International Journal of Scientific Research & Engineering Trends
Volume 6, Issue 3, May-June-2020, ISSN (Online): 2395-566X

Portable Sign Language Translator for Emergency Response Teams

Mannava Vivek
IV-year B.E., Department of Mechatronics
SCSVMV (deemed to be university)
Enathur, Kanchipuram, Tamilnadu.
vivekmannava@gmail.com

Vitapu Gnanasagar
IV-year B.E., Department of Electronics and Instrumentation
SCSVMV (deemed to be university)
Enathur, Kanchipuram, Tamilnadu.
vitapugnanasagar@gmail.com

Abstract – **This paper details the development of a system that uses image classification models to identify the hand signs made by speech disabled persons. It allows for easier communication between the dumb/deaf person and others. The system utilizes deep learning in a portable device that helps emergency personnel understand the sign language made by the speech-disabled person without having to wait for an interpreter. As it operates on a database of sign language datasets, it can easily be trained to translate any sign language local to the region.**

Keywords – **Sign language recognition, deep learning, image classification, Raspberry pi.**

I. INTRODUCTION

Sign language is the use of hand gestures and facial expressions to convey one's feelings and thoughts to those around him. While it is a prominent means of communication among people with speech and hearing impairment, others, being able to talk and hear, seldom have use of sign languages. Also, authorities estimate that there are approximately 300 different types of sign languages in use across the globe, with new sign languages evolving among groups of deaf children and adults. These two factors cause a gap in communication between impaired and non-impaired people.

While normal circumstances present no difficulty, certain special situations when a deaf person is to communicate his/her grievance with a police official or an emergency response personnel, unable to communicate freely, is a definite hurdle. Research made by various prestigious institutes and private organizations has made systems available that are capable of recognizing the hand gestures and converting them into standard text or speech. Still, these systems have sophisticated hardware requirements and are not much use in day-to-day life. These high-end systems use one of two methods available for sign recognition. The methods are

1. Sensor-based
2. Image-based

A sensor-based system requires the use of gloves or other attachments that contain sensors to monitor the motion of the fingers, which can then be analyzed to identify the sign made. This method is cumbersome and requires complex devices to monitor every movement. The use of these devices makes the system more expensive, and it also requires frequent calibration for efficient operation.

The alternative is the image-based recognition system. The systems that are currently in development require various lighting and image capturing apparatus to be set up before it can operate. This, too, makes the system expensive and requires much maintenance.

A portable system capable of converting the signs made by the impaired person into human or machine decipherable text, with negligible cost, and universal operability, would go a long way in facilitating accessible communication.

This project is one such attempt at providing a capable device that can translate sign language into standard languages while being universally portable. It is designed to use a raspberry pi controller and deep learning algorithms to convert the signs into standard languages.

II. SIGN LANGUAGE DATABASE

As mentioned earlier, sign language is a collection of different hand signs. However, it is an independent language, different from spoken/written languages. Each sign language has its own set of signs for alphabets, numerals, and words/phrases, or even sentences. For better outreach and accessibility, various databases have been established to provide information on the signs used in each of the different languages.

For this work, we have chosen ASL (American Sign Language) offered by American Sign Language Linguistic research project.

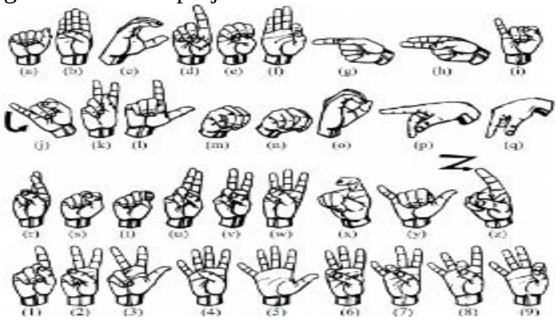

Fig.1. ASL gestures.

International Journal of Scientific Research & Engineering Trends
Volume 6, Issue 3, May-June-2020, ISSN (Online): 2395-566X

III. DATA PRE - PROCESSING

The images gathered from the ASLLRP database include shots of various people performing the signs of various characters and phrases of the American sign Language. Image pre-processing includes steps for removing unwanted noise, adjusting the brightness and contrast of the image, and cropping the image as per requirement.

1. Removing Unwanted Noise

Various tools are popular in the market that is capable of eliminating unwanted noise from an image. These tools are utilized to filter the noise and obtain better image quality. Reduced noise also improves the accuracy of prediction, after the model is trained.

2. Adjusting the brightness and contrast

The same tools available for image noise reduction also allow you to vary the brightness, contrast, and other variables of a photo. The optimum values of these parameters are set to identify the skin area of the hand quickly.

3. Cropping the Image

The RGB image frames are converted into HSV images to help in human color perception. They differentiate the image into spaces based on hue, saturation, and value. This makes it easier to identify the hand object and specify the object boundaries. The region within the set value is isolated to identify the hand object.

IV. DATA PROCESSING

The image in which the hand is isolated is further analyzed using thresholding techniques. Thresholding is a simple, yet effective method of partitioning an image into foreground and background. The entire image is divided into pixels, and for each pixel above a set value, the pixel is set a 1 (white), while areas below a specific value are considered background, and the pixels are set to 0 (black). This effectively converts the HSV into a binary image, with the region containing the hand object in white, and the rest of the image in black. Copies of the binary images are made in various orientations to increase the data available for training the deep learning model.

Training a deep learning model is essentially tuning its parameters so that you can map a particular input to an output label. The more data available for training the model, the better the accuracy of the final model. Hence, multiple variations of each image are used to fine-tune the operation of the model.

The more the number of parameters, the more the number of examples required to train an efficient model. By accounting for orientation and other parameters ourselves, we reduce the model's loss and ensuring the model is trained in the right way.

After the images in multiple orientations are obtained, they are separated into individual datasets, depending on the character or phrase they represent. These datasets will be utilized in training the model.

V. TRAINING A MODEL

A deep learning model is a collection of deep neural networks, which the computer uses to train the model. A deep neural network contains an input layer, an output layer, and multiple hidden layers. The binary images from the pre-processing stage are fed to the TensorFlow module to train the model. Each image is sent through the multiple layers present in the deep neural network, effectively training the model on the different parameters.

Training the model also includes generating the meta graph file and checkpoint file. TensorFlow also allows you to generate the precision and recall values, the f-scores in the classification report, and the confusion matrix to verify your model's operation.

```
Length of images_labels 98400
Length of train_images 82000
Length of train_labels 82000
Length of test_images 8200
Length of test_labels 8200
Length of test_images 8200
Length of val_labels 8200
```

Fig. 2. Log Data.

Classification Report

	precision	recall	f1-score	support
0	1.00	1.00	1.00	188
1	1.00	1.00	1.00	200
2	1.00	0.99	1.00	183
3	1.00	1.00	1.00	201
4	0.99	1.00	1.00	199
5	1.00	0.99	0.99	189
6	1.00	1.00	1.00	206
7	1.00	1.00	1.00	202
8	1.00	1.00	1.00	197
9	1.00	1.00	1.00	220
10	1.00	1.00	1.00	197
11	1.00	1.00	1.00	221
12	1.00	1.00	1.00	210
13	1.00	1.00	1.00	219
14	1.00	1.00	1.00	207
15	1.00	1.00	1.00	197
16	1.00	1.00	1.00	194
17	1.00	1.00	1.00	192
18	1.00	1.00	1.00	200
19	1.00	1.00	1.00	201
20	1.00	1.00	1.00	209
21	1.00	1.00	1.00	193
22	1.00	1.00	1.00	206
23	1.00	1.00	1.00	179
24	1.00	1.00	1.00	228
25	1.00	1.00	1.00	211
26	1.00	1.00	1.00	189
27	1.00	1.00	1.00	212
28	1.00	1.00	1.00	193
29	1.00	1.00	1.00	193
30	1.00	1.00	1.00	187
31	1.00	1.00	1.00	194
32	1.00	1.00	1.00	191
33	1.00	1.00	1.00	188
34	1.00	1.00	1.00	203
35	0.99	1.00	1.00	191
36	1.00	1.00	1.00	206
37	1.00	1.00	1.00	205
38	1.00	1.00	1.00	210
39	1.00	1.00	1.00	188
40	1.00	1.00	1.00	201
accuracy			1.00	8200
macro avg	1.00	1.00	1.00	8200
weighted avg	1.00	1.00	1.00	8200

Fig. 3. Classification report.

The classification report and the confusion matrix are used to check the performance of the classification model on a set of test data, in this case a set of images of hand signs, whose true values are already known.

International Journal of Scientific Research & Engineering Trends
Volume 6, Issue 3, May-June-2020, ISSN (Online): 2395-566X

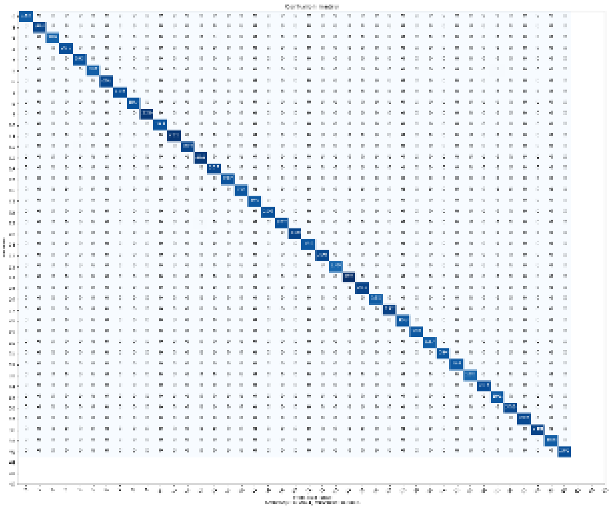

Fig. 4. Confusion matrix.

VI. RASPBERRY PI

Raspberry Pi is a single-board computer developed by the Raspberry Pi foundation, UK. It can plug into a computer monitor or an LCD screen, and uses a standard keyboard and mouse to provide computer-like capabilities. For this project, we used a Raspberry pi 3 model B+ that comes with a 1.4GHz 64-bit quad-core processor with Raspbian as the default operating system. However, you can install other Linux based operating systems like Ubuntu MATE or Gentoo Linux. You can also run it windows IoT core. Raspberry Pi has the capability to run scripting languages like python, thus allowing it run python libraries like TensorFlow, albeit at limited computational capacity. The TensorFlow library for Raspberry Pi is called TensorFlow Lite, and it requires some changes to be made to the original TensorFlow model before it can be run on the Raspberry pi. Other dependencies are similar to the ones for a normal computer.

The trained model can be exported to multiple devices that have lower computational capabilities than the system in which the model was trained. Since our application uses a Raspberry Pi, the model has to be converted in the TensorFlow Lite format. TensorFlow Lite is designed to execute models efficiently on mobile and embedded systems with limited computational and memory capabilities. Some of this efficiency comes from a special format in storing the trained models. Converting a TensorFlow model into TensorFlow Lite format reduces the file size and introduces optimizations that do not affect the accuracy of the model. The conversion is done by TensorFlow lite converter, a tool available as Python API that converts trained TensorFlow models. The TensorFlow lite converter makes it easy to quantize TensorFlow models by reducing the precision of values and operations within a model, like changing values from full floating points to half-precision points (16-bit) or 8-bit integers.

VII. OPERATION

The Raspberry Pi device containing the trained and converted model is connected with a Raspberry Pi camera v2 and a touch screen display, and the drivers are installed. Executing the TensorFlow Lite model will make this setup a portable Sign language translator. The user will have the options to calibrate the system to the signaling person's skin tone and the type of sign language used. Once calibrated, the signs made by the deaf person is recorded using the Raspberry Pi camera. The video is then split into frames and each frame containing a sign is converted into a binary image using the thresholding technique. The data from the binary image is passed through dense and flatten layers, and max_pooling layers, before being compared with the training data. The output label that presents the highest accuracy is displayed on the LCD screen, thus successfully converting sign languages into standard text-based languages.

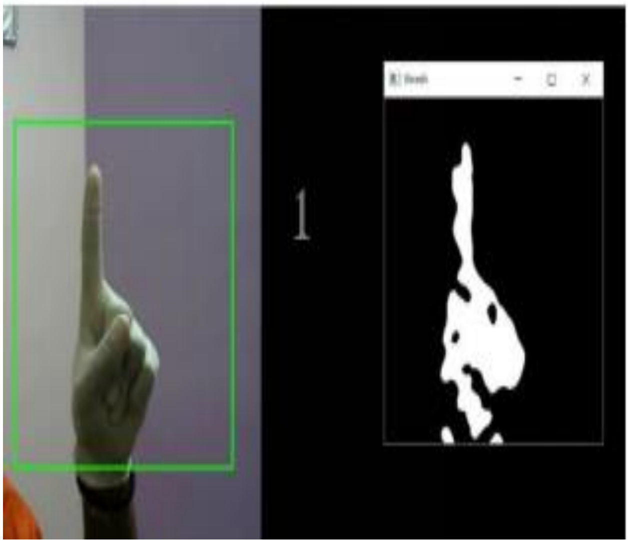

Fig. 5. Recognizing number 1.

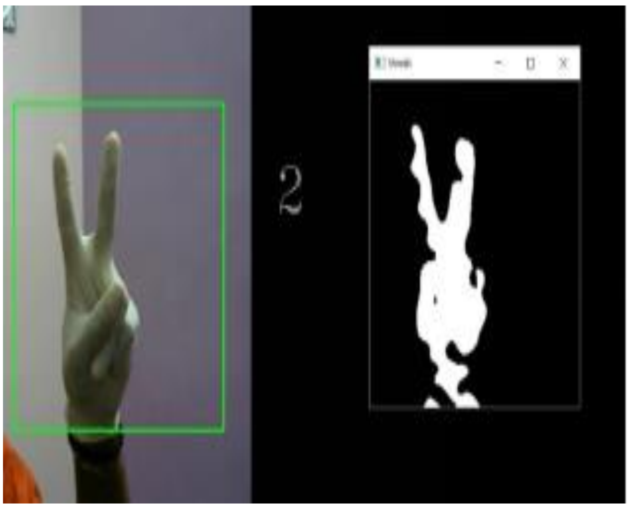

Fig. 6. Recognizing number 2.

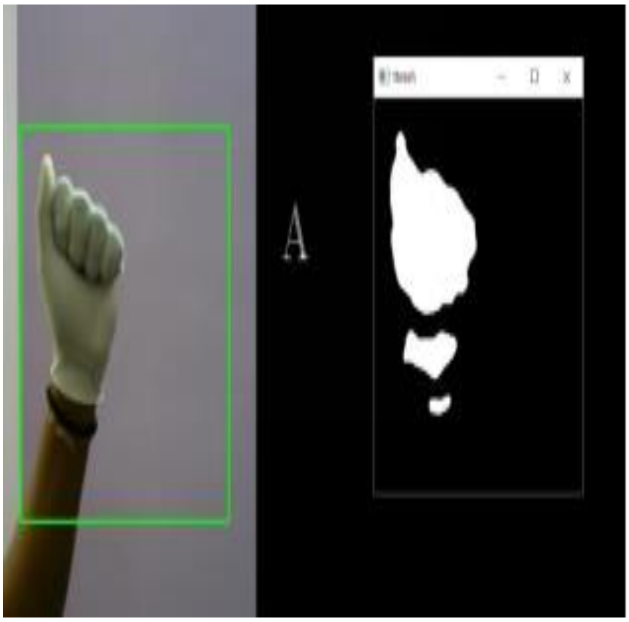

Fig. 7. Recognizing letter A when image is unclear.

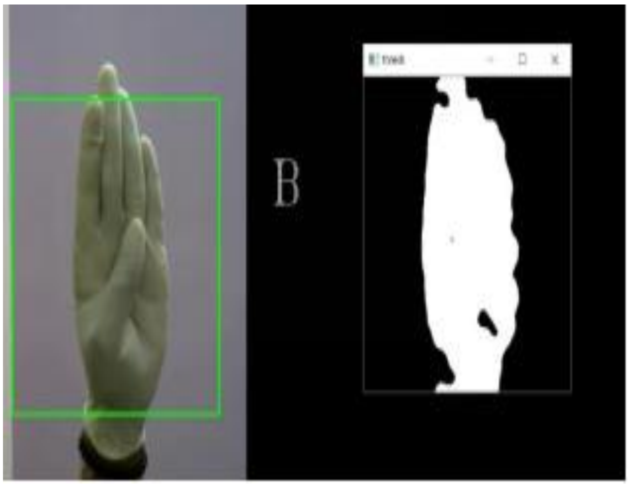

Fig. 8. Recognizing letter B when hand is out of bounds.

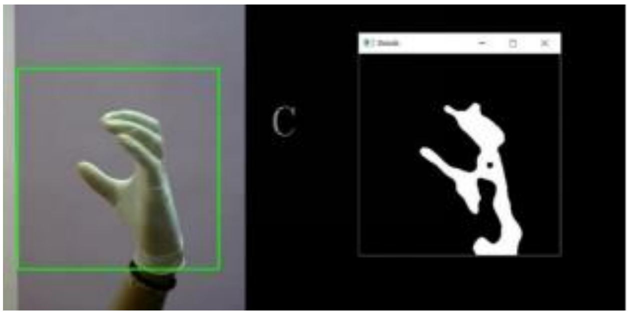

Fig. 9. Recognizing letter C when sign is unorthodox.

VIII. CONCLUSION

This system, while having limited capabilities and is not very suitable for high level image processing applications, it will be sufficient to ensure proper communication between the speech-impaired person and the emergency or police personnel, enabling them to express their grievances or complaints without having to resort to acting out their message.

This system is not restricted for use by the emergency services, but also in general day-to-day life, and widespread use of this system, will be a huge step forward in making life normal for those with the impairment.

REFERENCES

[1]. S. Nikam and A. G. Ambekar, "Sign language recognition using image based hand gesture recognition techniques," 2016 Online International Conference on Green Engineering and Technologies (IC-GET), Coimbatore, 2016, pp. 1-5, doi: 10.1109/GET.2016.7916786.

[2]. Kour, Kamal & Mathew, Lini. (2017). Sign Language Recognition Using Image Processing. International Journal of Advanced Research in Computer Science and Software Engineering. 7. 142. 10.23956/ijarcsse.v7i8.41.

[3]. Ss, Shivashankara & S, Dr.Srinath. (2018). American Sign Language Recognition System: An Optimal Approach. International Journal of Image, Graphics and Signal Processing. 10. 10.5815/ijigsp.2018.08.03.

[4]. Gautam, Amit Kumar. "American Sign Language Recognition System Using Image Processing Method." (2017).

[5]. Necati Cihan Camgoz, Oscar Koller, Simon Hadfield, Richard Bowden, "Sign Language Transformers: Joint End-to-end Sign Language Recognition and translation", IEEE Conference on Computer Vision and Patter recognition, 2010.

[6]. Dan C. Cires¸an, Ueli Meier, Jonathan Masci, Luca M. Gambardella, Jurgen Schmidhuber, "Flexible, High Performance Convolutional Neural Networks for Image Classification", Proceedings of Twenty Second International Joint Conference on Artificial intelligence, 2011.

[7]. krishna, M & Neelima, M & Mane, Harshali & Matcha, Venu. (2018). Image classification using Deep learning. International Journal of Engineering & Technology. 7. 614. 10.14419/ijet.v7i2.7.10892.

[8]. S. He, "Research of a Sign Language Translation System Based on Deep Learning," 2019 International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM), Dublin, Ireland, 2019, pp. 392-396, doi: 10.1109/AIAM48774.2019.00083.

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

[10]. H. Muthu Mariappan and V. Gomathi, "Real-Time Recognition of Indian Sign Language," 2019 International Conference on Computational Intelligence in Data Science (ICCIDS), Chennai, India, 2019, pp. 1-6, doi: 10.1109/ICCIDS.2019.8862125.
[11]. Gay, Warren. (2014). Raspberry Pi Hardware Reference. 10.1007/978-1-4842-0799-4.
[12]. M. Elmahgiubi, M. Ennajar, N. Drawil and M. S. Elbuni, "Sign language translator and gesture recognition," 2015 Global Summit on Computer & Information Technology (GSCIT), Sousse, 2015, pp. 1-6, doi: 10.1109/GSCIT.2015.7353332.
[13]. Carol Neidle and Christian Vogler [2012] "A New Web Interface to Facilitate Access to Corpora: Development of the ASLLRP Data Access Interface," Proceedings of the 5th Workshop on the Representation and Processing of Sign Languages: Interactions between Corpus and Lexicon, LREC 2012, Istanbul, Turkey.
[14]. The National Center for Sign Language and Gesture Resources (NCSLGR) Corpus.
[15]. http://www.bu.edu/asllrp/
[16]. http://secrets.rutgers.edu/dai/queryPages/