IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

SIGN LANGUAGE RECOGNITION BY IMAGE PROCESSING

[1]Sabari Priya A, [2]Adrija Nair, [3]Sreejin Madhavan, [4]Kavitha Issac

1,2,3 Students, Department of Electrical and Electronics Engineering, Mar Athanasius College of Engineering, Kothamangalam, Kerala, India

4 Associate Professor, Department of Electrical and Electronics Engineering, Mar Athanasius College of Engineering, Kothamangalam, Kerala, India

Abstract: The paper presents a prototype for a sign language interpreter which can communicate with American Sign Language (ASL). Sign language serves as a visual stimulation of speech and language development. It helps to reduce negative social behaviors, increase social interactions, and develop cognitive structures. With early introduction, sign language provides emotional, social, and academic support for language development. The reliable paradigm is based on building a user-hand gesture-only human-computer interface (HCI). The theories of hand segmentation and the hand detection system are used to construct hand gesture recognition using Python and OpenCV. From the database, the training images are labelled with different names. Each of the images is converted to a binary feature vector. The feature extraction procedure is carried out from the picture collected from real-time video of hand signs using camera and Convolutional Neural Network (CNN). These features are compared with the features of database photos and the system produces output based on the predicted highest similarity. Without a fixed or monochromatic background, this model offers satisfactory accuracy and can be used with the help of an application. The camera is used to record video of the hand gestures of the person, which is then analyzed and converted into textual description. The output is converted to speech, which is heard by the user. The proposed system includes sign language recognition of numbers, alphabets and some common words. Image processing algorithms along with neural networks are used to map the gesture to appropriate text in the training data and hence raw images/videos are converted into respective text that can be read and understood. Such a protocol will undoubtedly be very helpful for bridging the communication gap between those with speaking and hearing abilities and those without them, especially when combined with a large source database.

Keywords **– American Sign Language, Human- Computer Interface, Convolutional Neural Network**

I. INTRODUCTION

Dumb people are usually deprived of normal communication with other people in the society. It has been observed that they find it really difficult at times to interact with normal people with their gestures, as only a very few of those gestures are recognized by most people. Since people with hearing impairment or deaf people cannot talk like normal people, they have to depend on some sort of visual communication in most of the time. Sign Language is the primary means of communication among the deaf and dumb community. Like any other language it has also got grammar and vocabulary but uses visual modality for exchanging information. The problem arises when dumb or deaf people try to express themselves to other people with the help of these sign language grammars but normal people are usually unaware of these grammars. As a result, it has been seen that communication of a dumb person is only limited within his/her family or the deaf community. The importance of sign language is emphasized by the growing public approval and fund accumulation for international projects. In this age of technology, the demand for a computer based system is highly increasing in the dumb community. Interesting technologies are being developed for speech recognition but no real commercial product for sign recognition is actually present in the current market. The idea is to make computers understand human language and develop a user friendly Human Computer Interface (HCI). Making a computer understand speech, facial expressions and human gestures are some steps towards it. Gestures are the non-verbally exchanged information. Since human gestures are perceived through vision, it is a subject of great interest for computer vision researchers. The main aim is to determine human gestures by creating an HCI. Coding of these gestures into machine language demands a complex algorithm.

There are two main steps in building an automated recognition system for human actions. The first step is to extract features from the frame sequences. This will result in a representation consisting of one or more feature vectors, also called descriptors. This representation will aid the computer to distinguish between the possible classes of actions. The second step is the classification of the action. A classifier will use these representations to discriminate between the different actions (or signs). The sign language recognition system is developed based on a vision-based method and uses a webcam for real-time dynamic video input. For training purpose, a database is created using sufficient number of images for each sign of ASL. The theories of hand segmentation and the hand detection system can be used to construct hand gesture recognition using Python and OpenCV. The classifier is trained using the features of the database images. When a user conveys a sign of ASL infront of the webcam, features are taken from the new images and are encrypted as new features. Then these new encrypted features are compared with the trained vocabulary of the classifier. The output confusion matrix shows the precision and exactness of the prophecy. As the system is quite simple and compact, it is highly user-friendly. The system is free from the limitation of uniform background and the nuisance associated with the glove-based method.

II. Hand Gesture Recognition

Sign language recognition is the process of converting the signs and gestures shown by user into text. It is an aspect of human-computer interaction that demonstrates an academic treatise and is vital to popularise the notion of a human-to-human connection that must imply the correlation between the user and the machine. The Computer Vision Study concentrates on gesture recognition in the Open CV framework using the Python language.

III. American Sign Language

The proposed system uses American Sign Language (ASL) data set to identify the sign made by a gesture. American Sign Language is a natural language that serves as the predominant sign language of deaf communities. In sign language, every sign has a meaning assigned to it, so that it becomes easy to understand and interpret by the people. The people, based on their language and the place in which they live, develop discrete and non-identical sign languages. There is no sign language accepted universally. People use various sign languages across the world. When using ASL, only one hand is used. Therefore, it becomes easy for implementing the system. ASL does not depend on any of the spoken languages and it has its own path of development. It makes use of gestures of the hand.

The data set has numbers labelled from zero to nine, 26 alphabets and some words. Every sample of the data set is characterized with its equivalent sign. A unique sign letter corresponds to every sample. The semantics of ASL also deviates from that of English and is different from the English language that people use for communication daily. ASL consists of 26 symbols, known as American manual alphabets. Fig.1 shows a chart that consists of ASL manual alphabets and numbers.

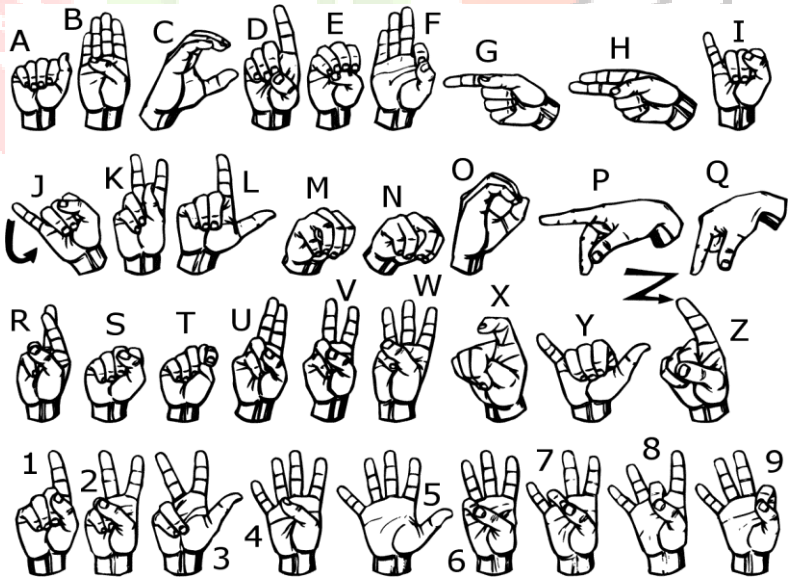

Figure 1: ASL manual letters and numbers

In addition to alphabets and numbers, some words included in the dataset are shown in Fig.2.

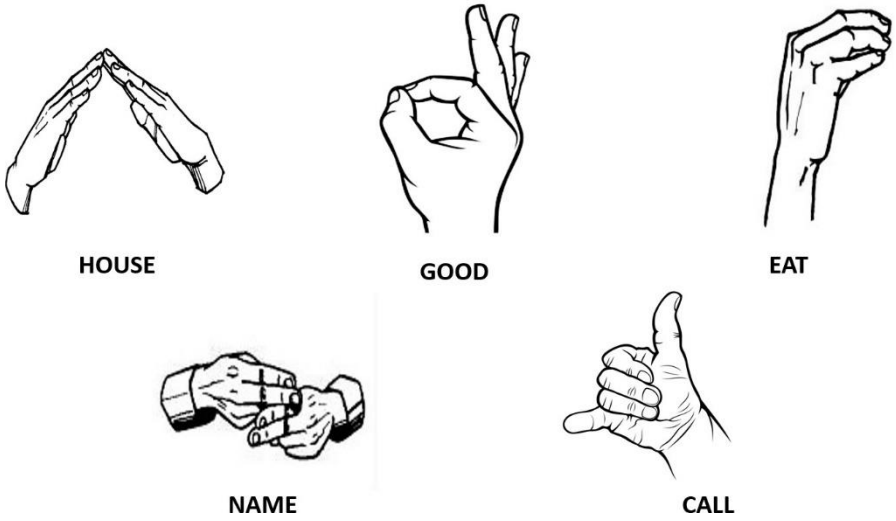

Figure 2: Some ASL words

IV. IMAGE PROCESSING

Image processing is a method to perform some operations on an image, in order to get an enhanced image and to extract some useful information from it. It is a type of signal processing in which input is an image and output may be image or characteristics/features associated with that image. Image processing basically includes thefollowing three steps:

1. Importing the image via image acquisition tools.
2. Analysing and manipulating the image.
3. Output in which result can be altered image or report that is based on image analysis.

There are two types of methods used for image processing namely, analogue and digital image processing. Analogue image processing can be used for hard copies like printouts and photographs. Digital image processing techniques help in manipulation of the digital images by using computers.

Digital image processing consists of the manipulation of images using digital computers. It has various applications ranging from medicine to entertainment, passing by geological processing and remote sensing. Multimedia systems, one of the pillars of the modern information society, rely heavily on digital image processing. Digital image processing consists of manipulation of finite precision numbers. The processing of digital images can be divided into several classes: image enhancement, image restoration, image analysis and image compression.

V. PATTERN RECOGNITION

On the basis of image processing, it is necessary to separate objects from images by pattern recognition technology, then to identify and classify these objects through technologies provided by statistical decision theory. Under the conditions that an image includes several objects, the pattern recognition consists of three phases, as shown in Fig.3.

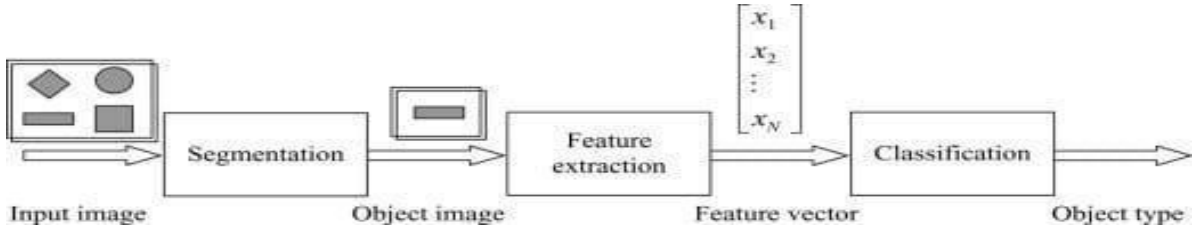

Figure 3: Phases of Pattern Recognition

The first phase includes the image segmentation and object separation. In this phase, different objects are detected and separated from other background. The second phase is the feature extraction. In this phase, objects are measured. The measuring feature is to quantitatively estimate some important features of objects and a group of these features are combined to make up a feature vector during feature extraction. The third phase is classification. In this phase, the output is just a decision to determine thecategory in which every object belongs to. Therefore, for pattern recognition, images are the

input and the output is object types and structural analysis of images. The structural analysis is a description of images in order to correctly understand and judge the important information of images.

VI. System Architecture

The database of numerous images of the signs is created for training the classifiers. The system extracts features from the database images for recognition and these features train the category classifiers. Live video acquired by a webcam is processed by the system using effective image processing techniques. Images are captured after every 35 frames from the video input. These images are encoded by the trained encoder and the encrypted features are saved as a table with the same variable names of the original training data. Based on the trained classification model, this table is then processed to predict the class labels of the images. Fig. 4 shows the architecture of the proposed system.

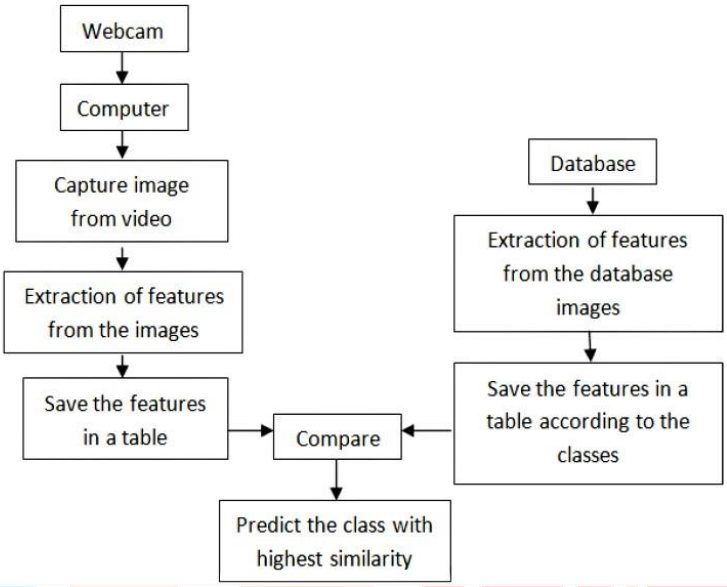

Figure 4: Architecture of the proposed system

VII. Technology Stack

i. TensorFlow

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google.

TensorFlow provides stable Python (for version 3.7 across all platforms) and C APIs; and without API backwards compatibility guarantee: C++, Go, Java, JavaScript and Swift (early release). Third-party packages are available for C#, Haskell Julia, MATLAB, R, Scala, Rust, OCaml and Crystal. New language support should be built on top of the C API. However, not all functionality is available in C yet. Some more functionality is provided by the Python API.

Application: The application for which TensorFlow forms the foundation is the automated image-captioning software, such as Deep Dream.

ii. OpenCV

OpenCV (Open Source Computer Vision Library) is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then Itseez (which was later acquired by Intel). The library is a cross-platform and free for use under the open-source BSD license.

OpenCV's application areas include 2D and 3D feature toolkits, Egomotion estimation, Facial recognition system, Gesture recognition, Human-computer interaction (HCI), Mobile robotics, Motion understanding, Object identification, Segmentation and recognition. To support some of the above areas, OpenCV includes a statistical machine learning library that contains boosting, decision tree learning, gradient boosting trees, expectation-maximization algorithm, k-nearest neighbor algorithm, naive Bayes classifier, artificial neural networks, random forest, Support Vector Machine (SVM) and Deep Neural Networks (DNN). Computer Vision is widely used in robotics, navigation, obstacle avoidance, security applications, biometrics (iris, finger print, face recognition), surveillance-detecting certain suspicious activities or behaviors, autonomous vehicles etc.

OpenCV provides a list of seven free and open source software packages. They are:-

1. OpenCV Functionality
2. Image/video I/O, processing, display (core, imgproc, highgui)
3. Object/feature detection (objdetect, features2d, nonfree)
4. Geometry-based monocular or stereo computer vision (calib3d, stitching, videostab)
5. Computational photography (photo, video, superres)
6. Machine learning and clustering (ml, flann)
7. CUDA acceleration (gpu)

iii. Keras

Keras is an open-source neural-network library written in Python. It is capable of running on top of TensorFlow, Microsoft Cognitive Toolkit, R, Theano, or PlaidML. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular and extensible. It was developed as part of the research effort of project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System), and its primary author and maintainer is Franois Chollet, a Google engineer.

Features: Keras contains numerous implementations of commonly used neural- network building blocks such as layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier to simplify the coding necessary for writing deep neural network code. In addition to standard neural networks, Keras has support for convolutional and recurrent neural networks. It supports other common utility layers like dropout, batch normalization, and pooling.

Keras allows users to productize deep models on smartphones (iOS and Android), on the web, or on the Java Virtual Machine. It also allows use of distributed training of deep learning models on clusters of Graphics Processing Units (GPU) and Tensor Processing Units (TPU) principally in conjunction with CUDA. Keras applications module is used to provide pre-trained model for deep neural networks. Keras models are used for prediction, feature extraction and fine tuning. Trained model consists of two parts - model Architecture and model Weights. Model Weights are large files so we have to download and extract the feature from ImageNet database. Some of the popular pre-trained models are listed below,

1. ResNet
2. VGG16
3. MobileNet
4. InceptionResNetV2
5. InceptionV3

iv. NumPy

NumPy is a library for the Python programming language, adding support for large, multi- dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. The ancestor of NumPy, called Numeric was originally created by Jim Hugunin with contributions from several other developers.

Features: NumPy targets the CPython reference implementation of Python, which is a non-optimizing bytecode interpreter. Mathematical algorithms written for this version of Python often run much slower than compiled equivalents. NumPy addresses the slowness problem partly by providing multidimensional arrays and functions and operators that operate efficiently on arrays, requiring rewriting some code, mostly inner loops using NumPy.

Using NumPy in Python gives functionality comparable to MATLAB since they are both interpreted, and they both allow the user to write fast programs as long as most operations work on arrays or matrices instead of scalars. In comparison, MATLAB boasts a large number of additional toolboxes, notably Simulink, whereas NumPy is intrinsically integrated with Python, a more modern and complete programming language. Moreover, complementary Python packages are available; SciPy is a library that adds more MATLAB-like functionality and Matplotlib is a plotting package that provides MATLAB-like plotting functionality.

Python bindings of the widely used computer vision library OpenCV utilize NumPy arrays to store and operate on data. Since images with multiple channels are simply represented as three-dimensional arrays, indexing, slicing or masking with other arrays are very efficient ways to access specific pixels of an image. The NumPy array act as universal data structure in OpenCV for images, extracted feature points, filter kernels and many more. It vastly simplifies the programming workflow and debugging.

v. Neural Network

A neural network is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates. In this sense, neural networks refer to systems of neurons, either organic or artificial in nature. Neural networks can adapt to changing input; so the network generates the best possible result without redesigning the output criteria.

A neural network works similarly to the human brains neural network. A neuron in a neural network is a mathematical function that collects and classifies information according to a specific architecture. The network bears a strong resemblance to statistical methods such as curve fitting and regression analysis. In this system, Convolutional Neural Network (CNN) has been utilised.

Convolutional Neural Network (CNN) is a special architecture of artificial neural networks, proposed by Yann LeCun in 1988. One of the most popular uses of this architecture is image classification. For example, Facebook uses CNN for automatic tagging algorithms, Amazon - for generating product recommendations and Google - for search through among users photos.

Instead of the image, the computer sees an array of pixels. For example, if image size is 300 x 300. In this case, the size of the array will be 300 x 300 x 3. Here 300 is width, next 300 is height and 3 is RGB channel values. The computer is assigned a value from 0 to 255 to each of these numbers. This value describes the intensity of the pixel at each point. The image is passed through a series of convolutional, nonlinear, pooling layers and fully connected layers, and then generates the output.

VIII. DATABASE CREATION

For detecting the signs from the real-time video images, a database is created witha good quality of images of different signs of ASL (dimension of each image is 227 x 227 pixels and the aspect ratio is 1:1). These sets of images are further used for both training and validation purpose. Fig.5 shows the dataset created for ASL alphabet A.

Figure 5: Dataset for letter A in ASL

The model construction depends on machine learning algorithms. In this system, the machine algorithm used is neural networks. Such an agorithm looks like:

1. Begin with its object: model = Sequential()
2. Add layers with their types: model.add(type_of _layer())
3. After adding a sufficient number of layers, the model is compiled.

At that moment, Keras communicates with TensorFlow for construction of the model. During model compilation it is important to write a loss function and an optimizer algorithm. It looks like: model.compile(loss= name_of_loss_function, optimizer= name_of_optimizer_alg). The loss function shows the accuracy of each prediction made by the model.

IX. FEATURE EXTRACTION

A neural network designed to process multi-dimensional data like image and time series data is called a Convolutional Neural Network (CNN). It includes feature extraction and weight computation during the training process. The name of such networks is obtained by applying a convolution operator which is useful for solving complex op-erations. The true fact is that CNNs provide automatic feature extraction, which is the primary advantage. The specified input data is initially forwarded to a feature extraction network, and then the resultant extracted features are forwarded to a classifier network as shown in Fig.6. The feature extraction network comprises loads of convolutional and pooling layer pairs. Convolutional layer consists of a collection of digital filters to perform the convolution operation on the input data. The pooling layer is used as a dimensionality reduction layer and decides the threshold. During backpropagation, a number of parameters are required to be adjusted, which in turn minimizes the connections within the neural network architecture.

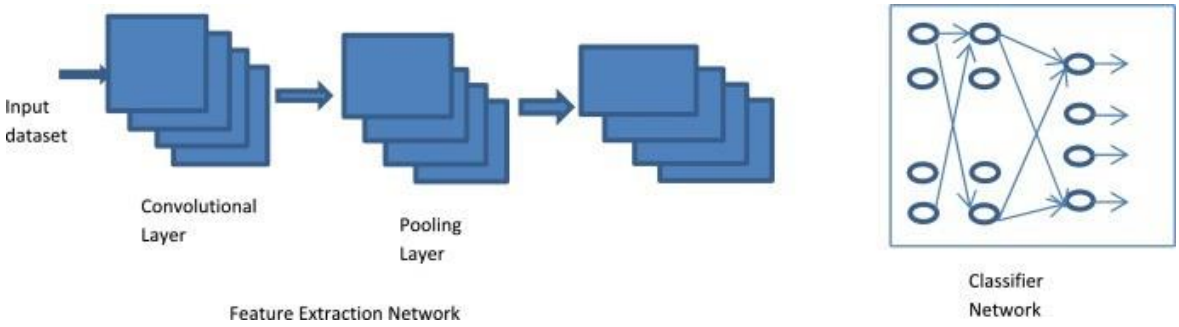

Figure 6: Feature Extraction Network

X. IMAGE CLASSIFICATION

From the database, the training images are labeled with different names. Each of the images is converted to a binary feature vector. These sets of featured vectors are used to train Convolutional Neural Network (CNN) multiclass image category classifier through a supervised learning algorithm, which models a function through analyzing the training images and recognize the new set of test images, with some allied confidence. For seeking better performance, multiple classifiers have been trained.

XI. TESTING

Testing is a process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner.

Software testing is an important element of the software quality assurance and represents the ultimate review of specification, design and coding. The increasing feasibility of software as a system and the cost associated with the software failures are motivated forces for well planned testing.

'Beta testing' is carried out and the requirements traceability are:

1. Match requirements to test cases.
2. Every requirement has to be cleared by at least one test case.
3. Display in a matrix of requirements vs. test cases.

Table 1 Verification of Test Cases

Sl. No.	Test case	Input description	Expected output	Test status
1	Loading model	Initializing trained model and loading it	Loaded model without errors	Pass
2	Converting video to frames	Capturing video and converting it into frames	Image frames of captured video stream	Pass
3	Recognize hand gesture	Image frame that contains hand object	Label	Pass

XII. VALIDATION

K-fold cross-validation can be performed in the model, where the main training data set is partitioned randomly into k discrete sub-samples of nearly equal size. Among the subsamples, (k-1) sub-samples are used for training and the remaining set works as validation sample. Features are extracted from the new predicting images and are encoded as new features. These encoded new features are compared with the trained vocabulary of the classifier. The output confusion matrix shows the accuracy of the prediction.

XIII. SOFTWARE AND HARDWARE REQUIREMENTS

Operating System: Windows, Mac, Linux
SDK: OpenCV, TensorFlow, Keras, NumPy
The hardware interfaces required are:
Camera: Good quality, 3MP
Ram: Minimum 8GB or higher
GPU: 4GB dedicated Processor: Intel Pentium 4 or higher
HDD: 10GB or higher
Monitor: 15” or 17” colour monitor
Mouse: Scroll or Optical Mouse or Touch Pad
Keyboard: Standard 110 keys keyboard

XIV. OUTPUT

The ASL signs shown within the bounding box are captured by the webcam and accordingly the output is shown on the screen. In addition to it, audio output is also obtained. Fig.7 shows the output screen for ASL number 2. It can be seen that the output ’2’ is obtained on the screen.

Figure 7: Output screen for ASL number 2

Fig.8 shows the output screen for ASL alphabet Q. It can be seen that the textual output ’Q’ is obtained on the screen.

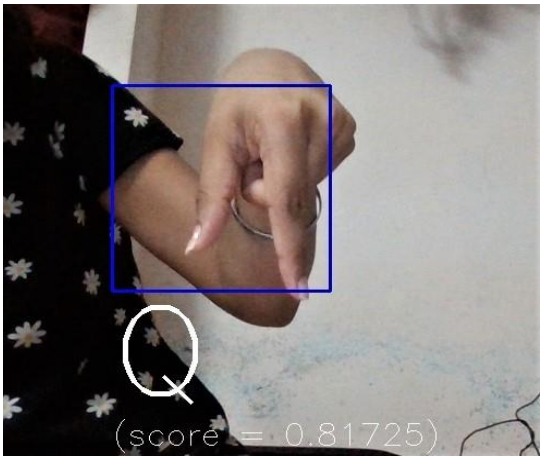

Figure 8: Output screen for ASL alphabet Q

Fig.9 shows the output screen for ASL word HOUSE. It can be seen that the textual output 'HOUSE' is obtained on the screen.

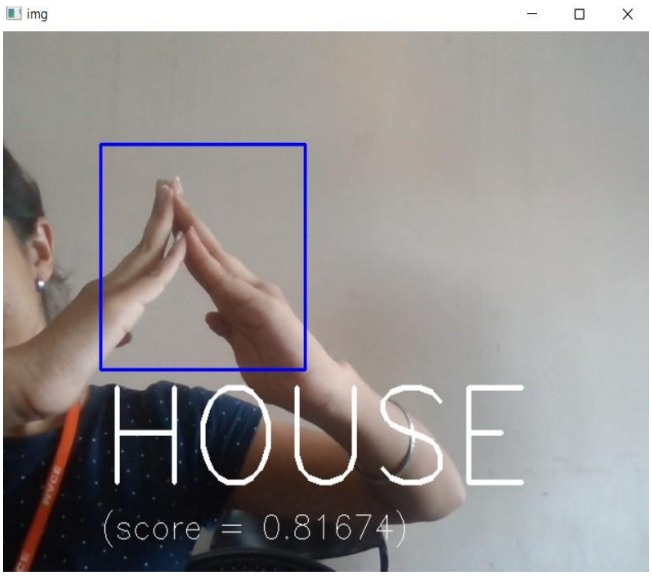

Figure 9: Output screen for ASL word HOUSE

XV. Conclusion

The sign language recognition model can eliminate the obstacle that the hearing impaired people face while communicating with others. Database creation, feature extraction, Classifier Training and Validation of the results are the prime steps of the design. The model does the real-time conversion from hand gesture to the alphabetical letter using webcam and Visual Studio Code. A human-computer interface is formed using the real-time hand gesture. It can successfully convert ASL to alphabetical letters with an average accuracy of 85.2% in real time. The vigorous model is highly accessible and reliable. It can make the user become completely free from the perturbation associated with the glove based method. The user has to render his gesticulations in front of the webcam. The Python code does the processing over the gesture and converts it to the corresponding letter. So, it is quite simple, practical and non-troublesome. Such a sign language recognition model will be definitely a blessing for hearing impaired people.

References

[1] Dipalee Golekar, Ravindra Bula, Rutuja Hole, Sidheshwar Katare and Sonali Parab, "Sign Language Recognition using Python and OpenCV," in Interna- tional Research Journal of Modernization in Engineering Technology and Science, Vol. 4, February 2022

[2] Murthy G.R.S. and Jadon R.S., "Hand gesture recognition using neural networks, in IEEE Advance Computing Conference (IACC), 2010

[3] Pramada S., Saylee D., Pranita N., Samiksh N. and Vaidya M.S., "Intelligent sign language recognition using image processing," in IOSR Journal of Engineer- ing (IOSRJEN), 2013

[4] Pavlovic V.I., Sharma R. and Huang T.S., "Visual interpretation of hand ges- tures for human-computer interaction: A review," in IEEE Transactions on Pattern Analysis & Machine Intelligence, 1997

[5] Ramamoorthy A., Vaswani N., Chaudhury S. and Banerjee S, "Recognition of dynamic hand gestures," in Pattern Recognition, 2003

[6] Liang R.H. and Ouhyoung M.A., "Realtime Continuous Alphabetic Sign Language to Speech Conversion VR System," in Computer Graphics Forum, 1995

[7] Ranganath S.,"Real-time gesture recognition system and application,"in Image and Vision Computing, 2002

[8] Rautaray S.S. and Agrawal A, "Real time hand gesture recognition system for dy- namic applications," in International Journal of Ubiquitous Computing, 2012

[9] A. S. Nikam and A. G. Ambekar, "Sign language recognition using image based hand gesture recognition techniques," in Online International Conference on Green Engineering and Technologies (IC-GET), 2016

[10] A. Kumar, K. Thankachan and M. M. Dominic, "Sign language recognition," in 3rd International Conference on Recent Advances in Information Technology (RAIT), 2016