IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Decoding Hand Gestures for Individuals with Different Abilities Using the Convolutional Neural Network

POTHURAJU RAJU [1*], Dr. P. KRISHNA SUBBA RAO[2*]

[1*] Research Scholar, Department of Computer Science and Systems Engineering, AUCE(A), Andhra University, Visakhapatnam, Andhra Pradesh, India

[2*] Professor, Department of Computer Science and Engineering, Gayatri Vidya Parishad College of Engineering(A), Visakhapatnam, Andhra Pradesh, India

Abstract: Differently abled people like deaf and dumb people use sign language for communicating with others. It is usually found difficult to communicate with them due to a lack of understanding of the universal sign language. This project aims to develop an application that will translate sign language into text, thereby reducing the communication gap between normal people and people with speech impairment. This application was developed using a deep learning algorithm which is CNN (Convolutional Neural Network). The model was trained on ASL (American Sign Language) based gestures. ASL is a universal language and contains 26 alphabets of well-distinguished images that can be used to train the model. The objective of this project is for the camera attached to the computer will capture the gestures of the hand and the feature extraction is used to recognize the hand gesture then based on the hand gesture the required text will be displayed on the screen. This application aims at bridging the gap in the process of communication between Deaf and Dumb people from the rest of the world.

Index Terms **- Hand Gestures, Sign language into text, Convolution Neural Network (CNN), Communication, Computer Vision, ASL (American Sign Language)**

I. INTRODUCTION

Communication between people will be efficient if the people involved in the communication are aware of the language used. We as human beings talk with each other in order to convey our thoughts, and share our experiences. But this is not the case with everyone. There are people with disabilities like deaf and dumb people for whom the mode of communication is different. Sign language paves the path for them in order to communicate with the rest of the world. But most of us are not aware of sign language and lack of knowledge of it makes us difficult to communicate with deaf and dumb people. In order to bridge this gap, A model is created which will detect the gesture shown in front of the webcam and displays the corresponding English alphabet on the screen. This model is trained using ASL (American Sign Language) based gestures. This system is built using OpenCV, and Keras uses Deep Learning to communicate with differently-abled people in real-time video streams.

1.1 PROBLEM STATEMENT

Hand gesture detection in real-time video streams with the help of deep learning and computer vision concepts in order to identify and decode the American sign language-based gestures which help to communicate with the deaf and dumb people.

II. TECH STACK

2.1 DEEP LEARNING

Deep Learning is a subfield of machine learning which is based on learning and analyzing its own with the help of dense neural networks which are actually designed based on the structure of the human brain. Deep learning algorithms attempt to draw similar conclusions as humans would by continually analyzing data. For our system, we, have used a deep learning algorithm named convolution neural networks for processing the input images.

2.2 CONVOLUTION NEURAL NETWORK (CNN)

A convolutional neural network is a class of deep neural networks primarily used to analyze visual images. In our system, there is a need to analyze gestures. With the help of CNN, we have achieved that. There are basically three layers in CNN.

1. Convolutional layer
2. Pooling layer
3. Fully Connected layer.

The convolutional layer is used to detect the features given in the input whereas the pooling layer consists of a pooling operation that aims in reducing the size of images while preserving the important characteristics. The fully connected layer connects every input neuron to every output neuron. This layer receives an input vector and produces a new output vector

2.3 OPENCV

OpenCV is a great tool for image processing and computer vision concepts. For detecting the gestures made in front of the webcam it uses the OpenCV library. We have used a method called video capture which is used to capture the video by webcam.

2.4 KERA'S

Keras is a deep-learning framework for python. It is one of the powerful libraries for developing and evaluating deep learning models. We have used the load model method from Keras. model library for loading our model to detect the gesture made in the video stream.

III. METHODOLOGY

3.1 Data Gathering

Data gathering entails acquiring information from a variety of origins. We're taking an American Sign Language Dataset (ASL) in our project. A dataset is a collection of data; in this case, we are having 26 English alphabets of different positions arranged in 26 folders; a total of 17113 images.

3.2 Pre−Processing

Before the data is trained and tested, it must first be pre-processed. There are different steps in preprocessing of data.

1. Background Subtraction
2. Gray Scale Conversion

3.3 Dividing Dataset into Training and Test Data

The machine learning models are trained by dividing the set of data into a Testing set and a Training set to see how accurately they will predict. As a general guideline, 80% of the dataset should be given to the training set and the remaining available 20% should be assigned to the test set. It's important to observe that no photos from a set of training data appear in test set data; if this happens, it'll be difficult to tell if the algorithm learned to generalize from the training set or merely memorized the data.

Training Dataset: It is a part of the dataset utilized for training the model.

Testing Dataset: This is a part of the dataset utilized for testing the model.

3.4 Designing the Model

1. Loading the dataset and performing image preprocessing
2. Using a Sequential classifier from Keras to construct a basic model
3. Parameterizing the model
4. Model compilation
5. Instructing our model and live prediction.

3.5 Training and Testing the Model

There are phases in testing the model to ensure that it can predict well. The initial stage is to predict the testing set. Table 1 shows the results we got during training the model which includes 20 iterations for checking the loss and accuracy.

Table 1 shows that the accuracy improves and loss decreases. As the line of accuracy is steady, it means that there is no need to repeat the iteration process to improve the accuracy of the model.

3.6 Model Implementation

In the real-time video, we use our model. The video is considered a frame and it changes from second to second. The frame gets compared with the model and shows the output when the person shows different signs. By using our model that has already been created the input image will be processed and predicts the output.

ALGORITHM 1: Pre-processing and Training on Dataset

INPUT: Images as well as their pixel values.

OUTPUT: Trained Model

STEP 1: Load Images and their pixel values.

STEP 2: Process the images i.e., converting them into grayscale, and extracting edges.

STEP 3: Load the images and their respective labels.

STEP 4: Splitting data into training and testing batches.

STEP 5: ad Sequential model from Kera's. Training the model by dividing it int batches and controlling the weights using the Adam optimizer.

STEP 6: Save the model for live prediction.

ALGORITHM 2: Deployment of Hand Gesture Decoding Model

INPUT: Real-time video capture

OUTPUT: Classifying the hand gestures

STEP 1: Load the saved model from the disk. Also, load the video capture from OpenCV.

STEP 2: Real-time classification is required. Load the OpenCV real-time stream.

STEP 3: Hand gestures are detected through the grayscale window and the corresponding output is displayed on the screen

STEP 4: When we press ESC, then the stream will come to end.

IV. RESULTS AND DISCUSSION

4.1 Results

"Epoch"	"Loss"	"Accuracy"	"Val_Loss"	"Val_acc"
(1\20)	.3872	.8734	.0308	.9954
(2\20)	.2463	.9157	.0174	.9961
(3\20)	.1959	.9361	.0105	.9983
(4\20)	.1586	.9476	.0110	.9973
(5\20)	.1279	.9550	.0056	.9990
(6\20)	.1177	.9619	.0031	.9993
(7\20)	.1057	.9659	.0019	.9995
(8\20)	.0815	.9730	.0042	.9995
(9\20)	.0935	.9702	.0025	.9995
(10\20)	.0696	.9780	.0015	.9995
(11\20)	.0691	.9782	.0029	.9993
(12\20)	.0628	.9811	.0014	.9998
(13\20)	.0600	.9826	.0012	.9998
(14\20)	.0704	.9793	.0012	.9998
(15\20)	.0541	.9830	.0021	.9993
(16\20)	.0599	.9823	.0011	.9998
(17\20)	.0414	.9880	.0015	.9998
(18\20)	.0516	.9858	.0012	.9993
(19\20)	.0412	.9878	.0015	.9990
(20\20)	.0386	.9881	.0011	.9998

Training the model:

Step 1: As part of training the model, we have measured 20 epochs. The accuracy is gradually growing, while the value of the loss is gradually reducing. So, our model is working well.
Step 2: The accuracy and losses of hand gestures is observed in the below table. Our model's overall accuracy is as given
Step 3: The below graph is the plotting representation of our model. The values we got in Table 1 are represented in the form of a graph.

Figure 1

Hand Gesture detector:

Step 1: When we run the model the main window (figure 2) will be displayed. From the main window, we can navigate to the live detection window as well as the team details window
Step 2: When we click on the live detection button the window (figure 3) will be displayed.
Step 3: The hand gesture shown in front of the webcam is formatted by converting it into grayscale followed by detecting the edges
Step 4: The gesture shown is detected and the corresponding character is displayed

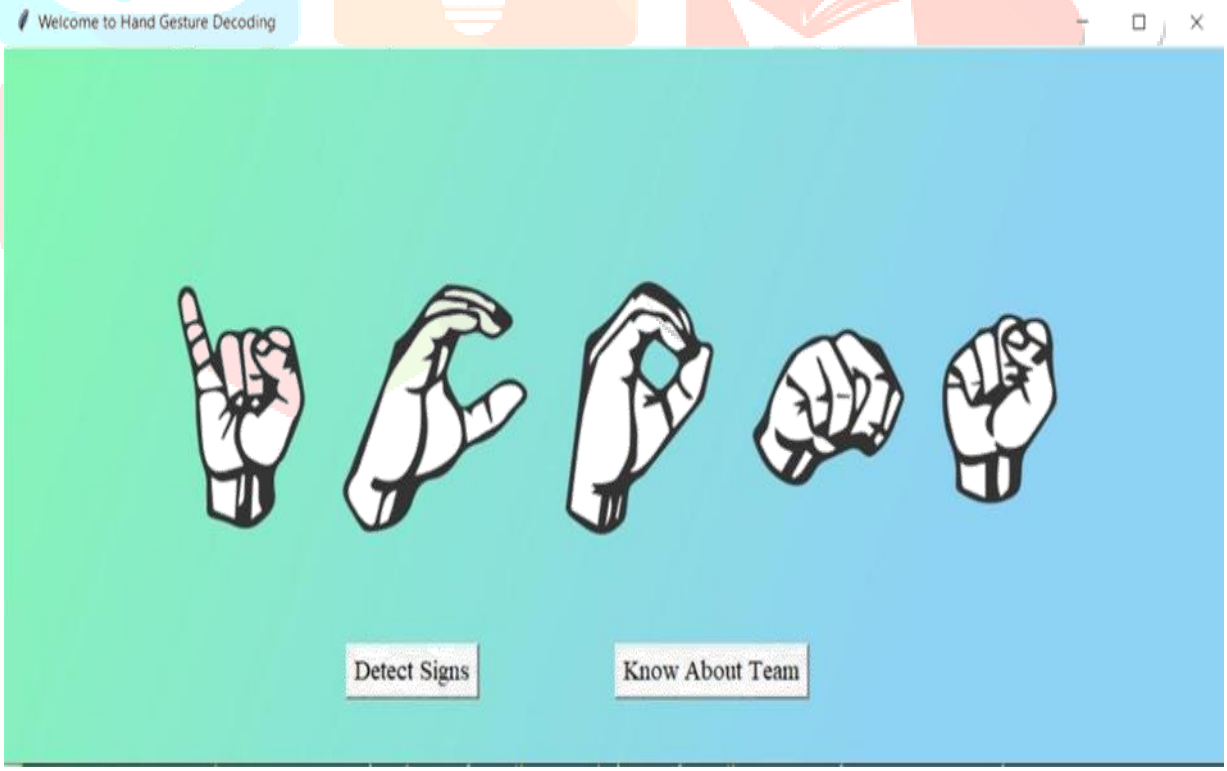


Figure 2 - Main window

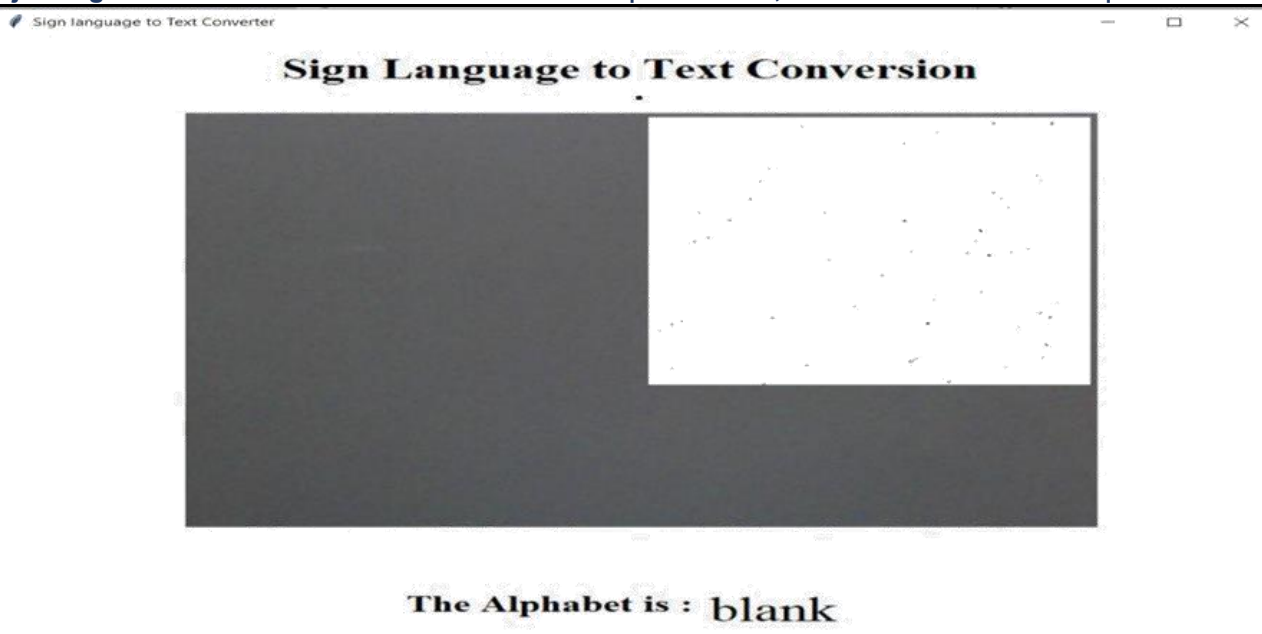
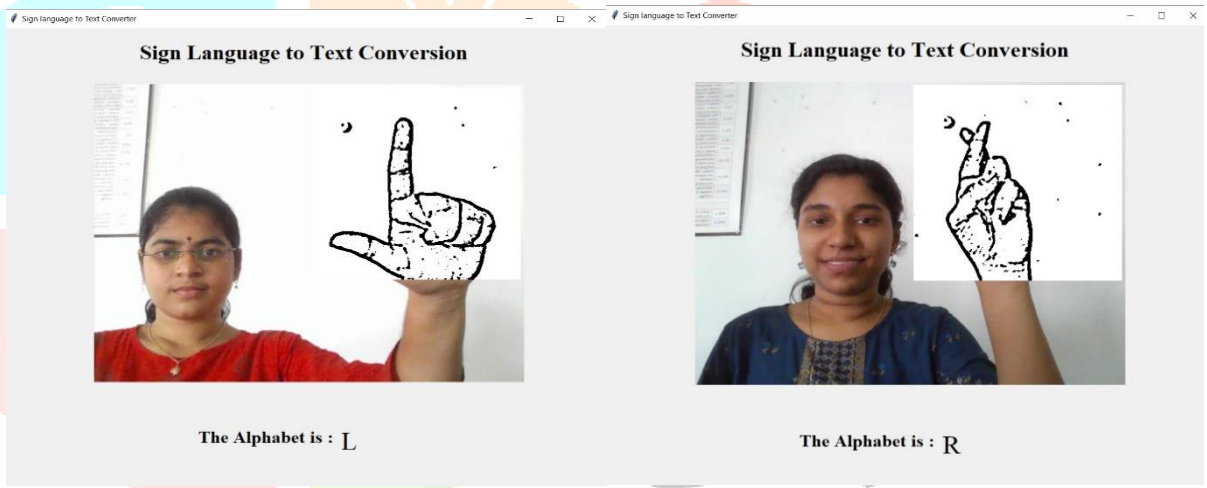
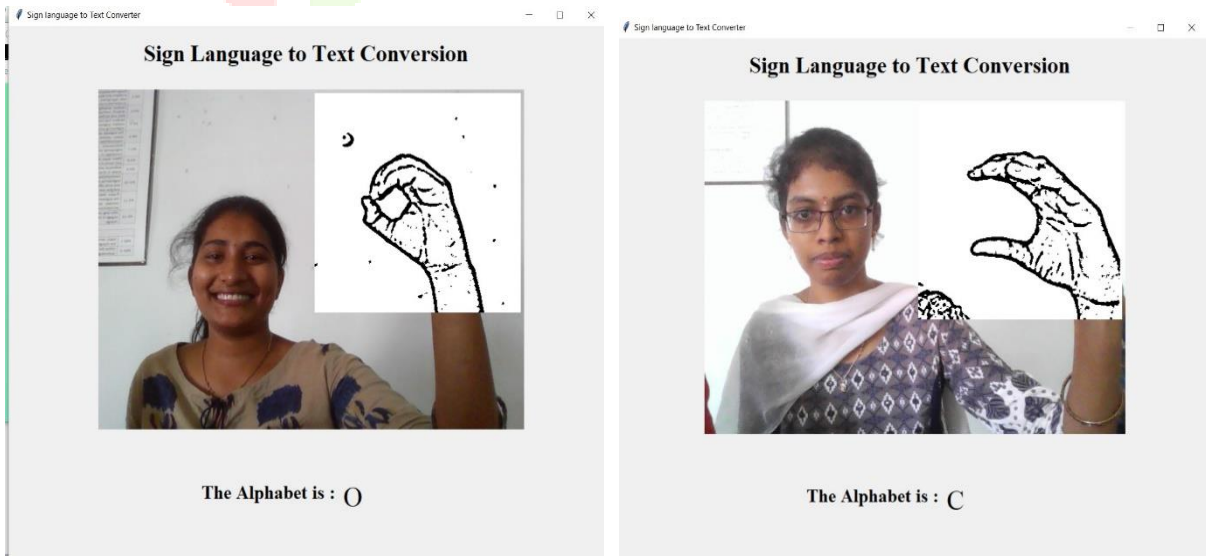


Figure 3 - Live detection window

Some Live Detection Images:





V. REFERENCES

[1] Chen, L.; Fu, J.; Wu, Y.; Li, H.; Zheng, B. Hand Gesture Recognition Using Compact CNN via Surface Electromyography Signals. Sensors 2020

[2] Do, N.; Kim, S.; Yang, H.; Lee, G. Robust hand shape features for dynamic hand gesture recognition using multi-level feature LSTM. Appl. Sci. 2020

[3] Pu Tu1, Chen Huang and Jinxia Zhu1 A Hand Gesture Recognition Algorithm Based on Multi-scale Hybrid Features : Pu Tu et al 2022 J. Phys.: Conf. Ser. 2218 012038

4. S. Shajun Nisha, M. Nagoor Meeral, in Handbook of Deep Learning in Biomedical Engineering, 2021

5. Mohammed, M.A.; Abdulkareem, K.H.; Mostafa, S.A.; Ghani, M.K.A.; Maashi, M.S.; Garcia-Zapirain, B.; Oleagordia, I.; Alhakami, H.; Al-Dhief, F.T. Voice pathology detection and classification using convolutional neural network model. Appl. Sci. 2020, 10, 3723. [Google Scholar] [CrossRef]

6.. Pedoeem, J.; Huang, R. YOLO-LITE: A Real-Time Object Detection Algorithm Optimized for Non-GPU Computers. In Proceedings of the IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 10–13 December 2018; pp. 2503–2510.

7. Fong, S.; Liang, J.; Fister, I.; Fister, I.; Mohammed, S. Gesture Recognition from Data Streams of Human Motion Sensor Using Accelerated PSO Swarm Search Feature Selection Algorithm. J. Sens. 2015, 2015, 205707.

8. Mambou, S.; Krejcar, O.; Maresova, P.; Selamat, A.; Kuca, K. Novel Hand Gesture Alert System. Appl. Sci. 2019.

9. Mezari, A.; Maglogiannis, I. An easily customized gesture recognizer for assisted living using commodity mobile devices. J. Healthc. Eng. 2018.

10.https://www.sciencedirect.com/topics/engineering/convolutional-neural-network

11.https://towardsdatascience.com/what-is-deep-learning-and-how-does-it-work-2ce44bb692ac

12. https://towardsdatascience.com/understand-the-architecture-of-cnn-90a25e244c7

13. Tan, Y.S.; Lim, K.M.; Lee, C.P. Hand gesture recognition via enhanced densely connected convolutional neural network. Expert Syst. Appl. 2021

14. Wilson FR. The hand: How its use shapes the brain, language, and human culture. New York: Pantheon Books; 1998.