds) Computer Vision - ECCV 2014 Workshops. ECCV 2014. Lecture Notes in Computer Science, vol 8925. Springer, Cham. https://doi.org/10.1007/978-3-319-16178-5_40

[4] Jaoa Carriera, A. Z. (2018). Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on (pp. 4724-4733). IEEE. Honolulu.

[5] V. N. T. Truong, C. Yang and Q. Tran, "A translator for American sign language to text and speech," 2016 IEEE 5th Global Conference on Consumer Electronics, 2016, pp. 1-2, doi: 10.1109/GCCE.2016.7800427.

[6] H, Muthu & V, Dr. (2021). Indian Sign Language Recognition through Hybrid ConvNet-LSTM Networks. EMITTER International Journal of Engineering Technology. 9. 182-203. 10.24003/emitter.v9i1.613.

[7] Kishore P.V., Prasad M.V., Prasad C.R., Rahul R. 4-Camera model for sign language recognition using elliptical fourier descriptors and ANN. 2015 International Conference on Signal Processing and Communication Engineering Systems, Guntur, India, 2015, pp. 34–38.

[8] A depth-based Indian Sign Language recognition using Microsoft Kinect / T. Raghuveera, R. Deepthi, R. Mangalashri, R. Akshaya // Sādhanā. – 2020. – Vol. 45, N 1. – P. 34.

[9] Marwa Elpeltagy, Moataz Abdelwahab, Mohamed E. Hussein, Amin Shoukry, Asmaa Shoala, Moustafa Galal, Multi-modality-based Arabic Sign Language recognition, IET Computer Vision, vol. 12, no. 7, Oct 2018, 1031- 1039. https://doi.org/10.1049/iet-cvi.2017.0598.

[10] Borghetti, Michela & Sardini, Emilio & Serpelloni, Mauro. (2013). Sensorized Glove for Measuring Hand Finger Flexion for Rehabilitation Purposes. Instrumentation and Measurement, IEEE Transactions on. 62. 3308-3314. 10.1109/TIM.2013.2272848.

[11] CABRERA, MARIA & BOGADO, JUAN & Fermín, Leonardo & Acuña, Raul & RALEV, DIMITAR. (2012). GLOVE-BASED GESTURE RECOGNITION SYSTEM. 10.1142/9789814415958_0095.

[12] He, Siming. (2019). Research of a Sign Language Translation System Based on Deep Learning. 392-396. 10.1109/AIAM48774.2019.00083.

[13] International Conference on Trendz in Information Sciences and Computing (TISC). : 30-35, 2012.

[14] Herath, H.C.M. & W.A.L.V.Kumari, & Senevirathne, W.A.P.B & Dissanayake, Maheshi. (2013). IMAGE BASED SIGN LANGUAGE RECOGNITION SYSTEM FOR SINHALA SIGN LANGUAGE

[15] Escalera, S., Baró, X., Gonzàlez, J., Bautista, M., Madadi, M., Reyes, M., . . . Guyon, I. (2014). ChaLearn Looking at People Challenge 2014: Dataset and Results. Workshop at the European Conference on Computer Vision (pp. 459-473). Springer, . Cham.

[16] Huang, J., Zhou, W., & Li, H. (2015). Sign Language Recognition using 3D convolutional neural networks. IEEE International Conference on Multimedia and Expo (ICME) (pp. 1-6). Turin: IEEE.

[17] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on (pp. 248-255). IEEE. Miami, FL, USA .

[18] Soomro, K., Zamir , A. R., & Shah, M. (2012). UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. Computer Vision and Pattern Recognition, arXiv:1212.0402v1, 1-7.

[19] Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., & Serre, T. (2011). HMDB: a large video database for human motion recognition. Computer Vision (ICCV), 2011 IEEE International Conference on (pp. 2556-2563). IEEE

[20] Zhao, Ming & Bu, Jiajun & Chen, C.. (2002). Robust background subtraction in HSV color space. Proceedings of SPIE MSAV, vol. 1. 4861. 10.1117/12.456333.

[21] Chowdhury, A., Sang-jin Cho, & Ui-Pil Chong. (2011). A background subtraction method using color information in the frame averaging process. Proceedings of 2011 6th International Forum on Strategic Technology. doi:10.1109/ifost.2011.6021252 International Journal of Engineering Research & Technology (IJERT) http://www.ijert.org ISSN: 2278-0181 IJERTV9IS120029 (This work is licensed under a Creative Commons Attribution 4.0 International

[22] TalkingHands.co.in, “Talking Hands,” 2014. [Online]. Available: http://www.talkinghands.co.in/. [Accessed: 21- Jul- 2017].

[23] A. Agarwal and M. K. Thakur, “Sign Language Recognition using Microsoft Kinect,” Sixth International Conference on Contemporary Computing (IC3), September 2013.

[24] MailOnline, ‘'SignAloud gloves translate sign language gestures into spoken English,” 2016. [Online]. Available: http://www.dailymail.co.uk/sciencetech/article-3557362/SignAloudgloves-translate-sign-language-movements-spoken-English.html. . [Accessed: 10- Feb- 2018].

[25] Alexia. Tsotsis, “MotionSavvy Is A Tablet App That Understands Sign Language,” 2014. [Online]. Available: https://techcrunch.com/2014/06/06/motionsavvy-is-a-tablet-app-thatunderstands-sign-language/. [Accessed: 10 – Feb- 2018].

[26] P. Paudyal, A. Banerjee and S. K. S. Gupta, “SCEPTRE: a Pervasive, Non-Invasive, and ProgrammableGesture Recognition Technology,” Proceedings of the 21st International Conference on Intelligent User Interfaces, pp. 282-293, 2016.

[27] R. Y. Wang and J. Popovic, “Real-Time Hand-Tracking with a Color Glove,” ACM transactions on graphics (TOG), vol. 28, no. 3, 2009.

[28] R. Akmeliawati , M. P. L. Ooi and Y. C. Kuang, “Real-Time Malaysian Sign Language Translation using Colour Segmentation and Neural Network,” Instrumentation and Measurement Technology Conference Proceedings, 2007.

[29] F. S. Chen, C. M. Fu and C. L. Huang, "Hand gesture recognition using a real-time tracking method and hidden Markov models", Image and vision computing, vol. 21, pp. 745-758, 2003.

[30] M. A. Rahaman , M. Jasim, M. H. Ali and M. Hasanuzzaman, “RealTime Computer Vision-Based Bengali Sign Language Recognition,” 17th International Conference on Computer and Information Technology (ICCIT), 2014.

[31] S. Padmavathi, M. S. Saipreethy and V. Valliammai, “Indian Sign Language Character Recognition using Neural Networks,” IJCA Special Issue on Recent Trends in Pattern Recognition and Image Analysis RTPRIA, 2013.

[32] A. Chaudhary, J. L. Raheja and S. Raheja, “A Vision based Geometrical Method to find Fingers Positions in Real Time Hand Gesture Recognition,” JSW, pp. 861-869, 2012.

[33] A. B. Jmaa, W. Mahdi, Y.B. Jemaa and A.B. Hamadou, “Hand Localization and Fingers Features Extraction: Application to Digit Recognition in Sign Language,” International Conference on Intelligent Data Engineering and Automated Learning, pp. 151-159, 2009.

[34] G. Awad, J. Han and A. Sutherland, “A Unified System for Segmentation and Tracking of Face and Hands in Sign Language Recognition,” 18th International Conference on Pattern Recognition, 2006.

[35] Z. H. Al-Tairi, R. W. Rahmat, M.I. Saripan and P.S. Sulaiman, “Skin Segmentation Using YUV and RGB Color Spaces,” J Inf Process Syst, vol. 10, no. 2, pp. 283-299, June 2014.

[36] H. Lahiani , M. Elleuch and M. Kherallah, “Real Time Hand Gesture Recognition System for Android Devices,” 15th International Conference on Intelligent Systems Design and Applications (ISDA), 2015.

[37] C. W. Ng and S. Ranganath, “Real-time gesture recognition system and application,” Image and Vision computing, 2002.

[38] S. C. Agrawal, A. S. Jalal and C. Bhatnagar, “Recognition of Indian Sign Language using Feature Fusion,” 4th International Conference on Intelligent Human Computer Interaction (IHCI), 2012.

[39] M. Hrúz, J. Trojanová and M. Železný, “Local Binary Pattern based features for Sign Language Recognition,” Pattern Recognition and Image Analysis, 2012.

[40] J. Davis and M. Shah, “Visual gesture recognition,” IEE ProceedingsVision, Image and Signal Processing, 1994.

[41] C. Y. Kao and C. S. Fahn, “A Human-Machine Interaction Technique: Hand Gesture Recognition based on Hidden Markov Models with Trajectory of Hand Motion,” Procedia Engineering, vol. 15, pp. 3739- 3743, 2011.

[42] N. Dalal and B. Triggs, “Histogram of Oriented Gradients for Human Detection,” Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. 886-893, 2005. [doi = 10.1109/CVPR.2005.177]

[43] B. C. Ennehar, O. Brahim, and T. Hicham, “An appropriate color space to improve human skin detection,” INFOCOMP Journal of Computer Science, vol. 9, no. 4, pp. 1-10, 2010.

[44] L. Maaten and G. Hinton, “Visualizing Data using t-SNE,” Journal of Machine Learning Research, vol. 9, pp. 2579-2605, November 2008.

[42] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel et al, “1.6. Nearest Neighbours – scikit-learn 0.19.1 documentation,” 2011. [Online]. Available: http://scikitlearn.org/stable/modules/neighbors.html#nearest-neighboralgorithms. [Accessed: 12- Sep- 2017].

[43] C. Vogler, D. Metaxas, “Handshapes and movements: Multiple channel ASL Recognition,” Gesture-Based Communication in HumanComputer Interaction, pp. 247-258, 2004.

[44] L. R. Rabiner, “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition,” Proceedings of the IEEE, vol. 77, no. 2, February 1989.

[45] L. Baum, “An Inequality and Associated Maximization Technique in Statistical Estimation for Probabilistic Functions of Markov Process,” Inequalities III: Proceedings of the Third Symposium on Inequalities, ssvol. 3, pp. 1-8, 1972.

Sign Language Recognition Using Custom CNN model

Dr P Praveen[1], B Anu[2], P Shiva Sowmya[3],G Anjali Kiran[4], G Sai Srujan[5]

[1.]Associate Professor in School of Computer Science and Artificial Intelligence,SR University,Warangal,Telangana State.

[2,3,4,5] School of Computer Science and Artificial Intelligence,SR University,Warangal,Telangana State.

prawin1731@gmail.com,anubaluguri22@gmail.com,anjalipatel.gundam@gmal .com, saisrujangundeti@gmail.com, sowmyapanthangi555@gmail.com

Abstract. One of the vital challenges that deaf people face in the present society is communication. The use of sign language translators is not a viable solution. Recently Deep learning models showcased outstanding results in classification and recognition systems. So, researchers are focusing more on developing deep learning models with at most accuracy. The study aims to represent a Convolutional neural network-based Indian Sign Language identification model for deaf persons and further convert it into the text, which is more accurate than prior studies. In this study, a CNN-based, ISL hand-gesture recognition model is developed, which outperforms many other existing models. A custom dataset was prepared, to train the model. The developed model can recognize 26 alphabets represented through hand gestures in ISL. We found that the model produced an accuracy of 98%.

Keywords: Deep Learning, Convolutional Neural Network, Indian Sign Language Recognition, Open CV, Gaussian Blur, Grayscale, Normalization.

1 Introduction

According to Census, India has over one million deaf people and over ten million people with hearing loss. To aid in the teaching and interpretation of sign language, a variety of programs have been developed. However, progress in recognizing sign languages with modern technology has been promising but very limited. Software that can recognize and interpret sign language is in high demand. It can also act as the bridge between sign language users and non-sign language users. As a result, it is critical to establish the link between the deaf and the spoken by creating a deep learning model to translate sign language to text. Humans can communicate with others in different ways for example, speaking in some languages like Telugu, Hindi and many more. Sign language Recognition will reduce the communication gap between normal people and deaf or muted people.

Sign language is the combination of static and dynamic gestures. Static hand gestures don't require any hand movement. The 26 alphabets, ten digits, and a few static word signs make up sign language. Each country employs its own as there is no universally acknowledged sing language. Sign languages include American Sign Language, British Sign Language, Indian Sign Language, and many more. Even though Indian sign language is used by the majority of deaf people in India, it is not widely employed in schools. Many organizations that work with deaf people have made efforts to promote Indian Sign Language, but most people are unaware of these signals due to their complexity, making communication between the deaf and the spoken difficult. Throughout the previous few decades, studies have predominantly focused on the identification of American Sign Language. American Sign Language represents the alphabet with a single hand, whereas Indian Sign Language (ISL) represents the alphabet with both hands.

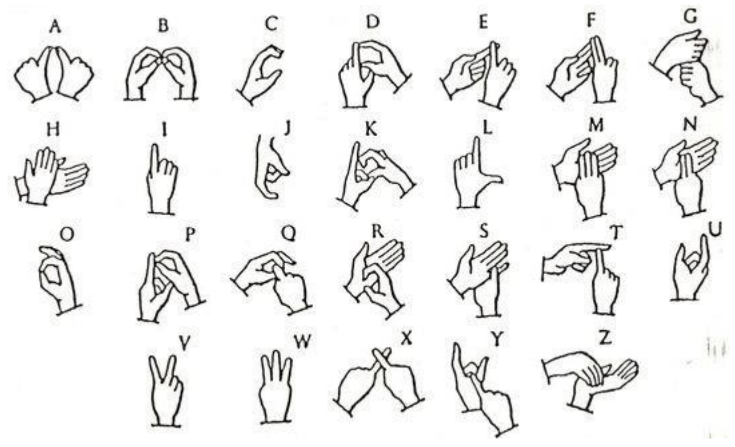


Fig.1.Alphabets in Indian Sign Language

2 Related Work

Many researchers have conducted studies on Sign Language Recognition throughout the last few decades. Based on previous research, we found that there are two existing methodologies. They are glove-based systems and image processing and recognition. Many early Sign Language Recognition systems used data-gloves and accelerometers to acquire details of the hands. The measurements were measured directly using Data Glove. In a glove-based system, a person has to wear a glove that consists of sensors like a flex sensor, which tracks the motion of the hand [1]. It compares with the data received directly from sensors based on our finger movements. It also contains an accelerometer and five flex sensors[5,9]. The accelerometer detects hand movements, which generates the initial bit of binary number to be compared in a lookup table, which determines the language. For recognition, the data glove provided analog signal data to the microcontroller. Finally, a pre-recorded voice matched with a recognized indication was used to demonstrate the outcome.

The authors, Borghetti, Sardini, and Serpelloni, created a numerical data-based sign language recognition system. It consists of a 3-axis accelerometer and Hall sensors. The data glove made use of four Hall sensors that were attached to the fingers. An accelerometer is used to detect finger bending, while Hall sensors are used to monitor hand orientation. The analog sensor data sent into MATLAB code is used to identify indications given by the gloves. Only digits ranging from 0 to 9 were studied in this experimental setup. In digit recognition, the created system achieved 96 percent accuracy [10].

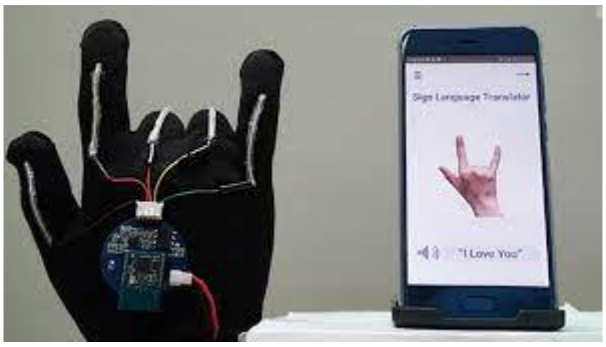


Fig.2. Example of Data Glove

The authors, Muthu and Gomathi, proposed hybrid CNN-RNN architecture to create a real-time Indian sign language (ISL) detection system. The ISL dataset was used to train the system[6]. On the test data, the proposed model achieved 95.99%. Kishore, Prasad M, V, Prasad C.R, and Rahul suggested a 4-camera model for segmenting and classifying hand motions using features extracted from elliptical Fourier descriptors. Their system had a recognition rate of around 92.23 percent [7]. Raghuveera, Deepthi, Mangalashri, Akshaya proposed a model to recognize ISL singlehanded signs, double-handed signs, and fingerspelling signs from 4,600 photos, with a 71.85% accuracy [8]. Based on 3D skeletal point data from the Kinect sensor and SVM, few researchers recognized 37 Indian Sign Language signs. Furthermore, each of these sensors has advantages and disadvantages in terms of moving data. All of these methods, however, relied on sensors to detect the indications.

The image processing and recognition approach captures photos with a camera, compares them to existing images, and detects the sign A still hand picture frame is captured using a webcam [2,4]. The processing of these frames improves them. Using feature extraction and classification techniques, the sign language is then translated into English text. This translation is converted to speech using the text-to-speech API. Before doing feature extraction, the images need to be processed such that only valuable information is analyzed while redundant, distracting noise, and superficial data are ignored. For faster computation, the images are first reduced to 100 x 100 pixels. The picture is then transformed to grayscale before being translated to binary.

The YCbCr model is used to detect skin color simultaneously. Finally, an edge detector, called a Canny edge detector, is used to detect edges.

Fig.3.Example of Kinect color and depth video stream

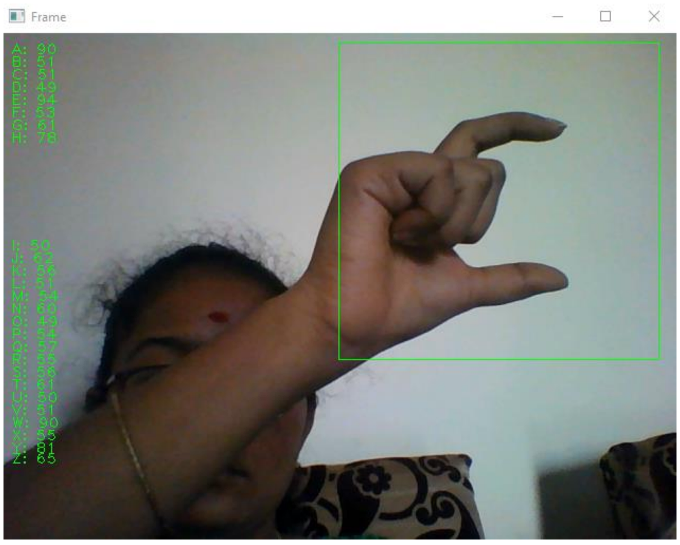
Recently Microsoft Kinect has offered a reasonable depth camera. Because of its capabilities, Kinect is now widely employed by scientists. Kinect is capable of delivering both color and depth video streams at the same time, and background segmentation is simple to accomplish with depth data [3,8]. Kinect is used to recognize signs. Recently there are no data sets accessible as the outcomes are limited.

2 Methodology

2.1 Data Collection

For Indian Sign Language recognition, to obtain good accuracy we have collected our own dataset through webcam using open CV. To collect the dataset, we have created two folders, train and test folders. The images present in train and test folder are used for training and testing the built model.

The train and test folders contains the sub folders, named as A, B, C…. Z. These are the classes we have to predict. Any image which is captured belongs to one of these classes. Folder A consists of images in which, alphabet A is represented in Indian Sign Language. And all other folders also consist of their respective images.



```
A: 90
B: 51
C: 51
D: 49
E: 94
F: 53
G: 61
H: 78

I: 50
J: 62
K: 56
L: 51
M: 54
N: 60
O: 49
P: 54
Q: 57
R: 55
S: 56
T: 61
U: 50
V: 51
W: 90
X: 55
Y: 81
Z: 65
```

Fig.4. Dataset collection

Fig.4 is the output display screen of data collection. The green square shown in the figure is region of interest, the sign which we represent with our hands should be placed in that region itself. Only the image covered in that region will be saved. While capturing the image, to save the image into its corresponding folder we have to press the alphabet to which it belongs, through keyboard.

In the left side of figure 3 we have labels like A: 90, B: 51 and so on, these are the count of images present in each folder. According to figure 3, we have 90 images in folder A, 51 images in folder B, etc. In the same way data is collected for both testing and training.

2.2 Data Preprocessing

Before storing the image into its respective folder, the image is resized to 224 x 224 pixels and a RGB image (colored image) is transformed to grayscale image.

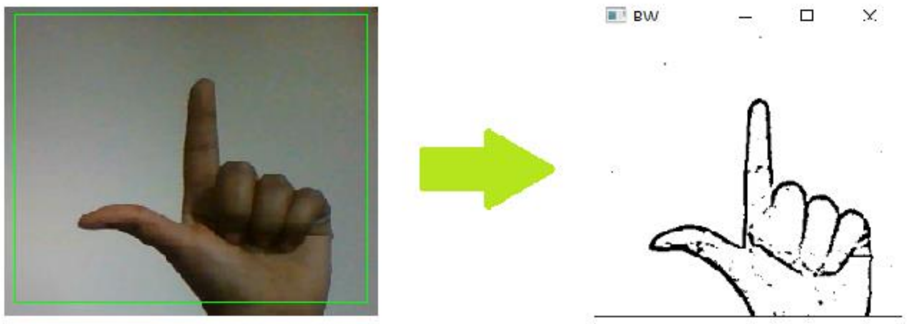


Fig.**5**. Image Preprocessing

As shown in the figure 5, after capturing the image through webcam, the RGB (colored image) is transformed into a grayscale image. In the grayscale image, the

outline of the hand gesture is produced using the Gaussian blur approach and by applying threshold for the image.
Normalization, a preprocessing technique is applied to all the images before training or testing the model. Using this technique all pixel values in an image are ranged between 0 to 1. This is done through rescaling the image. To rescale the image, each pixel value is divided by 225, in an image the maximum pixel value is 225.

2.3 Constructing the models

We developed a CNN model by adding three convolutional layers. The first layer consists of 32 convolutional filters with size 3x3 and with activation function ReLu. This convolutional layer is followed by the pooling layer in which max-pooling operation is used. In pooling layer size of each filter is 2x2. The depth of the input image for the pooling layer determines the number of filters in the pooling layer. The output image of the convolutional layer functions as the pooling layer's input image. The number of filters employed in the convolutional layer will determine the depth of the output image of the convolutional layer. The number of filters in the pooling layer is indirectly equivalent to the number of filters in the convolutional layer. That is the cause we don't mention the number of filters in the pooling layer.

The architecture of the second and third convolutional layers is the same, with 64 convolutional filters in each. Each filter is 3x3 in size and has the relu activation function. Following each convolutional layer is a pooling layer identical to the first pooling layer.

The last pooling layer's output image is flattened and added to the fully connected network. Five hidden levels and one output layer make up a fully connected network. The first hidden layer contains 128 neurons, the second contains 122 neurons, 96,80, and the 64 neurons in the remaining hidden layers respectively. The ReLu activation function is employed in all of the hidden layers.

26 neurons are present in the output layer because we have 26 alphabets to predict. And we used the Softmax activation function in the output layer. The softmax activation function is widely used for categorical classification problems and we are dealing with a categorical classification problem, we used the softmax activation function.

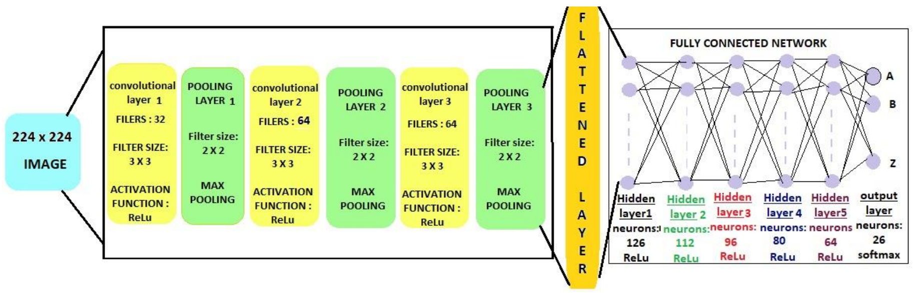


Fig.6. Proposed model Architecture

3 Implementation

In the proposed model, we 3 convolutional layers along with pooling layers. In 3 convolutional layers we used activation function as ReLu (Rectified Linear Unit). Convolutional layer produces a output image, and for each pixel of output image ReLu operation is applied.

$$\text{Output} = \max(\, 0, \text{input}\,)$$

Max pooling operation is used in pooling layers. This max pooling operation considers the maximum value in the whole filter space as output. The output image size of pooling layer will be

$$\text{Output image size} = [\ \frac{n-f}{s} + 1, \frac{n-f}{s} + 1, d]$$

In the above equation, n represents the size of input image to the pooling layer, f represents the filter size in pooling layer, s represents the stride value and d represents the depth of the input image to pooling layer.

While training the model we use an optimizer, for reducing the training time we use an optimizer. In our model we used Adam as optimizer, Adam (Adaptive moment estimation) is formed by combining another two optimization algorithms, RMSProp (Root Mean Square Propagation) and Momentum. In Adam optimizer the weights and bias parameters are updated as follows

$$W = W - \alpha \frac{V_{dw}}{\sqrt{S_{dw} + \epsilon}} \qquad B = B - \alpha \frac{V_{dB}}{\sqrt{S_{dB} + \epsilon}}$$

Where α represents learning rate, W represents weight , B represents bias parameter , $\epsilon = 10^{-8}$ and

$V_{dw} = \beta_1 V_{dw}$(prev) + (1- β_1) dW where $\beta_1 = 0.9$

$S_{dw} = \beta_2 S_{dw}$(prev) + (1- $\beta_2)(dW)^2$ where $\beta_2 = 0.999$

$V_{dB} = \beta_1 V_{dB}$ (prev) + (1- β_1) dB where $\beta_1 = 0.9$

$S_{dB} = \beta_2 S_{dB}$(prev) + (1- $\beta_2)(dB)^2$ where $\beta_2 = 0.999$

Initially, random weights and bias parameters will be considered, later on in each and every iteration the weights and bias parameters will get updated according the mentioned formulas, which helps the model in increasing its accuracy rate. And as the Adam is formed by combination of both the algorithms Momentum and RMSProp, it has the advantage of both these algorithms. Thus the Adam optimizer works better than many other optimization techniques.

4 Results

4.1 Dataset

We collected approximately 6500 images with the help of laptop, webcam and opencv library. As there are many datasets available in internet. But they were not reliable. And our dataset is divided into 2 parts in the ratio 70:30. Training dataset contains 5200 images and testing dataset contains 1300 images.

Alphabet	Train Dataset	Test Dataset	Total
A	250	50	300
B	250	50	300
C	250	50	300
D	250	50	300
E	250	50	300
F	250	50	300
G	250	50	300
H	250	50	300
I	250	50	300
J	250	50	300
K	250	50	300
L	250	50	300
M	250	50	300
N	250	50	300
O	250	50	300
P	250	50	300
Q	250	50	300
R	250	50	300
S	250	50	300
T	250	50	300
U	250	50	300
V	250	50	300
W	250	50	300
X	250	50	300
Y	250	50	300
Z	250	50	300

Table 1 : The above table depicts the number of images captured for each alphabet in training and testing datasets. This table gives clear information about the count of images in training and testing datasets which will be very useful in further steps.

Sample Datasets:

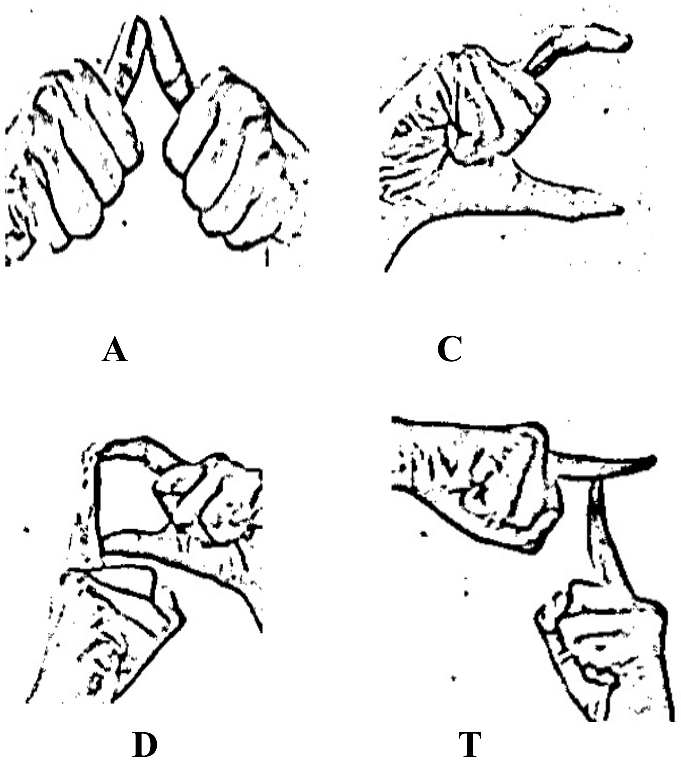


Fig 7: alphabets in sign language

The above figure represents the Indian sign language for the alphabets A,C,D,T. These are the final outputs of the original images , stored in respective folders and given as input to custom CNN model. Actually, the original image is converted into gray scale image where the coloured images turns to black and white images which has 2 channels. Then after, we apply Gaussian blur that removes the high frequency components and thresholding is applied on the produced output.

4.2 Procedure

Training and Testing accuracy are the main factors for evaluating the performance of our proposed model. In order to get good accuracy, the model has been trained for different epochs. To train the data, we used custom CNN model. Finally, we observed that our model is performing well at batch size=10, epochs=,30. Our model is predicting the sign language with an accuracy of 98%.

BATCH SIZE 10

MODEL ACCURACY in %

Epochs	Train accuracy	validation accuracy
1	49.7	99.7
5	96.1	99.5
10	98.3	100
15	98.2	100
20	98.3	100
25	98.3	100
30	98.4	100

Table2: training and validation accuracies at different epochs.

MODEL LOSS

Epochs	Train Loss	Validation Loss
1	1.64	0.02
5	0.12	0.02
10	0.06	0.02
15	0.03	0.02
20	0.03	0.02
25	0.01	0.02
30	0.01	0.02

Table 3: train and validation loss at at different epochs.

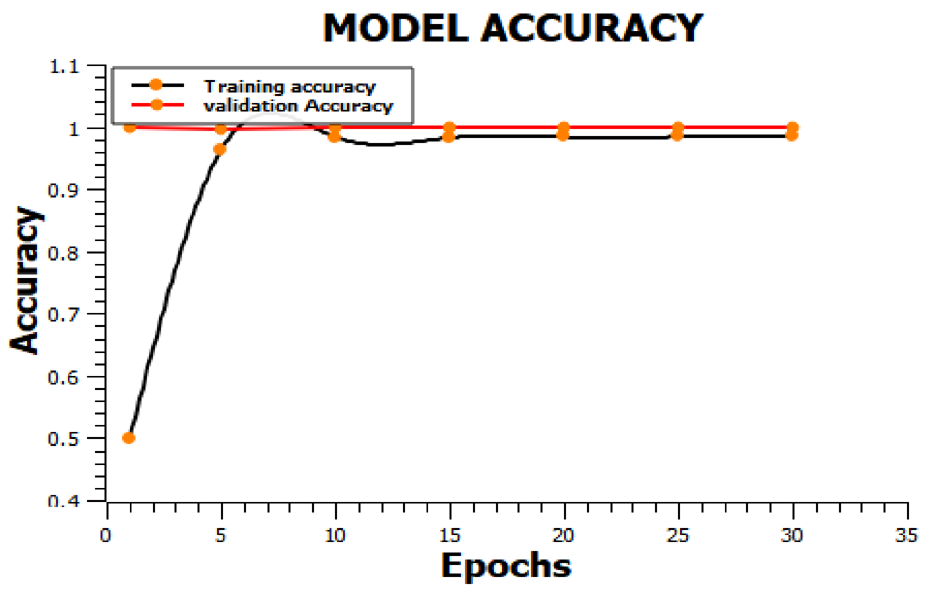


Fig 8: Graph between number of epochs and accuracy

Fig 2 gives information about the model accuracy and epochs. The accuracy value differs from one epoch to another. In the fig 2, as the epoch value increasing , accuracy is also increasing. This is one of the way to tune our model to get good results

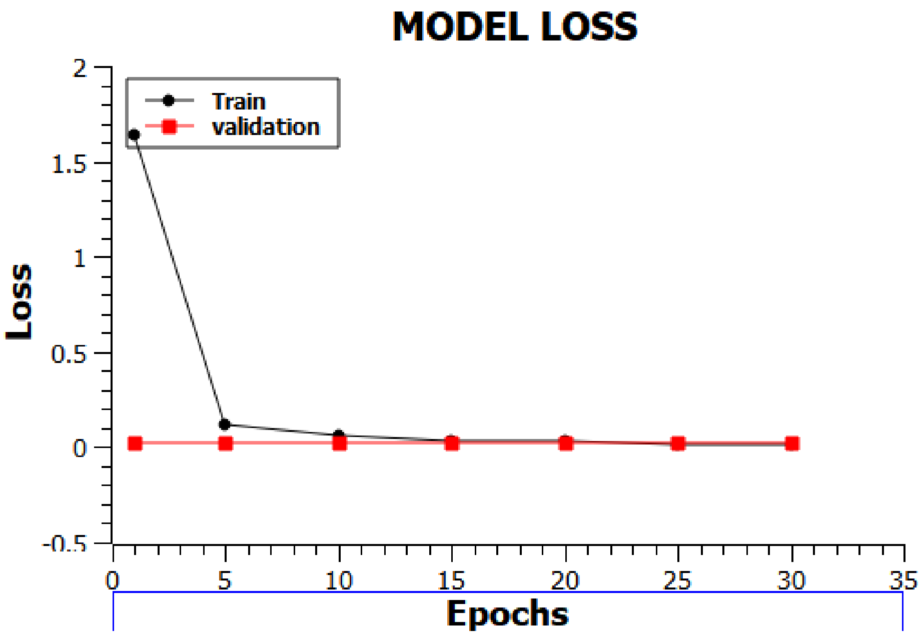
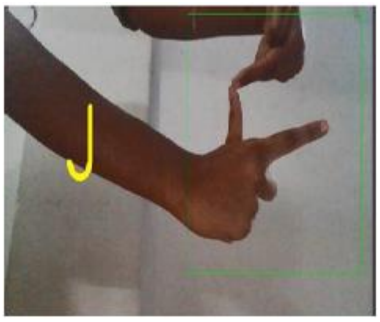


Fig 9:Graph between number of epochs and Loss

Fig 3 gives information about the model loss and epochs. The loss value differs from one epoch to another. In the fig 3, as the epoch value increasing, loss of data is also decreasing. This is one of the ways to tune our model to get good results.



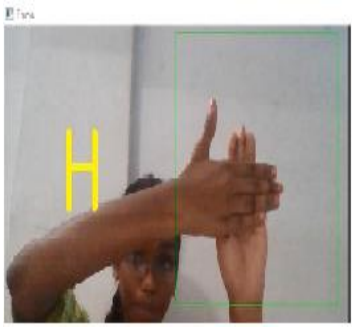


Fig.10.Predicting J

Fig.11.Predicting H

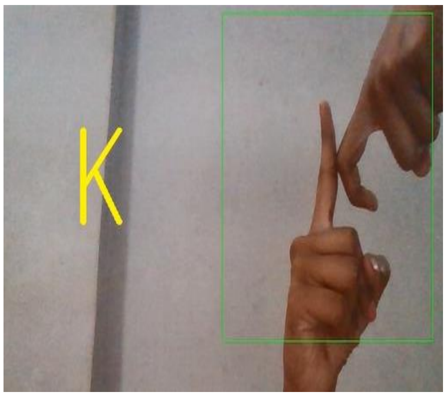


Fig.12.Predicting K

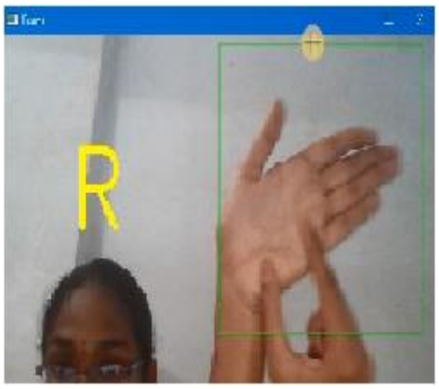


Fig.13.Predicting R

The above figures are the outputs of our proposed model. We can see the alphabet is displaying on the screen for the given input sign. The input sign should be captured in green color box which is of size 224x224.After the completion of image processing the modified image is given as input to CNN model.

Conclusion & Future Scope

The research presents a technique for classifying and recognizing Indian sign language signs using CNN. Our primary objective is to create a better real-time Sign language recognition system so that the system may be employed everywhere. It is accomplished by creating a custom dataset. We have collected our custom dataset and developed a custom CNN model to predict the images. Our model detects the alphabet, which is signed in the Indian signed language. We found the accuracy of our custom CNN model is about 98%. Further, extensions to our project would be converting these predicted alphabets to words and not only detecting static hand gestures, but we can also extend this model to detect dynamic hand gestures. In the future, we can also develop a two-way communication system by converting words to signs which will be very helpful for physically impaired people. A user-friendly web-based application can also be created in future that takes image as input and gives alphabet as output. We can also develop this model by adding speech channel too. After predicting the output, the text will be converted to speech.

References

[1] K. Bantupalli and Y. Xie, "American Sign Language Recognition using Deep Learning and Computer Vision," 2018 IEEE International Conference on Big Data (BigData),Seattle,WA,USA,2018,pp. 4896-4899,doi:10.1109/BigData.2018.8622141.

[2] M. Geetha and U. C. Manjusha, , “A Vision Based Recognition of Indian Sign Language Alphabets and Numerals Using B-Spline Approximation”, International Journal on Computer Science and Engineering (IJCSE), vol. 4, no. 3, pp. 406-415. 2012.

[3]Pigou L., Dieleman S., Kindermans PJ., Schrauwen B. (2015) Sign Language Recognition Using Convolutional Neural Networks. In: Agapito L., Bronstein M., Rother C. (eds) Computer Vision - ECCV 2014 Workshops. ECCV 2014. Lecture Notes in Computer Science, vol 8925. Springer, Cham. https://doi.org/10.1007/978-3-319-16178-5_40

[4] Jaoa Carriera, A. Z. (2018). Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on (pp. 4724-4733). IEEE. Honolulu.

[5] V. N. T. Truong, C. Yang and Q. Tran, "A translator for American sign language to text and speech," 2016 IEEE 5th Global Conference on Consumer Electronics, 2016, pp. 1-2, doi: 10.1109/GCCE.2016.7800427.

[6] H, Muthu & V, Dr. (2021). Indian Sign Language Recognition through Hybrid ConvNet-LSTM Networks. EMITTER International Journal of Engineering Technology. 9. 182-203. 10.24003/emitter.v9i1.613.

[7] Kishore P.V., Prasad M.V., Prasad C.R., Rahul R. 4-Camera model for sign language recognition using elliptical fourier descriptors and ANN. 2015 International Conference on Signal Processing and Communication Engineering Systems, Guntur, India, 2015, pp. 34–38.

[8] A depth-based Indian Sign Language recognition using Microsoft Kinect / T. Raghuveera, R. Deepthi, R. Mangalashri, R. Akshaya // Sādhanā. – 2020. – Vol. 45, N 1. – P. 34.

[9] Marwa Elpeltagy, Moataz Abdelwahab, Mohamed E. Hussein, Amin Shoukry, Asmaa Shoala, Moustafa Galal, Multi-modality-based Arabic Sign Language recognition, IET Computer Vision, vol. 12, no. 7, Oct 2018, 1031- 1039. https://doi.org/10.1049/iet-cvi.2017.0598.

[10] Borghetti, Michela & Sardini, Emilio & Serpelloni, Mauro. (2013). Sensorized Glove for Measuring Hand Finger Flexion for Rehabilitation Purposes. Instrumentation and Measurement, IEEE Transactions on. 62. 3308-3314. 10.1109/TIM.2013.2272848.