ISSN No. 0976-5697

VOLUME 12 SPECIAL ISSUE 2, JUNE 2021

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

SIGN LANGUAGE RECOGNITION USING CONVOLUTIONAL NEURAL NETWORKS IN MACHINE LEARNING

Aishwarya Girish Menon
School of Computing and Information Technology
REVA University
Bengaluru, India
aishwaryagm1999@gmail.com

Anusha S.N
School of Computing and Information Technology
REVA University
Bengaluru, India
anushasn24@gmail.com

Arshia George
School of Computing and Information Technology
REVA University
Bengaluru, India
arshiageorge@gmail.com

A. Abhishek
School of Computing and Information Technology
REVA University
Bengaluru, India
gurudatth1907@rediffmail.com

Gopinath.R
School of Computing and Information Technology
REVA University
Bengaluru, India
gopinath.r@reva.edu.in

Abstract: Sign language is a way or means of communication used by individuals with speaking and hearing impairments. It is one of the essential means of communication for such individuals to stay connected with the rest of the world and to express their ideas, needs or beliefs. There is a great need for an efficient and cost-effective real-time translation software or tool in the modern-day world to understand what the disabled individual is trying to express with accuracy. The proposed system is a real-time translation software or tool used for the conversion of hand gestures into natural languages such as English used by people for communication. The translated data will interpret the alphabet or number associated with the sign shown to the live camera feed. The software proposed in this project is created using Python, NumPy, OpenCV, LabelImg and TensorFlow. The image or video obtained from the camera device will be processed using convolutional neural networks (CNN). The CNN model is pre-trained with a large dataset from open sources or using a custom dataset on sign language gestures. Based on the recognition rate and prediction analysis from the CNN model, the provided image or video will be classified as the respective Alphabet or number from the American Sign Language Set. This helps the individuals to understand the sign language used by disabled individuals with ease.

Keywords: Sign Language, NumPy, Python, OpenCV, CNN, LabelImg, TensorFlow, Machine Learning, Real-Time Translation

I. PROBLEM STATEMENT

Sign Language communication is the most basic form of communication between physically disabled and normal, healthy individuals. The existing systems for the conversion of sign language symbols to natural languages require haptic-feedback hardware and, software that do not require haptics are not cost effective and accurate in prediction of the symbols in real-time quickly. Propose an ideal system that is fast in real-time, cost effective with less hardware and higher percentage of prediction accuracy.

II. INTRODUCTION

Sign Language communication is a basic technique used by physically impaired individuals to understand and interpret their thoughts. Most people cannot understand the sign language used by these individuals and there is a need for a user-friendly and efficient real-time translation tool to translate the hand gestures made by the disabled person, into natural languages such as English in the form of letters, words or sentences. Most of the currently existing projects are based on a haptics feedback system which involves a glove being placed on the individual's arm. This doesn't provide much accuracy and it requires extra hardware.

In the proposed project a gesture-based technique is used which consists of machine learning and its method of convolutional neural networks being used for image processing and identification. This technique doesn't require extra hardware components. The proposed software is created using Python as the basic programming language and its supported libraries which include TensorFlow, NumPy and OpenCV. Python is a general-purpose high-level programming language that is interpreted. NumPy is a Python library that supports high-level mathematical operations on multidimensional Three-Dimensional arrays or matrices. TensorFlow is an Open-Source software library mainly used for Machine Learning tasks with a specific focus on training models based on Deep Neural Networks. OpenCV library is used for Real-Time Computer Vision programming functions.

In this paper, the proposed system consists of three parts: Creating the dataset for training, training the CNN model on the data captured and predicting the sign language data in real-time. LabelImg is a graphical annotation tool used in this project to aid in training the model by assigning labels to the region of interest for each captured image dataset used for training and testing the model.

Convolutional neural networks is an algorithm belonging to the set of Deep Neural Networks algorithms. It is used for the analysis of visual images, especially in real-time. CNN trained model will be used to process and identify the sign language shown at the camera in real-time with ease. This project helps in creating an efficient real-time translation system for sign languages using a gesture-based technique, which is more efficient, quicker and accurate than the haptic feedback-based system currently present.

III. LITERATURE SURVEY

The authors in [1] have proposed a system based on skin-color modelling technique. Skin color is determined to differentiate the region of interest that is the hand from the given image and the non-pixels which include the image's remaining background area. Keras is used for training the images. The result obtained by the authors after experimentation using the model was an average testing accuracy of 93% for number recognition, 90% for alphabet recognition and 97% for static word recognition. The system is implemented using a desktop system with an HD webcam for capturing input images. The required dataset was gathered by the continuous capture of images using python. A CNN network having 16 layers was created for this project. SSL and character training were carried out separately by dividing the dataset into two sets, one for training and the other for testing. This helped to determine algorithm performance. TensorFlow and Keras were used to train the network using GT-1030 GPU. 30 individuals tested the system in which six members were sign language interpreters and the remaining members were students who either knew sign language or did not know. Around 30 samples were used to validate Student's T-test and prove the efficiency of the system in identifying each and every hand gesture shown by individuals who did not provide hand gesture images during the training stage. In order to test the accuracy of the model's prediction, the number of accurately classified and recognized gestures on-screen were divided by product of the number of users and multiplied using the total number of trials. A gesture is classified as accurate if the gestures were translated accurately to respective words, sentences, numbers or alphabets in text form within fifteen seconds. If the output is generated after fifteen seconds, then it isn't used for the calculation of the accuracy rate. The main objective of the paper was to create an efficient system to translate sign language gestures in real-time into corresponding words, sentences, letters or numbers.

The authors in [2] proposed a hand gesture recognition system that does not require physical gloves being used by the individual providing input where the haptics prediction system is used for determining the alphabet or number shown by the person using sign language. Here the system converts the Gesture video obtained through live video camera feed of the computer into simple English words and convert the words into meaningful sentences. The CNN process gives the matched results. The CNN model performs convolution-based operations where the image features are detected. For example, in the case of an image of the sign language gesture shown, the CNN model recognizes fingers, hands and edges of the hand. This procedure is called feature extraction. The connected final layers will serve as a classifier for features extracted. It provides probability which determines to which category the image being analyzed belongs. This method is called classification. The model results after prediction, determines the meaning of the hand gesture fed to it. Such a system is feasible in understanding what the disabled individual wishes to interpret or communicate.

The authors in [3] proposed a system where a camera is used for obtaining input. The camera used captures thirty frames every second for real-time videos. The captures were then analyzed to determine the dynamic gestures. The extraction of the region of interest with skin was done using skin filters and by converting each frame of the image to SV color spaces. Four cameras were used for image acquisition at different angles from the center. The main idea is to obtain the frames having linearly independent characteristics and use them as keyframes while the rest are to be discarded. This system uses TD-HMM method where the computation rate is reduced by using similar Gaussian mixture components. Three different HMM implementations exist which include Light-HMM, tied density HMM and multi-stream HMM. With the help of TD-HMM, the authors achieved an accuracy rate of 91.3% while they achieved an 85.14% accuracy rate using HMM and SVM algorithms for Taiwanese Sign Language gestures.

The authors in [4] proposed a modern learning and translation tool for sign language implemented in Machine Learning. The proposed system consists of a camera that captures a video feed. This video feed is processed frame by frame. A library called OpenCV is used to process this video feed. The contours for the frames in the video are identified by darkening the image and obtaining the white border of the hand. This border is used to identify the contours of the hand. The contours are then used to identify the type of symbol provided in the video feed; it is required to train the system with the dataset of images that contain alphabets of the Sign Language. These are mapped to their equivalent English alphabet and fed to the system. The system gets trained on this data and stores the training results as a file. A support vector machine model is used. SVM based training algorithm is used to build the model used for assigning new samples to either of the given set of categories based on the previous categorization of training samples. This makes it a binary linear classifier that is non-probabilistic. A library called scikit-learn provides the necessary SVM Model ready for training. An American Sign Language Interpreter has been implemented based on a Support Vector Machine Classifier (SVM). The system requires some constraints, like a white background and the palm of the hand to face the camera. Further, similar symbols were sometimes misinterpreted for one another.

The authors in [5] proposed a system that uses haptic feedback gloves to predict the alphabets or sentences using Raspberry pi development board. The user wears the glove on their hand. Once the glove is powered on, the user is required to make the required sign language gestures. The analog values from the flex sensors and IMU coordinates are transmitted to the Arduino Nano's microcontroller. The microcontroller compares each of the received data values with the values in the ideal range database which is predefined. If the necessary combination is found, it checks whether the Bluetooth connection to Raspberry Pi board is secure and transfers obtained signals to the board using wireless technology. Raspberry Pi computer board then displays output text on GLCD and the voice output is provided using a speaker for the same. The entire system then reverts to the initial stage to detect the next gesture.

IV. HARDWARE AND SOFTWARE REQUIREMENTS

The Software Requirements include: -

1. Python Programming Language
2. Google Collab or Jupyter Notebook
3. OpenCV
4. TensorFlow
5. SSD MobileNet v2
6. LabelImg graphical annotation tool.

The Hardware requirements include: -

Windows/Linux/MacOS based Computer System with the following minimum specifications: -

1. 8GB Random Access Memory
2. Intel i5 core processor
3. Webcam
4. 64-bit Operating System

V. PROPOSED SYSTEM

The proposed system first captures the images required for training and testing the model with the help of Python and OpenCV programming. The captured images are then stored in a separate new folder. The LabelImg image annotation tool is installed and set in the computer system.

Once setup is completed, the directory where the captured images were stored is selected and the LabelImg tool is used to label the sign shown on each of the previously captured images by drawing detection boxes over the region of interest. The same directory is used as save directory. This creates an XML file for each captured image that was labelled. Next, the partitioning of the captured images along with their respective XML files is performed to split the data into two files, train and test for training and testing the model being created.

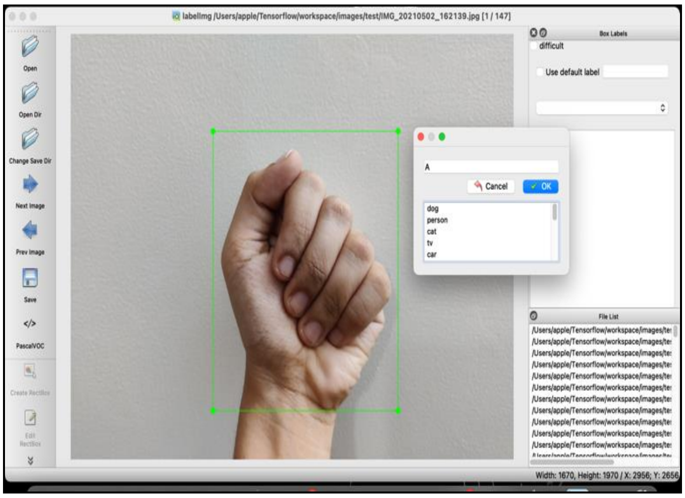

Figure 1. Labelling Images using LabelImg Tool

Figure 1 represents the LabelImg workspace used for labelling the custom dataset captured using python into its associated labels. The figure shown above shows the labelling of the image dataset related to the English alphabet 'A' by dragging a green-colored rectangular detection box across the region of interest which is the sign shown by hand in the image.

Perform setup for TensorFlow Object Detection API and then install Jupyter Notebook. Once this is done, different paths are set which are necessary while coding such as image path, model path, annotation path etc. A Label map is created to represent each Sign Language Symbol. TF records are generated by using the generate_tf_record script. TF records are special formats used by TensorFlow API during programming. Then clone the official TensorFlow object detection library from the GitHub repository of TensorFlow Model Zoo which is used for training the model. Copy the model Configuration from the SSD MobileNet v2 model to the train folder of the proposed system. Update the copied configuration to function efficiently and accordingly for the proposed system. Train the model for the System proposed using the respective Python scripts for training Deep Learning or Convolutional Neural Networks model using command prompt. Once training is finished and loss and epoch rates are analyzed to determine the model's prediction accuracy. Load the trained model on the system.

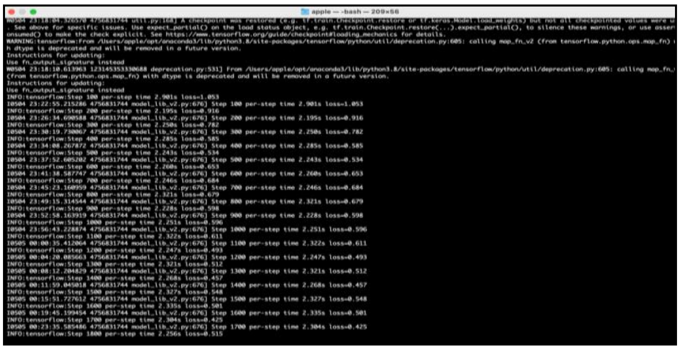

Figure 2. Training the model using machine learning on command prompt with loss score for respective epochs.

Figure 2 represents the Training process in the command prompt which displays the loss rate for respective epochs which improves to provide accurate predictions when the number of epochs is increased, and training is performed for longer durations.

Finally run the real-time detection code to boot the real-time detection screen which shows the detection box around the sign shown at the camera in real-time with a prediction accuracy rate and the name of the predicted label around the sign shown, on-screen as seen in Figure 4.

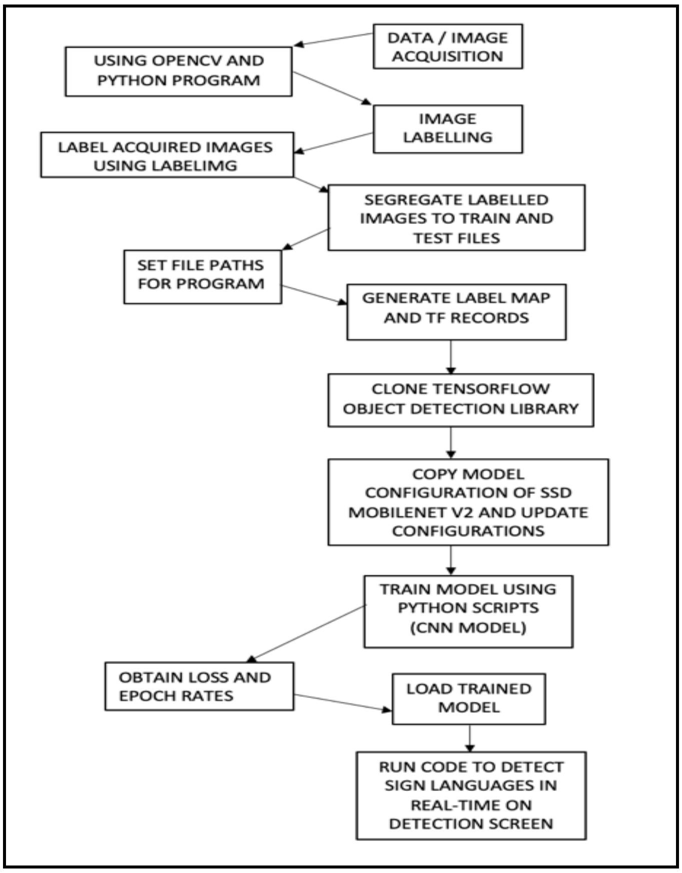

Figure 3. Flow Diagram of Data Collection, Model Training and Real-Time Detection

Figure 3 represents the summarized flow diagram of the entire training and testing process carried out during the prediction model creation.

VI. Experimental Results

The results obtained in this experiment consists of a Real-time Object Detection screen using the trained CNN model, to detect different sign language symbols shown by the user at the camera. The screen as shown in Figure 4 will display a detection box around the region of interest that is, the sign language shown by the user and displays the label it predicts. In simpler terms, it represents the alphabet or number associated with the sign language shown, as well as the prediction accuracy rate which shows how sure the trained model is that the shown symbol is a particular alphabet or number.

The training process resulted in an average rate of 0.086 loss score for 10,000 epochs. Since the loss score is low, the model is working close to the accurate range.

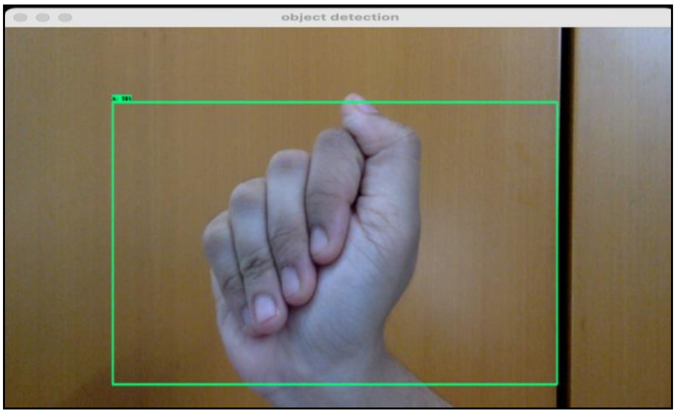

Figure 4. Real-Time Object Detection Screen with Predicted Label and Accuracy Rate

Figure 4 represents the output screen obtained once the trained model is loaded and executed for displaying the real-time sign language recognition screen. The figure represents a sample output that shows that the sign or gesture shown on the image above is translated to the alphabet ‘A’ in the English language. The percentage shown represents the fact that the classification model is 96.0% sure that the shown gesture is the alphabet ‘A’ in this image. The prediction accuracy rate for each alphabet or number varied between 96.0% to 99.0%.

VII. Conclusion

The proposed Sign Language recognition system focuses on recognizing static sign language gestures shown in real-time and its classification and conversion to the respective number or alphabet. For this purpose, the machine learning model is constructed using a supervised learning method with convolutional neural networks.

The system helps to avoid the purchase of any costly hardware components which are required to build existing systems based on Haptics Feedback Technology. This project will help individuals with hearing or speaking impairments to communicate with others in an efficient manner.

VIII. FUTURE WORK

Further improvements that can be performed on the proposed system include the combining of sign language alphabet symbols to English Sentences and Words with further conversion of these sentences or words to other regional languages, using natural language processing.

Another future improvement involves the creation of a mobile, hand-held device that can convert sign languages in real-time on the go.

IX. ACKOWLEDGEMENT

We would like to thank REVA University, our respected guide and other faculty for all the guidance and support provided to us in the completion of this project and technical paper.

X. REFERENCES

[1] N. Adaloglou, T. Chatzis, I. Papastratis, A. Stergioulas, Georgios Th. Papadopoulos, V. Zacharopoulou , G.J. Xydopoulos , Kl Atzakas2 , D. Papazachariou ,P. Daras, “A Comprehensive Study on Sign Language Recognition Methods”, arXiv:2007.12530v1 [cs.CV] 24 Jul 2020

[2] S. Bachani, S. Dixit, R. Chadha, A. Bagul, “Sign Language Recognition Using Neural Network”, International Research Journal of Engineering and Technology (IRJET), Volume: 07 Issue: 04, 2020

[3] L.K.S. Tolentino, Ronnie O.S Juan, August C. Thio-ac, M.A.B. Pamahoy, J.R.R. Forteza, Xavier J.O. Garcia, “Static Sign Language Recognition Using Deep Learning”, International Journal of Machine Learning and Computing, Vol. 9, No. 6, December 2019

[4] R.G Rajan, M.J Leo, “A Comprehensive Analysis on Sign Language Recognition System”, International Journal of Recent Technology and Engineering (IJRTE), Volume-7, Issue-6, March 2019

[5] S.Javed, Ghousia Banu S, J.A.S Kani and A. Rahman,” Wireless Glove for Hand Gesture Acknowledgment: Sign Language to Discourse Change Framework in Territorial Dialect”, Juniper publications, Robotics and Automation Engineering Journal, Volume 3 Issue 2 June 2018

[6] Suharjitoa, R. Anderson, F. Wiryana, M.C Ariesta, G.P Kusuma, “Sign Language Recognition Application Systems for Deaf-Mute People: A Review Based On Input-Process-Output”, 2nd International Conference on Computer Science and Computational Intelligence 2017, ICCSCI 2017, 13-14 October , Bali, Indonesia, 2017

[7] Mrs.Dipali Rojasara Dr.Nehal G Chitaliya, “Indian Sign Language Recognition –A Survey”, International Journal of Engineering Research & Technology (IJERT) Vol. 2 Issue 10, October – 2013

[8] S.S Kumar, V.L. Iyangar, “Sign Language Recognition Using Machine Learning”, International Journal of Pure and Applied Mathematics Volume 119 No. 10 2018

[9] Aditya Das, Shantanu Gawde, Khyati Suratwala, Dhananjay Kalbande, "Sign Language Recognition Using Deep Learning on Custom Processed Static Gesture Images", Smart City and Emerging Technology (ICSCET) 2018 International Conference on, pp. 1-6, 2018.

[10] Shadman Shahriar, Ashraf Siddiquee, Tanveerul Islam, Abesh Ghosh, Rajat Chakraborty, Asir Intisar Khan, Celia Shahnaz, Shaikh Anowarul Fattah, "Real-Time American Sign Language Recognition Using Skin Segmentation and Image Category Classification with Convolutional Neural Network and Deep Learning", Region 10 Conference TENCON 2018 - 2018 IEEE, pp. 1168-1171, 2018.

[11] Yuto Hirota, Tetsuya Oda, Nobuki Saito, Aoto Hirata, Masaharu Hirota, Kengo Katayama, "Proposal and Experimental Results of a DNN Based Real-Time Recognition Method for Ohsone Style Fingerspelling in Static Characters Environment", Consumer Electronics (GCCE) 2020 IEEE 9th Global Conference on, pp. 476-477, 2020.

[12] KP Nimisha, Agnes Jacob, "A Brief Review of the Recent Trends in Sign Language Recognition", Communication and Signal Processing (ICCSP) 2020 International Conference on, pp. 186-190, 2020.

[13] H. Muthu Mariappan and V. Gomathi, "Real-Time Recognition of Indian Sign Language," 2019 International Conference on Computational Intelligence in Data Science (ICCIDS), 2019, pp. 1-6, doi: 10.1109/ICCIDS.2019.8862125.

[14] G. A. Rao, K. Syamala, P. V. V. Kishore and A. S. C. S. Sastry, "Deep convolutional neural networks for sign language recognition," 2018 Conference on Signal Processing And Communication Engineering Systems (SPACES), 2018, pp. 194-197, doi: 10.1109/SPACES.2018.8316344.

[15] Ashwinkumar.U.M and Dr. Anandakumar K.R, "Predicting Early Detection of cardiac and Diabetes symptoms using Data mining techniques", International conference on computer Design and Engineering, vol.49, 2012.

[16] Khalil Bousbai, Mostefa Merah, "A Comparative Study of Hand Gestures Recognition Based on MobileNetV2 and ConvNet Models", Image and Signal Processing and their Applications (ISPA) 2019 6th International Conference on, pp. 1-6, 2019.

