Proceedings of the Second International Conference on Inventive Research in Computing Applications (ICIRCA-2020)
IEEE Xplore Part Number: CFP20N67-ART; ISBN: 978-1-7281-5374-2

Implementation of Virtual Assistant with Sign Language using Deep Learning and TensorFlow

Dipanshu Someshwar
Student, Department of Information Technology, Shah and Anchor Kutchhi Engineering College, Chembur,

Mumbai- 400088, India
Dipanshu.someshwar@sakec.ac.in

Dharmik Bhanushali
Student, Department of Information Technology, Shah and Anchor Kutchhi Engineering College, Chembur,

Mumbai- 400088, India
dharmik.bhanushali@sakec.ac.in

Vismay Chaudhari
Student, Department of Information Technology, Shah and Anchor Kutchhi Engineering College, Chembur,

Mumbai- 400088, India
vismay.chaudhari@sakec.ac.in

Swati Nadkarni
Associate Professor,
Department of Information Technology, Shah and Anchor Kutchhi Engineering College, Chembur,

Mumbai- 400088, India
swati.nadkarni@sakec.ac.in

***Abstract*— The paper is all about the system and interface developed, that allows deaf mutes to make use of various voice automated virtual assistants with help of Sign Language. Majority of Virtual Assistants work on basis of audio inputs and produces audio outputs which in turn makes it impossible to be used by people with hearing and speaking disabilities. The project makes various voice controlled virtual assistants respond to hand gestures and also produces results in form of text outputs. It makes use of concepts like Deep Learning, Convolutional Neural Network, Tensor Flow, Python Audio Modules. A webcam first captures the hand gestures, then Convolutional Neural Network interprets the images produced and produces rational languages. These languages are then mapped to pre-defined datasets using Deep learning. For this purpose, Neural Networks are linked with Tensor flow library. The designed system will then produce audio input for the Digital Assistant, using one of the Python text to speech module. The final audio output of the Digital Assistant will be converted into text format using one of the Python speech to text module which will be displayed on the viewing screen.**

Keywords*— *Deep Learning, Virtual Assistants, Tensor Flow, Convolutional Neural Network Hand Gestures, Sign Languages.

I. INTRODUCTION

Nowadays, Virtual Assistant devices have been part and parcel of our lives, but most of them are Voice Automated. Most commonly used Virtual Assistants are Alexa, Google Home, Apple Siri and Microsoft Cortana. These assistants listen to user's queries and respond accordingly making there life easier, thus they have been a very important part of Home Automation. Since these assistants are purely Voice Automated, Deaf-Mutes find it hard to make use of such technology as observed in [8]. The agenda of the project is to develop an interface that will help the Deaf-mutes to use these Virtual Assistants easily with easy. As of now, it might seem irrelevant to design such a system but in a longer run it might help deaf-mutes to equally enjoy their social and personal life. Designing such an interface will make them find their freedom while using such technologies and might boost their confidence in this Digital Age. This paper focuses on a research that gives an idea of combining two modern technologies that are Hand Gesture Recognition and Virtual Voice Assistants in order to make it possible for people with hearing/speaking difficulties to interact with Digital Gadgets and also communicate with the outside world. This research work has implemented Alexa which is an audio based Virtual Assistant. The proposed system has been successful in replacing Speech Recognition technique with Hand Gesture Recognition technique. The proposed system makes use of following technologies: TensorFlow which is the most important library used for designing and developing the model of these system, Convolutional Neural Network, is an Deep Learning Algorithm that have been used for serving the purpose of Image Recognition, that helps in converting the images in form of matrix that can be understood by the model and making it Classifier ready, and lastly OpenCV that will act as an Eye of the system that will capture and process Real-time Hand Gestures and predict results with help of Classifier.

With increasing trends in technology, personal assistant devices are becoming more and more popular. But such devices are voice automated. They need audio inputs and provide audio outputs. So what if someone does not have their own voice or are not in a condition to speak properly, that's where this project comes into light. Such people can easily communicate with these devices using an interface that takes hand gestures as an input and provides audio as well as text output. This project has the capacity to bridge the gap between such impaired people and booming technology.

II. LITERATURE REVIEW

Every existing Virtual Assistant in today's date is found to be Voice Automated thereby making it unusable by Deaf-mutes and people with certain disabilities. This leads to the need of a system which can help people with speaking or listening disabilities to make use of such Virtual Personal Assistants [8]. Artificial Neural Network is used in majority cases where static recognition is performed as shown in [1], but there are few drawbacks related to the efficiency of recognizing distinctive features from images which can be improved by using Convolutional Neural Network. Convolutional Neural Network when compared to its predecessors, recognizes important distinctive features more efficiently and without any human supervision. Artificial Neural Network uses one-to-one mapping which increases the number of nodes required thereby degrading the efficiency whereas Convolutional Neural Network uses one-to-many, keeping the number of nodes low and greatly improving the efficiency [5]. Many systems designed with such objectives tend to make use of more of physical hardware like the design observed in Cyber Glove thereby leading to need of manufacturing of such hardware gadgets and also making it mandatory for the users to wear it while accessing the Virtual Assistants [11]. Many systems are designed in such a way that there application is limited to only certain Sign language or series of Hand gestures [9] whereas the proposed system is designed in such a way that it gives us the flexibility of changing to any standard sign language just by changing the dataset and training the model for the same.

III. METHODOLOGY

The most basic explanation of workflow of the system goes as follows - A hand gesture is performed in front of the webcam just as observed in [10]. This sign gesture is converted to text and the text output is converted to audio and is served as an input to the assistant. The assistant processes the question and responds in audio format. This audio format is converted to text output. The text output will be then displayed on the display screen.

It has been understood by using a block diagram shown in Fig.1

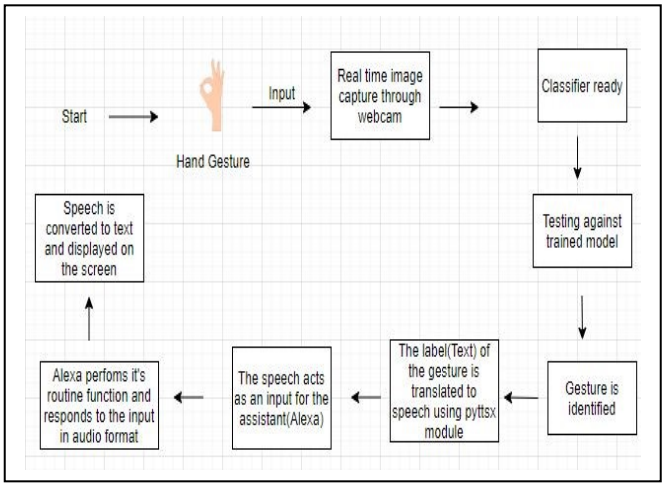

Fig. 1. Implemented System Workflow

Now to understand the technical workflow of the complete system, a brief explanation goes as follows - The very first step is to provide training dataset and train the system with a variety of hand gestures named with their respective labels. This is the most time consuming step. Better the underlying system software specifications, lesser the time required for training will be. Once training is complete, the next phase is prediction mode. It now uses the input image from a webcam and runs it through the classifier to find its closest neighbours based on the training examples and labels provided in the previous step. If a certain prediction threshold is crossed, it will append the label on frame as suggesting that system recognised the following hand gesture accordingly. Then Python Text-to-Speech module is used for speech synthesis to speak out the detected label to the digital assistant, after collecting series of hand gestures and execute it on presence of certain call commands which will be declared previously. If the spoken word is 'Alexa' it causes the nearby Echo to awaken and begin listening for a query and then Alexa responds to query in voice format. Meanwhile system will start the Voice to Text python module which will listen to the Alexa responded query and then convert it into text format and display it in output frame on the Display Screen. The entire process can be repeated multiple number of times. But the hand gestures will be only recognised if they meet the trained data set standards.

It can be understood by using a Flow chart shown in Fig.2

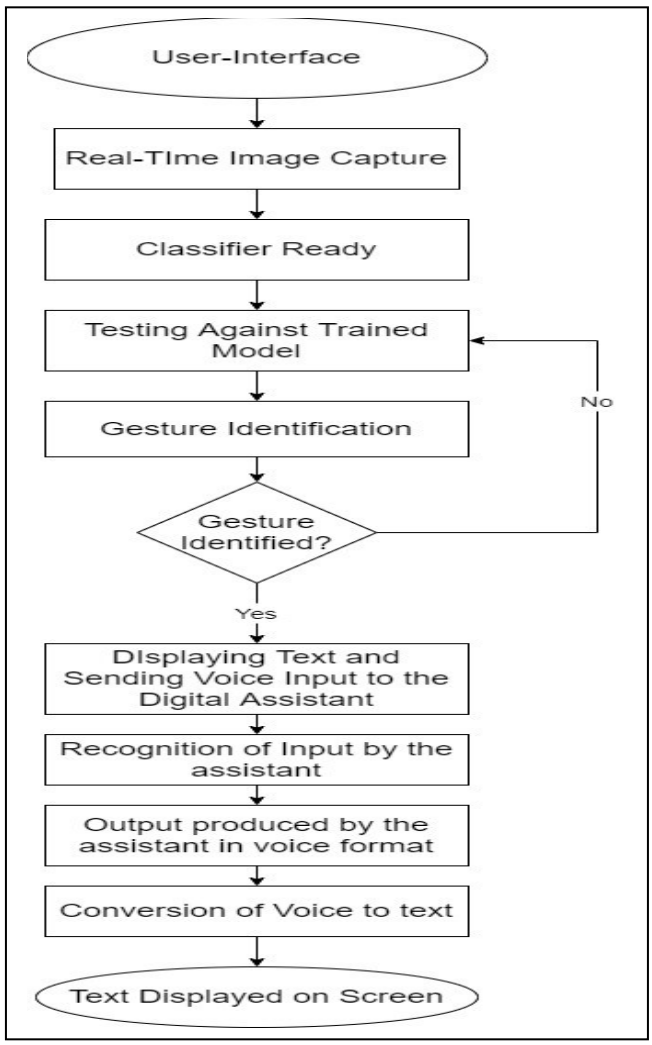

Fig. 2. Flow of the system

A. Training Dataset

Dataset is the most fundamental element of any Machine Learning Model. As it is a process of feeding into Machine's memory to help classify whatever it insights in future for the designed application. Since our system is an interface for Real-Time Classification of Hand Gestures our Dataset will purely consist of large number of Images in form of .jpeg, .jpg, these are the only two extensions that out model is accepting. The designed model makes use of a Labelled dataset method for training our system, thus assigning labels to folder names will simply use sub-files of images to be trained under assigned labels. Each label is being trained with about more than 2000 images captured at various possible angles in order to make system learn better and classify more accurately and quicker as observed in [2]. Once the model is completely trained for a set of particular labelled images it gets Classifier ready and can be used for testing the system's prediction rate. However it was noticed that retraining the same set of labels tends to give better results in terms of accuracy and speed of predicting the Hand gestures as observed in [3]. Basically the model will be trained more number of times for the same set of labels, higher is the success rate. But it is required to keep a note that any changes made with the labels folder before training will lead to the system that is being trained for the very first time. In simple terms if it is required to make any changes in the labels folder that is adding new labels or replacing the existing labels, the model will need to be trained again from the beginning. It was observed that for training of around 15 labels on an average configured system, it takes about 12-15 hours straight of model training for the first time. However retraining of same set of labels requires comparatively lesser amount of time.

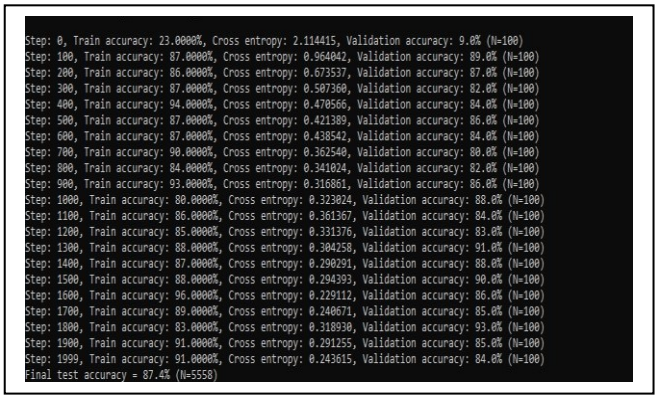

Fig. 3. Testing Accuracy

Fig. 3, is the part of training a dataset that shows the training accuracy obtained at respective number of step, along with the Cross Entropy value and Validation Accuracy for the same. Basically train accuracy is the value which let us understand how well the training is taking place while the Validation Accuracy let us know how the model will react while predicting the data it has not seen before.

B. Tensor Flow

The best part of using TensorFlow library is that it is an open Source Library with lots of pre designed models, useful in Machine Learning and especially Deep Learning. For understanding the conceptual use of Tensor Flow is required to understand the meaning of two terms, where the Tensor here is considered as N-Dimensional Array and Flow refers to graph of operations. Every mathematical computation in TensorFlow is considered as graph of operations where Nodes in the Graph are operations and Edges are nothing but tensors.

Any mathematical computation is written in form of data flow diagram in Python Frontend or C++ or Java, as in our case Python is used. Then, TensorFlow Execution Engine comes into picture and makes it deployable on any of the hardware of Embedded System let it be CPU or Android or IOS. TensorFlow is a Machine learning framework that comprises of uses the dataset to train Deep learning models and helps in prediction and also improvise future results.

The biggest advantage of using TensorFlow is it's feature of providing Abstraction, that is the developer does not need to work on every small aspects of designing the model as it is managed by the library itself, thus giving the developer the

freedom to focus on logic building, which was clearly explained in [7] .

TensorFlow in our system helps us in training the model using the provided dataset. TensorFlow object recognition algorithms helps us classify and identify different hand gestures when combined with use of OpenCV. By analysing thousands of photos, Tensorflow can help classifying and identifying real-time hand gestures. It makes possible to develop a model which can help identify 3D images and classify it on basis of 2D images from its feed dataset. TensorFlow is capable of processing more information and spot more patterns.

C. Deep Learning

Deep Learning is basically a subset of Machine Learning model which consists of algorithms that make use of multi-layer neural networks. Deep Learning makes use of Neural Network most of the times to implement its functioning. A Neural Network is a collection of layers that transforms the input in some way to produce output.

Image can be termed as matrix of pixel values so it may seem that classification can be an easier task simply based on matrix classification but that is not the case with complex matrix images or images with similar forms of matrix or a very huge dataset of images with minimal changes in the matrix. This may lead to clash in prediction scores and thereby affecting the accuracy and speed of classifier model. This is where Neural Network comes into picture and thus it is required to use deep learning over machine learning. Machine Learning works with lesser number of layers when compared with Deep Learning as observed from [12] and thus not preferred for technologies like Image Recognition which requires need of Convolutional Neural Networks.

D. Convolutional Neural Network

A convolutional Neural Network is nothing but a Deep Learning algorithm that is capable of assigning biases and weights to different objects in an Image and on basis of the same it can differentiate one image from another. It consists of processing different layers of Image Classification and it is designed with means of representing functioning of Neurons in Human Brain as explained in [4].

Even if the most minimalist pixelated image is considered, it still needs 4x4 matrix and required to consider the same image in different channels of colour formats like RGB, Greyscale, HSV, etc so it is very difficult to process thousands of images in high rates of pixels for instance 1020x1980 pixels. Here comes the need of Convolutional Neural Network that convolutes every image into its basic reduced form of matrix which can be differentiable at the same time. These increases the Accuracy and Speed and also reducing the processing of Classifier model. The convolutional layer is also supported with Pooling layer to decrease the processing need of classifier model. It also convolutes the matrix but on basis of dominant features. Pooling is majorly of two types; MAX Pooling and AVG Pooling, this is clearly explained in [6].

Inception-v3 Convolutional Neural Network has been implemented, while designing this system. Inception v3 is a 48 layers deep Neural network. Inception Network is better than most of Convolutional Neural Networks because it just does not dig deeper and deeper in the layers like other Convolutional Neural Networks instead it believes in working wider on the same layer before going deeper into the next layer. This is the reason, bottlenecks are used while training the model. Bottleneck in Neural Network is just a layer having less neurons as compared to the layers above or below it. TensorFlow bottleneck is the last step of pre-processing phase that starts before actual training of dataset starts.

E. OpenCV

OpenCV is an open source library for Computer Vision. Now since all the training and classification is ready to be executed when it needed an eye for the designed system to capture real-time images of Hand Gestures which can then be sent for classification and identification. OpenCV adds intelligence to Deep Learning models for visualization image processing. Here images are considered over 2 channels as: RGB Channel and Grey Scale Channel so once the image is captured by OpenCV it first converts into Grey channel so it can then undergo morphological processing as shown in [9]. OpenCV makes use of Numpy Library for numerical computation of Images in form of matrix of pixels.

A blue box of particular dimension has been designed with help of OpenCV in a way that it will consider hand gestures present inside this blue box. It then converts the image over different channels and then convert the image into convoluted form of matrix so the Classifier model can compare it with previously learned labelled images. It will then predict a suggestion of gesture on basis of the score generated. As OpenCV is converting real-time hand gesture it will be continuously suggesting predictions because of slightest of motion of real-time hand gesture. The confirmed prediction with highest score will enter the sequence until a CALL COMAND is executed. Then the entire sequence will enter the next stage of designed interface that is it will be converted into Audio format which will then wake the Virtual Voice Assistant and become the Input Query.

F. Python Text and Speech APIs

The Python text-to-speech library that used is very simple and easy to use. It makes use of modules like pyttsx3 and engine.io which let us change different properties like rate and intervals of text to speech conversion and outflow.

The Python speech-to-text library by which practicing makes use of speech recognition module. It let us adjust the ambient noise and also helps in recording the audio in form of mp4 files.

IV. RESULTS

A short demonstration of our project is given below with help of images. Here, it performed a Hand Gesture by representing a term "Weather" and once CALL COMMAND is received, a query asking weather will be sent to Virtual Voice Assistant and the real-time output will be converted into text and displayed on output frame.

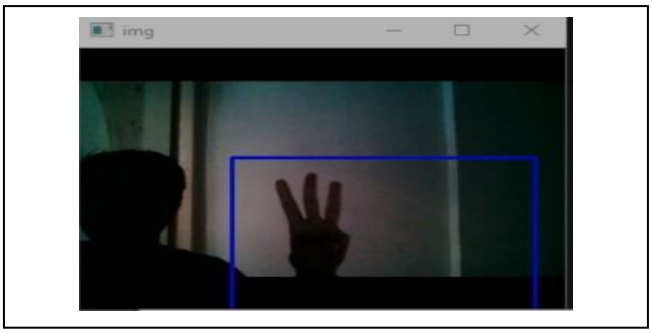

Fig. 4. Hand Gesture for term Weather

Fig. 4, shows how the designed system will capture real-time hand gestures, only the ones inside blue box, using Web Camera. In the above figure person is performing a hand gesture representing word 'Weather'.

Fig. 5. Sequence frame

Fig. 5, represents a window frame which will be displaying system's suggestions for captured hand gesture.

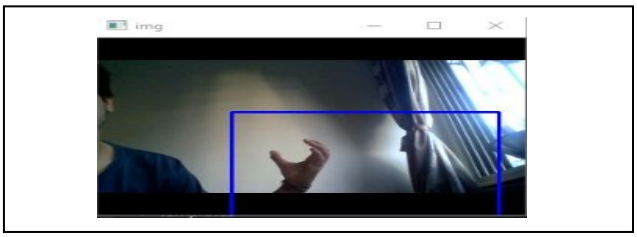

Fig. 6. Pre-Defined Call Command

Fig 6, represents the system capturing a real-time hand gesture for CALL Command, so that the previously captured words in the sequence frame can now be converted into Audio format as an input for Assistant.

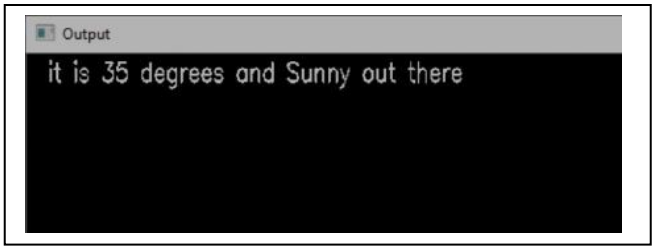

Fig. 7. Output Frame

Fig. 7, shows the output frame which displays the real-time response generated by Voice Assistant, but converted in text format and displayed on the screen.

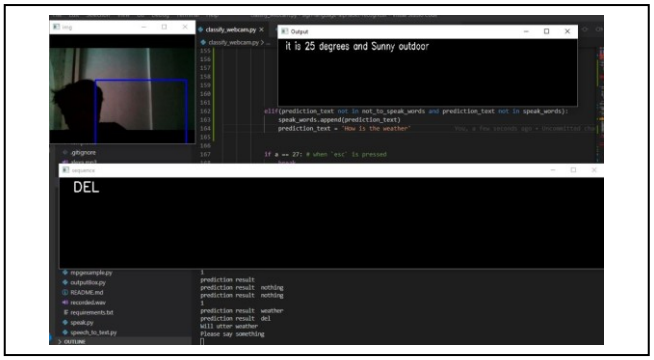

Fig. 8. Interface

Fig. 8, is a view of real-time interface of the designed system with all the window frames tied up together, performing their expected roles. Now, again if no gesture is being performed the sequence window will print DEL, which is also another CALL Command.

Fig. 9. Classifier's Accuracy

Fig. 9, shows us the score which indicates the accuracy by which the gesture is being classified by the model. It ranges between 0 to 1 as shown above.

V. CONCLUSION

The designed system was successfully able to capture Hand Gestures using the integrated Web Camera and process and convert into text format and display it onto the Input frame and then converted into Audio format on receiving a CALL COMMAND. The audio becomes a query for the Virtual Assistant and again the audio output was being successfully converted into Text format and displayed on the screen as shown above in results. Most of the times in the preferred

conditions, the system was able to provide the accurate and best of its results. However sometimes in poor light conditions and in absence of proper background the system struggled to produce correct and expected results.

While developing this system, many difficulties are faced like in recent updates these virtual assistants have stopped responding to digital voices of certain frequency and bandwidth, in order to not wake these devices in cases of commercial advertisements (like in case of ALEXA) so it is required to correctly select the Digital Voice, which is used in our Python text-to-speech library to avoid this issue.

Also as the designed system is trained completely using the chosen Dataset, thus selection of Dataset should be done on basis of the Standard Sign Language used by the deaf-mutes in targeted region as these languages tend to change locally and globally.

Basically the system can be considered as a boon to people with hearing disabilities or speaking disabilities or both at the same time. These system would not only bring technology into their Personal lives but also give rise of opportunities in their professional life.

VI. FUTURE SCOPE

The current designed system completely works on basis of underlying Dataset which is used to train the system, thus making its use limited to certain group of people which communicates using the similar Sign Language. However it is found out that there are various forms of Sign Languages globally, so dataset used needs to be changed according to the standard Sign Languages used in that nation or region.

Since the current system's Graphical User Interface is simple but not visually soothing, further it is aimed at designing and building a better Visual Interface that makes it even more vibrant and eye-catching eventually making it easier as well interesting to use, for our targeted audience.

The existing Virtual Voice Assistants are basically a form of Smart Speakers which are Voice Automated. Our designed interface is an add-on to make them accessible by deaf-mutes with help of a Web Camera and a laptop. So if this system gets a good response by the users, it can even take it to a larger scope by being able to integrate our system completely with those Virtual Assistant speakers in a better form where the speakers itself will consist of a Web Camera, which will act as an eye and respond to Hand Gestures, if combined with processing abilities.

As the current system can still not be called a complete Error Free product, it still needs to make it better by means of it's Overall Accuracy and Productivity in terms of generating end results. Our system has few limitations such as need of a plain background, white being the most favourable for better results, also presence of good amount of light while presenting the hand gestures. So it is required to overcome these difficulties in order to make system perform better.

ACKNOWLEDGMENT

We wish to express our profound gratitude to our Principal Dr. Bhavesh Patel and our project guide Ms. Swati Nadkarni for allowing us to go ahead with this project and giving us the opportunity to explore this domain. We would also like to thank the Review Committee for their invaluable suggestions, constant encouragement and support towards achieving this goal. Finally, we would also like to thank Mumbai University for believing in our project's scope and providing us the Grant for the requirements of the Project.

REFERENCES

[1] Yusnita, L., Rosalina, R., Roestam, R. and Wahyu, R., 2017. Implementation of Real-Time Static Hand Gesture Recognition Using Artificial Neural Network. *CommIT (Communication and Information Technology) Journal*, 11(2), p.85.

[2] Rathi, P., Kuwar Gupta, R., Agarwal, S. and Shukla, A., 2020. Sign Language Recognition Using ResNet50 Deep Neural Network Architecture. *SSRN Electronic Journal*

[3] V. Adithya, P. R. Vinod and U. Gopalakrishnan, "Artificial neural network based method for Indian sign language recognition," *2013 IEEE Conference on Information & Communication Technologies*, Thuckalay, Tamil Nadu, India, 2013, pp. 1080-1085.

[4] Guru99.com. 2020. *Tensorflow Image Classification: CNN(Convolutional Neural Network)*. [online] Available at: <https://www.guru99.com/convnet-tensorflow-image-classification.html>.

[5] Guo, T., Dong, J., Li, H. and Gao, Y., 2017. Simple Convolutional Neural Network on Image Classification. *IEEE 2nd International Conference on Big Data Analytics*, pp.1-2.

[6] Medium. 2020. *A Comprehensive Guide To Convolutional Neural Networks—The ELI5 Way*. [online] Available at: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>

[7] Medium. 2020. *Deep Learning With Tensorflow: Part 1 — Theory And Setup*. [online] Available at: <https://towardsdatascience.com/deep-learning-with-tensorflow-part-1-b19ce7803428>

[8] Issac, R. and Narayanan, A., 2018. Virtual Personal Assistant. *Journal of Network Communications and Emerging Technologies (JNCET)*, Volume 8(Issue 10, October (2018).

[9] Lai, H. and Lai, H., 2014. Real-Time Dynamic Hand Gesture Recognition. *International Symposium on Computer, Consumer and Control*, pp.658-661.

[10] Pankajakshan, P. and Thilagavathi B, 2015. Sign language recognition system. 2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS).

[11] K. A. Bhaskaran, A. G. Nair, K. D. Ram, K. Ananthanarayanan and H. R. Nandi Vardhan, "Smart gloves for hand gesture recognition: Sign language to speech conversion system," 2016 International Conference on Robotics and Automation for Humanitarian Applications (RAHA), Kollam, 2016, pp. 1-6, doi: 10.1109/RAHA.2016.7931887.

[12] Ertham, F. and Aydin, G., 2017. Data Classification with Deep Learning using Tensorflow. *IEEE 2nd International Conference on Computer Science and Engineering*, pp.757-7