American Hand Sign Recognition Detection and Recognition System

Raksha Beriya , Prof. K. K. Chhajed

Department of Computer Science & Engineering, P. R. Pote (Patil) Education & Welfare Trust's Group of Institutions, College of engineering & Management, Amravati.

Abstract

The only means of communication for someone who cannot hear or speak is through sign language. The use of sign language by people with physical disabilities is a great help in communicating their ideas and feelings. With the help of computer vision and neural networks we can detect the signs and give the respective text output. The project introduces an application of computer vision for American Hand sign detection. Research was carried out on a number of algorithms that could best differentiate a hand gesture. It was found our Long ShortTerm Memory. We use the Media-pipe Hand landmark detection algorithm to detect the hand landmarks plot the hand coordinates. Later the hand coordinates are mapped into NumPy array (Npy)format and feed to Long Short Term Memory model. The model predicts in real time and is able to convert text to speech.

Keywords: Computer vision, LSTM, Hand gestures, Numpy array format

1.Introduction

According to figures from the World Health Organization [1], by 2050, it is predicted that approximately 2.5 billion people will have some degree of hearing loss, and at least 700 million of them, or 1 in 10 people, will need hearing rehabilitation.Due to dangerous listening habits, almost 1 billion young individuals are at risk of developing permanent, preventable hearing loss. Sign language serves a link between deaf and dump people toc communicate people around them and vice versa. We created a method to translate sign language into text and speech to make it easier for regular people to comprehend and communicate with these disabled persons.The users have benefited from recent advancements in computer software and related hard labour technology. Physical gestures are a potent form of communication in daily life. They can effectively communicate a wide range of information and emotions. Waving one's hand side to side, for instance, might signify anything from "happy goodbye" to "caution". Another area where most human computer dialogues fall short is the full power of physical gesture.

One of the fundamental and significant issues in computer vision is the topic of hand gesture recognition. The development of automated human interaction systems that include hand processing tasks like hand detection, hand recognition, and hand tracking has been made possible by recent advancements in information technology and media.Identifying and locating a hand in an image is the initial step in any hand processing system. The variety in stance, orientation, location, and scale made the hand detection task difficult. Additionally, various lighting conditions increase variation.

We have collected static images of all alphabet's signs in American Sign Language but two of the alphabets 'J' and 'Z' consist of motion but we tried to collect static images of both of these. All the American hand signs are shown in Figure 1.

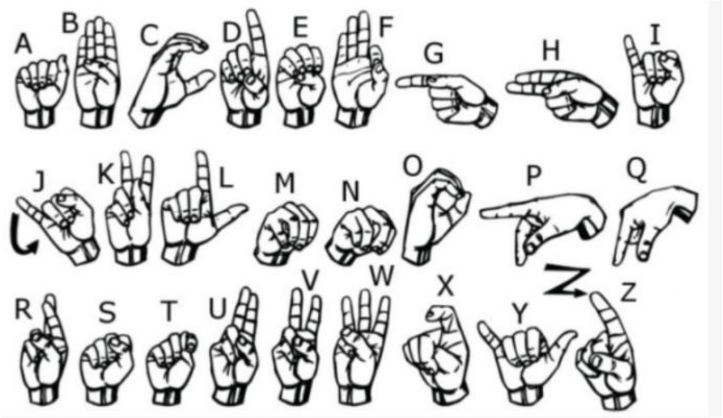


Figure 1 American Sign Language for all 26 alphabets.

We used Media pipe hand land marks detection for plotting and extracting keypoints of hand gestures. Figure 2 demonstrates how media pipe detects and localizes the key points. This task operates on image data with a machine learning model as static data or a continuous stream and outputs hand landmarks in image coordinates. It detects 21 hand-knuckle coordinates within the detected hand regions.

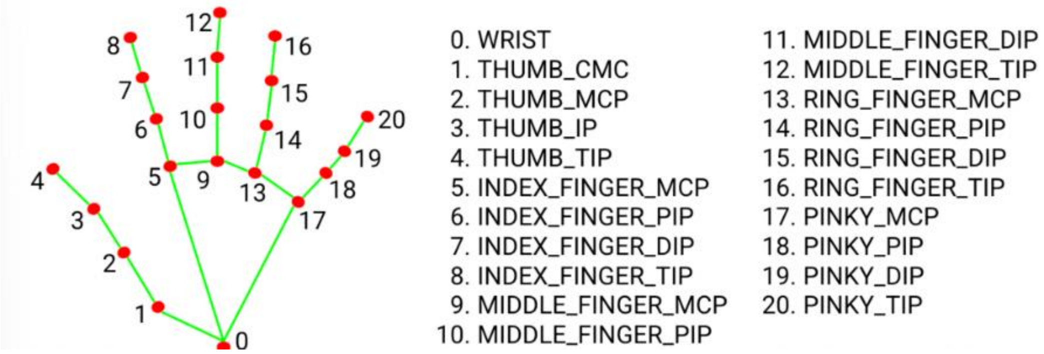


Figure 2 Localization of key points using media pipe

2. Literature Survey

For both dynamic and static hand gestures, a number of techniques are suggested. [2] Pujan Ziaie suggested a method in which one first determined how similar certain motions were, and then using the Bayesian Interface Rule, assigned probability to each.

These were used to estimate invariant classes using a KNN (KNearest Neighbour) modification. These classes are made up of Hu-moments with geometrical characteristics utilised for categorization, such as rotation, transformation, and scale in variation. This method worked quite well and produced findings that were 95% correct.[3] Pujan Ziaie also put out a comparable method known as the Locally Weighted Naive Bayes Classifier, which classifies data using HU-moments and a modified version of the KNN (k-Nearest Neighbour) algorithm. According to classification data, our approach was 93% accurate for diverse lighting circumstances and users. [4] Rajat Shrivastava put up a technique for feature extraction that made use of HU moments and hand orientation. For recognition, the Baum Welch algorithm was employed. 90% of the time the procedure is accurate. [5] Neha S. Chourasia, Kanchan Dhote, and Supratim Saha proposed a method that used a hybrid feature descriptor that included HU invariant moments and SURF. KNN and SVM were employed for classification. They had a 96% accuracy rate. [6] A K-L Transform-based hand gesture recognition system was proposed by Joyeeta Singa. This system had five steps, including hand edge detection using the Canny edge detector, feature extraction using the K-L Transform, and classification. Image acquisition, converting RGB to HSV, filtering, smoothing, finding the biggest BLOB, binary image, and image acquisition. [7,8] Huter proposed a system that uses Zemike moments (0 extract image features and used Hidden Markov Model for recognition. [9,10] Raheja proposed a technique that scanned the image all directions 13 to find the edges of finger tips. [11,12] Segan proposed a technique that used edges for feature extraction. This reduces the time complexity and also help for removing noise.

3. Methodology

This study aims to create or develop a system that is more accurate in detecting hand gestures using hand landmarks detection algorithm of media pipe.

3.1 Data collection

The Data collected was from The American Sign Language Letter database of hand gestures represents 26 alphabets A-Z. The data collected had 100 images of each class i.e. 200 images in total. These are the static images of hand gestures, but for letters such as J and Z contains motion. Figure 2 is a sample dataset.

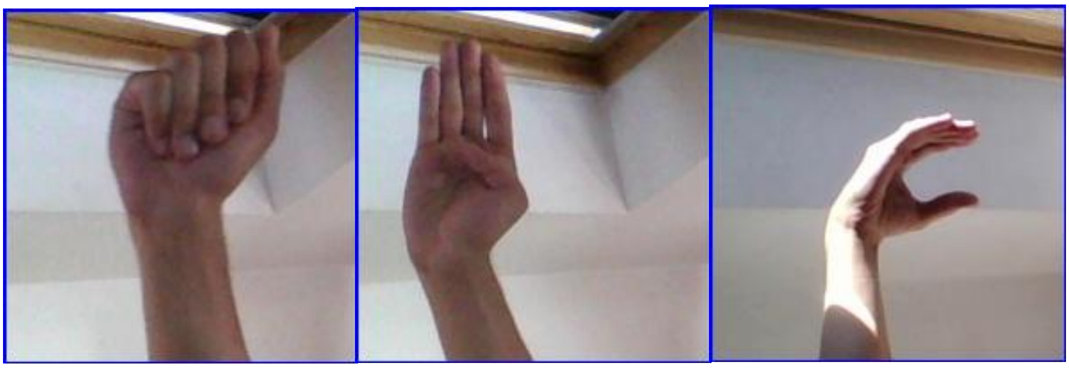


Figure 2 Sample dataset of American Hand Sign detection

3.2 Training Flow

Training flow Following Fig 3 is the training flow of the project

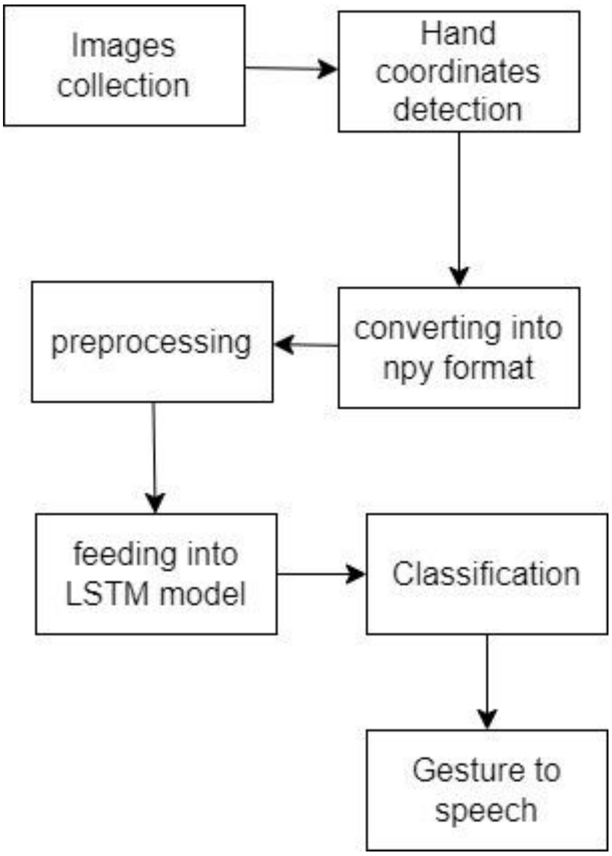


Figure 3 Training flow of American Hand gesture detection

3.3 Hand landmarks detection and key points extraction

When a video is streamed and a frame is captured Hand Land marker uses the bounding box defined by the hand landmarks model in one frame to localize the region of hands for subsequent frames. Figure 4 displays how landmarks are localized on hands.

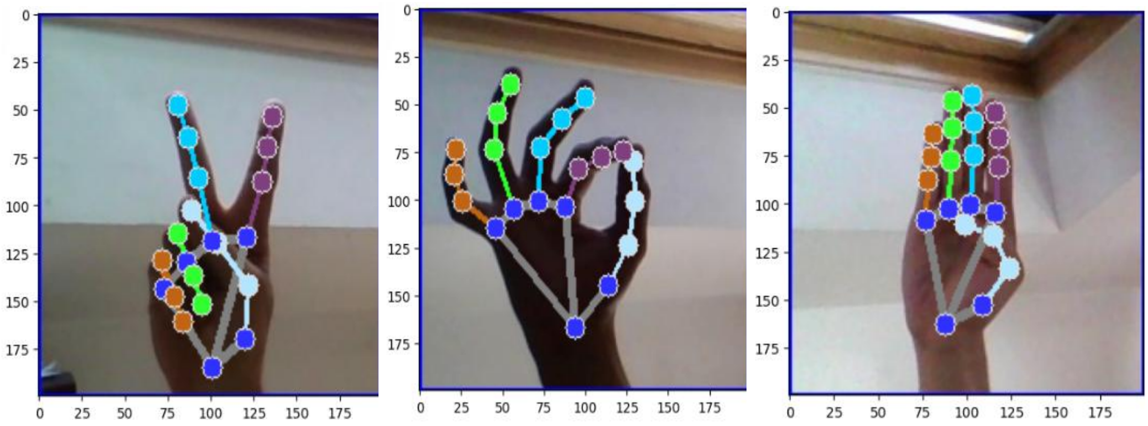


Figure 4 Displays how landmarks are localized on hands

The extracted key points later then converted into Numpy format. A simple format for saving numpy arrays to disk with the full information about them. The .npy format is the standard binary file format in NumPy for persisting a *single* arbitrary NumPy array on disk.

We convert 26 alphabets i.e 2600 images into numpy format. Each alphabet contains 15 sequences and each sequence consist 15 no of numpy format files.Thus in total we have 390 numpy format files.

3.4 Model Architecture

We already converted our key points extracted with the help of mediapipe hand detection model into numpy format and then we feed the training images of 26 alphabets into our LSTM model. Fig 5 demonstrates LSTM architecture [10].

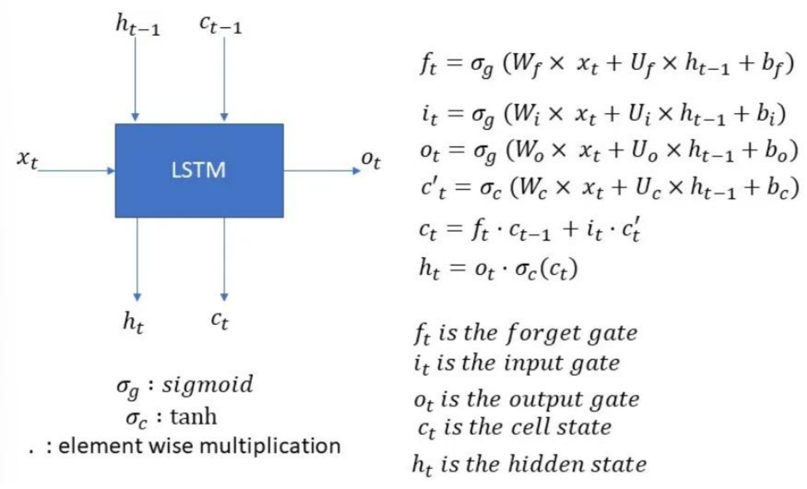

h_{t-1}
c_{t-1}
x_t
LSTM
o_t
h_t
c_t
$f_t = \sigma_g (W_f \times x_t + U_f \times h_{t-1} + b_f)$
$i_t = \sigma_g (W_i \times x_t + U_i \times h_{t-1} + b_i)$
$o_t = \sigma_g (W_o \times x_t + U_o \times h_{t-1} + b_o)$
$c'_t = \sigma_c (W_c \times x_t + U_c \times h_{t-1} + b_c)$
$c_t = f_t \cdot c_{t-1} + i_t \cdot c'_t$
$h_t = o_t \cdot \sigma_c(c_t)$
σ_g : sigmoid
σ_c : tanh
. : element wise multiplication
f_t is the forget gate
i_t is the input gate
o_t is the output gate
c_t is the cell state
h_t is the hidden state

Fig 5 LSTM architecture

For Our task of hand gesture detection of American Hand signs, we have used LSTM model. Fig 6 demonstrates model architecture of Hand gesture detection.

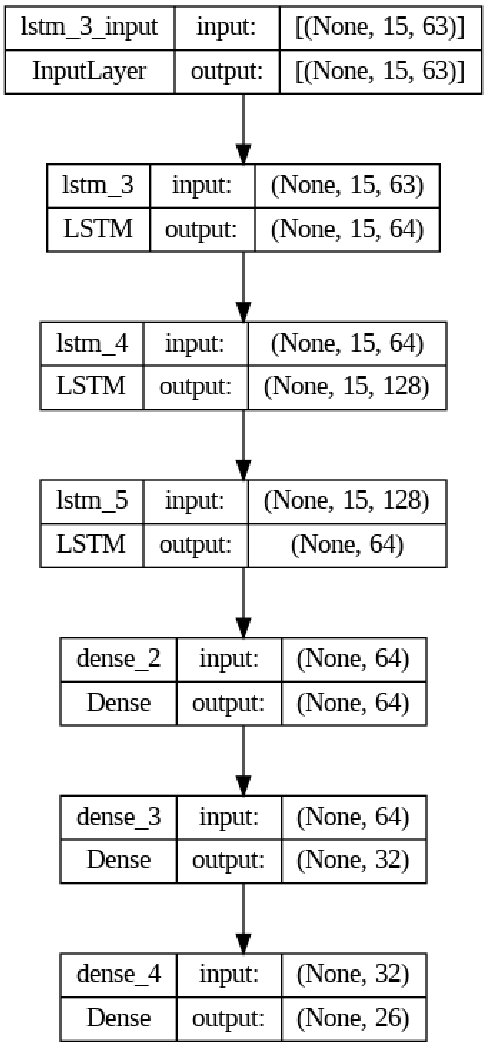

lstm_3_input | input: | [(None, 15, 63)]
InputLayer | output: | [(None, 15, 63)]
lstm_3 | input: | (None, 15, 63)
LSTM | output: | (None, 15, 64)
lstm_4 | input: | (None, 15, 64)
LSTM | output: | (None, 15, 128)
lstm_5 | input: | (None, 15, 128)
LSTM | output: | (None, 64)
dense_2 | input: | (None, 64)
Dense | output: | (None, 64)
dense_3 | input: | (None, 64)
Dense | output: | (None, 32)
dense_4 | input: | (None, 32)
Dense | output: | (None, 26)

Fig 6 Model Architecture

Here is the description of each layer in model architecture.

- In this case, the input shape to the LSTM layer is '(batch_size, timesteps, input_dim)' ; where 'batch_size' is the number of samples in each batch, 'timesteps' is the number of time steps in the input sequence, and 'input_dim' is the number of features in each time step.
- As we have described 'batch_size=64' and 'timesteps=15', the output shape of the LSTM layer would be ('64,15,64'), since we have specified 'return sequences=True' and used 64 LSTM units in this layer.The Output shape of the Dense layer would be ('64,15,63'), since we have specified 63 output unit in the layer.
- The second layer we have LSTM 128, return_sequences = True, activation='relu'. Here 128 is the number of LSTM units (also called cells or neurons) in this layer.It

determines the capacity of the layer to capture temporal patterns in the inpt sequence. Return_sequences=True argument specifies to return the output sequence from each time step of the layer.

- Third layer again we have 64 neurons in our layer whereas return sequences as False indicates the layer should return just the output from the last step and activation function relu.
- Fourth layer again we have 64 neurons in this layer with activation function is relu.
- Fifth layer we decrease the number of neurons from 64 to 32 decreasing the number of parameters in this layer.
- Lastly, we have added a dense layer where we will have 26 neurons as we have 26 alphabets in our dataset.Here we have softmax function, softmax is a mathematical function that converts a vector of real numbers into a probability distribution.Given we have our vector of z=[z1,z2,z3---z26]' where z1 to z26 are the 26 alphabets, the softmax function computes a vector of s=[s1,s2—s26]. Where si is the probability that the input belongs to corresponding class 'i'.

Additionally, categorical cross entropy was employed as our loss function. It assists with multi-class classification jobs in neural networks. It calculates the discrepancy between the actual probability distribution of classes and that which was expected. There are 26 possibilities for each letter in this list.

The Adam optimizer is an extension of the stochastic gradient descent (SGD) optimizer that incorporates adaptive learning rate and momentum. It is used in deep learning to update the weights of a neural network during training.

3.5 Live Prediction

Live prediction involves using the trained hand sign prediction model to identify American hand signs in real-time video streams. We first created a 'VideoCapture' object to connect to the camera with index 0 (which is typically the built-in camera on most computers). We then check if the camera is opened successfully using the 'isOpened()' method.Next, we don't need our entire images just a part of frame where we have hand sign. So thus, we crop our frames into shape of rectangle of size (0,40) and (300,400).

Then we set model_complexity=0, min_detection_confidence=0.5, min_tracking_confidence =0.5 of our mediapipe model. We use cv2.cvtColor() is a function in the OpenCV library that is used to convert an image from one color space to another. It takes two arguments: the image to be converted and the target color space. We first converted from BGR to RGB and again back to RGB to BGR.We now extract key points from the frames of actions in hand signs and return the results of extracted twenty hand landmarks.We loaded our trained model and we predicted the frame with the help of extracted key points.Figure 7 displays the live prediction of alphabets and text to voice of predicted alphabets.

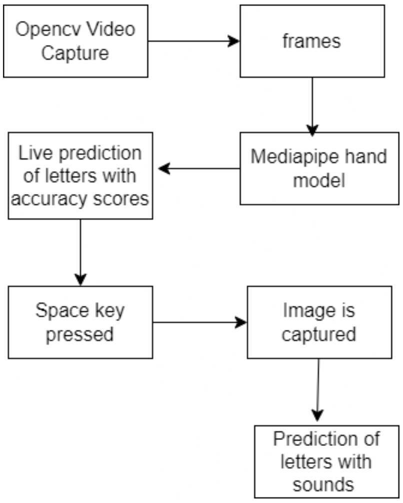

Opencv Video Capture
frames
Mediapipe hand model
Live prediction of letters with accuracy scores
Space key pressed
Image is captured
Prediction of letters with sounds

Fig 7 Live Prediction of alphabets in our paper

We predicted the frames using our trained model and we predicted American sign alphabets, we used cv2.putText to show the accuracy scores on the window.

Additionally, we have one unique feature where if the Space key is pressed on the keyboard, then a signal frame is captured and cropped into shape of rectangle and then passed the cropped frame to the trained model. Once the model predicts the frame the prediction is converted into the text and further the text is converted into speech.

4. Results and Analysis

4.1 System Configuration

Operating System: Windows, Mac, Linux

SDK: OpenCV, TensorFlow, Keras, Numpy

4.2 Hardware Requirements

The Hardware interfaces Required are:

Camera: Good quality 2MP

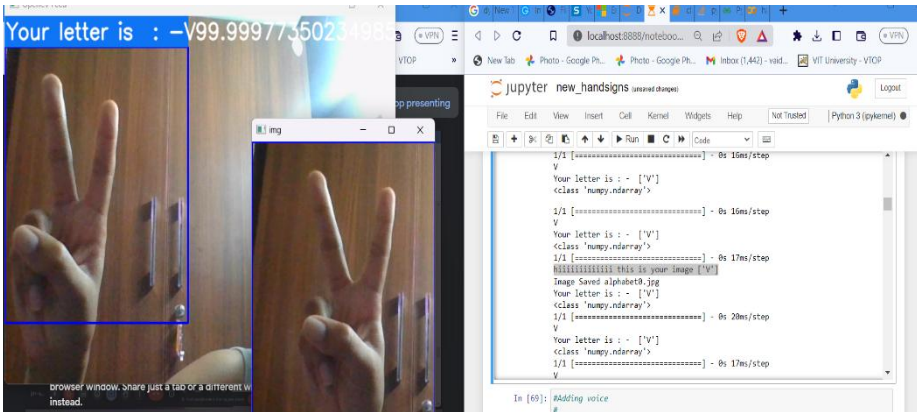
Ram: Minimum 4GB or higher

Processor: Intel or Amd Ryzen

4.3 Results

We received our training accuracy of 100% with very minimized categorical loss after 200 epochs. Later we evaluated our model with test images and we achieved the accuracy of 100% while categorical loss 0.02.

We further tested our model with the help of live prediction. Figure 8,9,10 is example images of live prediction.

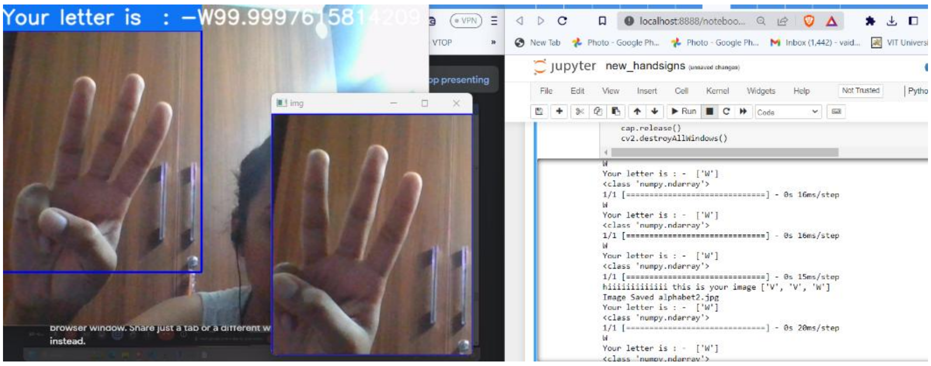

Your letter is : –V99.99977350234985
img
jupyter new_handsigns (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code

```
1/1 [==============================] - 0s 16ms/step
V
Your letter is : -  ['V']
<class 'numpy.ndarray'>

1/1 [==============================] - 0s 16ms/step
V
Your letter is : -  ['V']
<class 'numpy.ndarray'>
1/1 [==============================] - 0s 17ms/step
hiiiiiiiiiiii this is your image ['V']
Image Saved alphabet0.jpg
Your letter is : -  ['V']
<class 'numpy.ndarray'>
1/1 [==============================] - 0s 20ms/step
V
Your letter is : -  ['V']
<class 'numpy.ndarray'>
1/1 [==============================] - 0s 17ms/step
V
```

```
In [69]: #Adding voice
         #
```

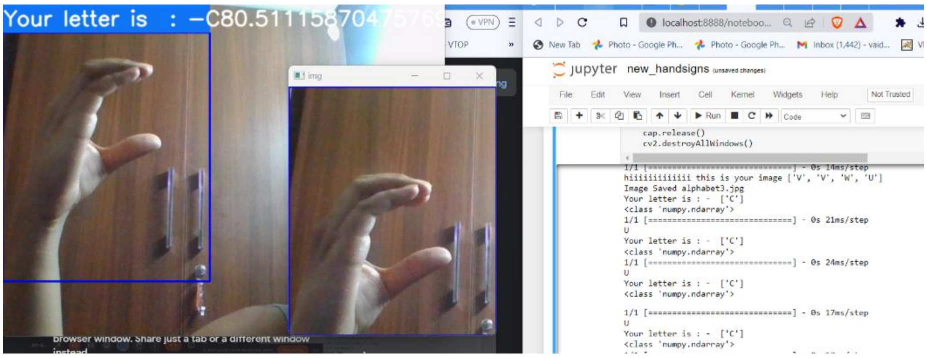
Fig 8 American Hand sign for v actual and predicted with accuracy score.


Your letter is : –W99.99976158142090
img
jupyter new_handsigns (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Run
Code

```
    cap.release()
    cv2.destroyAllWindows()
```

```
W
Your letter is : -  ['W']
<class 'numpy.ndarray'>
1/1 [==============================] - 0s 16ms/step
W
Your letter is : -  ['W']
<class 'numpy.ndarray'>
1/1 [==============================] - 0s 16ms/step
W
Your letter is : -  ['W']
<class 'numpy.ndarray'>
1/1 [==============================] - 0s 15ms/step
hiiiiiiiiiiii this is your image ['V', 'V', 'W']
Image Saved alphabet2.jpg
Your letter is : -  ['W']
<class 'numpy.ndarray'>
1/1 [==============================] - 0s 20ms/step
W
Your letter is : -  ['W']
<class 'numpy.ndarray'>
```

Fig 9 American Hand sign for W with actual and predicted with accuracy score.


Your letter is : –C80.51115870475769
img
VTOP
localhost:8888/noteboo...
New Tab
Photo - Google Ph...
Photo - Google Ph...
Inbox (1,442) - vaid...
jupyter new_handsigns (unsaved changes)
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Trusted
Run
Code
browser window. Share just a tab or a different window instead

```
    cap.release()
    cv2.destroyAllWindows()
```

```
1/1 [==============================] - 0s 14ms/step
hiiiiiiiiiiii this is your image ['V', 'V', 'W', 'U']
Image Saved alphabet3.jpg
Your letter is : -  ['C']
<class 'numpy.ndarray'>
1/1 [==============================] - 0s 21ms/step
U
Your letter is : -  ['C']
<class 'numpy.ndarray'>
1/1 [==============================] - 0s 24ms/step
U
Your letter is : -  ['C']
<class 'numpy.ndarray'>

1/1 [==============================] - 0s 17ms/step
U
Your letter is : -  ['C']
<class 'numpy.ndarray'>
```

Fig 10 American Hand sign for letter C with accuracy score.

We also added the text to speech part where once the letter is predicted the text to speech engine will say the predicted letter.

5. Conclusion

Applications today require a variety of pictures as sources of data for clarification and analysis. To carry out a variety of applications, a number of characteristics must be extracted. Degradation happens when a picture is changed from one form to another, such as when digitising, scanning, sharing, storing, etc. The produced picture must thus go through a process known as image enhancement, which consists of a collection of techniques meant to increase an image's visual presence. Fundamentally, image enhancement improves the readability or awareness of information in pictures for human listeners while also giving other autonomous image processing systems better input. The picture is then subjected to feature extraction utilising a variety of techniques to improve the image's computer readability.A sign language recognition system is an effective tool for gathering expert knowledge, spotting edges, and combining false information from several sources. Convolution neural network's goal is to obtain the proper categorization.

6. Future Scope

The suggested sign language recognition system may be expanded to recognise gestures and facial expressions in addition to sign language letters. Sentences will be displayed as a more proper translation of language rather than letter labels, which is more appropriate. This improves readability as well. The range of sign languages can be expanded even more if we train the model using phrases and develop a tool that lets us communicate with mentally disabled persons.

References

[1] Deafness and hearing loss -WHO '*https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss'*

[2] Pujan Ziaie, Thomas Muller and Alois Knoll. *A Novel Approach to Hand Gesture Recognition in a Human-Robot Dialog System: Robotics and Embedded Systems Group Department of Informatics Technische Universitat Munchen*.

[3]Pujan Ziaie and Alois Knoll. *An invariant-based approach to static Hand Gesture*

[4] Rajat Shrivastava. *A Hidden Markov Model based Dynamic Hand Gesture Recognition System using OpenCV*: Dept. of Electronics and Communication Engineering Maulana Azad National Institute of Technology Bhopal-462001, India

[5] Neha S. Chourasia, Kanchan Dhote, Supratim Saha. *Analysis on Hand Gesture Spotting using Sign Language through Computer Interfacing: International Journal of Engineering Science and Innovative Technology* (IJESIT) Volume 3, Issue 3, May 2014

[6] Joyeeta Singha, Karen Das. *Hand Gesture Recognition Based on Karhunen Loeve Transform:* Department of Electronics and Communication Engineering Assam Don Bosco University, Guwahati, Assam, India

[7] Hunter, E. Posture estimation in reduced model gesture input systems, *Proceedings of International Workshop on Automated Face and Gestures Recognition*, June 1995

[8] Chaudhary. A., Raheja, J. L.. Das, K.. Raheja, S., *A Vision based Geometrical Method to find Fingers Positions in Real Time Hand Gesture Recognition*, Journal of Software, Academy Publisher, Vol.7, 2012.

[9] Segan, J, *Controlling computers with gloveless gestures in Virtual Reality Systems.* 1993

[10] LSTM architecture –*https://towardsdatascience.com/tutorial-on-lstm-a-computational-perspective-f3417442c2cd*

[11] Kanchan M Pimple, Praveen P Likhitkar, Sagar Pande : *Convolutional neural networks for malaria image classification*

[12] Sagar Pande, Aditya Kamparia, Deepak Gupta: *Recommendations for DDOS Threats Using Tableau*