IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Gesture Recognition of Indian Sign Language using CNN

[1]Sumathi A, [2]Lakshmi Bhaskar, [3]Ashwini S Savanth

[1]Associate Professor, [2]Assistant Professor, [3]Assistant Professor

[1,2,3] Department of ECE,

[1,2,3]BNMIT, Bangalore, Affiliated to VTU, Karnataka, India

Abstract: Speaking disability community use hand gestures as a communication tool with the abled community. It is useful for connecting people with computers. This paper mainly focuses on converting sign language that has been communicated through the means of digital media such as live video chats, video messages, etc. to text. The speech-impaired individual performs the necessary signs/gestures that indicate the message they want to convey to the camera that records the video. This video is then pre-processed and implemented in the model that has been trained to recognize similar object patterns in videos. The model is trained using a dataset comprising of images. The model is then checked for accuracy and necessary steps are taken to ensure that it predicts accurately. This model is used to recognize sign language gestures performed by speech-impaired individuals for easier communication.

Index Terms **- Hand gestures, CNN, Hearing-impaired, Sign language.**

1. INTRODUCTION

Community needs and emerging new technologies are driving researchers to find new and innovative ways to address these needs. The deaf community uses Sign languages to communicate with other communities. The difficulty in communication exists between the hearing impaired and the normal community. To overcome this barrier between the two, several investigations were conducted to develop a system that can recognize sign language used by the deaf community. An assessment of some of these recent technologies is crucial in comparing their methodologies with the accuracy of their results.

Considering the current period as the age of technology, it is more important than ever to harness the use of these innovative technologies to improve human life. Especially people with hearing impairments, find it difficult to interact with people. The deaf community uses sign language to communicate which is not easily understood by all people. Sign language is not a universal language. Most of the countries use different sign language which is based on the spoken language of that country. In India, the Indian sign language (Figure 1) used is a deviation from American Sign Language (ASL).

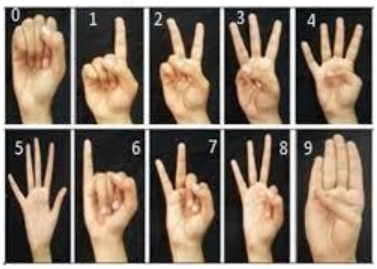

Figure1: Signs of Indian sign language

Over the past few years, machine learning is on the high for technological solutions. Machine learning is an algorithm that learns from the data, analyses the data, and applies the learning to make decisions. Today, various applications use large amounts of data, and with this ever-growing amount of data, automated techniques for intelligently analyzing data are required. Such analysis can be done with the help of machine learning as it can detect, predict and/or make decisions on the data.

2. LITERATURE SURVEY

The detection of sign language in real-time can be performed differently, latest few are CNN, deep learning, etc. few models are designed to identify the sign language using segmentation of skin color and neural network models. The training model used is multi-class CNN which splits the data into two categories dataset for training and a dataset for testing. The results were implemented as graphics on smart devices. The classification accuracy was up to 88.25%. few drawbacks were due to M and Z which requires multiple secondary templates [1]. Few researchers have worked on the sensor and vision-based systems for the accurate extraction of complex and constant variations in movements of the head and hand. These methods are efficient and supportive for deaf people. A dynamic real-time hand gesture recognition is proposed, in this technique, a video file is converted onto HSB color space, then pre-processing is performed after this segmentation of skin cells is performed. For precision, the depth of information is also

considered. Through the image frames, Hu moments and motion trajectory are extracted, and later SVM classification is performed. the performance of this method was up to 97.5% for four specific Indian sign languages. Another method using Artificial intelligence using CNN, illustrated a gesture recognition method for Indian sign language. Through this method a dataset was provided for mobile selfie sign language consisting of five distinct subjects, it was performed for 200 signs in various inclinations along with varied background environments. The results in comparison with other models for classification depict an accuracy of 92.88% for the recognition rate. A finger spelling recognition method for American Sign Language using a KNN classifier was implemented and showed around 99.8% for k=3. This method was more appropriate for basic education applications. A comparative study also shows a few methods using neural network structure with feed-forward option and recognition of dataset and training each sign language using MATLAB with results around 93.4% for three languages, Hausdorff distance algorithm and Hu invariants for processing the various hand movements and recognize different letters using OpenCV libraries with results up to 90.9% [2].

For better accuracy pool layer approach was better compared to the prediction approach, as it has a feature in which size is increased per vector frame and then processed by RNN. A 2048-dimensional vector representing convoluted features of the image is considered, which proved to be better. Even with this improvement, the RNN discovered random noise leading to overfitting in this method [3]. Another work-related gesture/motion detection and recognition system was carried out for the American Sign language protocol for hearing disabled persons. In this protocol, a support vector regression with the RBF kernel was used. It had a high correlation between measured and predicted values for data samples used, almost showing 100% accuracy. Further, this work would help to create a complex instruction set a communicate with devices, sensors, and information nodes [4].

A continuous gesture recognition system using a data glove-based was the earlier works by Starner and Pentland, it was efficient in recognizing one of the gestures of American Sign Language. Few others also [resented an approach with probabilistic neural networks classification of gestures using Fourier descriptors, Hu invariant moments as hand orientation and shape descriptors using Baum-Welch algorithm using LBR topology with the forward algorithm to train and recognize thus leading to 90% recognition rate for few gestures only. Yet another work for recognizing static finger spelling by capturing RGB images through a single camera was done, using CNN for both synthetic and real-time images from Japanese sign language gave a promising result, this also orients to further process 3D model-based methods, with extended training sample database for performance improvement [5].

Methodology

Implementation of this work involves two main steps: dataset preparation and training the model.

2.1 Dataset Preparation

The dataset is one of the important components of a machine learning problem. It is the key to training the model that will be used to analyze the gestures performed by the user. A model is a software component and needs multiple instances of training to produce desirable outputs. A dataset constituting more information will ensure that the model is trained better. For this work, the dataset that is going to be used is a collection of images consisting of a set of multiple similar images for a single gesture. When more images are used for a single gesture, the model will have better accuracy in analyzing the gesture performed by the user but using multiple images will also require more processing power to train, hence an optimal number of images must be taken.

To capture images for the dataset, a code was used constituting dependencies such as OpenCV, UUID, time, and os. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in commercial products. OpenCV is incorporated as a dependency for manipulation of images and accessing the camera to capture the images in real-time. Post-training the model, this dependency is used for the model to analyze the gestures caught on camera in real-time. UUID is a dependency that is used to rename image files. Once the images are captured via Open CV, they are not named in acceptable characters. UUID takes these images and renames them so that they are organized accordingly. Time is a dependency that consists of functions used to instill a delay. This code uses the time to create a delay between capturing images, to make it user-friendly. OS is a dependency, used to define the file paths. When the image is captured using OpenCV and stored in the required directories, OS functions are used.

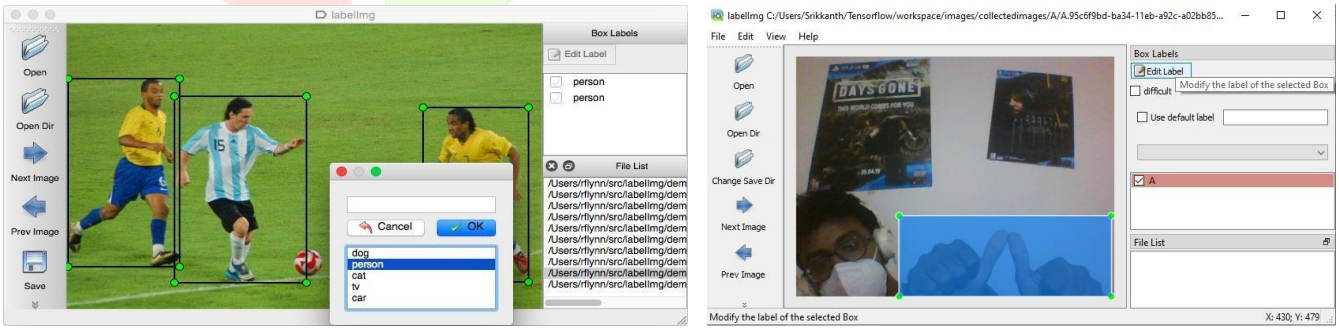

Figure 2: LabelIMG Interface

LabelIMG is a graphical image annotation tool written in Python and uses QT for its graphical interface. The interface is shown in Figure 2. LabelIMG supports labeling in VOC XML or YOLO text file format. LabelIMG is used to label images for object detection. After the images are labeled, an annotation is created for each image which contains the folder the image is in, file name, path, and attributes of the image, such as size, width, height, depth, the coordinates of the image, and the area of focus in the image. In this work, the images contained in the dataset are typically a person performing a gesture that denotes something. The image is opened with LabelIMG and the area of focus, i.e. the hand(s) with which the gesture is performed is highlighted and labeled with its respective name. This process is repeated for all the images in the dataset. LabelIMG allows setting the open directory and saves the directory to two different paths. To train a model, the dataset of each gesture is separated into two folders, a training folder, and a testing folder. The images in the training folder are used to train the model and the images in the testing folder are used for testing the model after the model is trained for the images present in the training folder. The separation is in a 70:30 split with 70 being in training and 30 in testing.

2.2 Training the model

The model was trained using the TensorFlow object detection API. Tensor Flow API object discovery is a framework for creating a deep learning network that solves object discovery problems. The existing pre-trained models are referred to as Model Zoo. It is a collection of pre-trained models trained on COCO, KITTI, and open image datasets. They help initialize models and training on a novel dataset. The various architectures used in the pre-trained model are described in Figure 3.

Model name	Speed	COCO mAP	Outputs
ssd_mobilenet_v1_coco	fast	21	Boxes
ssd_inception_v2_coco	fast	24	Boxes
rfcn_resnet101_coco	medium	30	Boxes
faster_rcnn_resnet101_coco	medium	32	Boxes
faster_rcnn_inception_resnet_v2_atrous_coco	slow	37	Boxes

Figure 3: Training

```
In [2]: labels = [{'name':'1', 'id':1},
                  {'name':'2', 'id':2},
                  {'name':'3', 'id':3},
                  {'name':'4', 'id':4},
                  {'name':'5', 'id':5}]

        with open(ANNOTATION_PATH + '\label_map.pbtxt', 'w') as f:
            for label in labels:
                f.write('item { \n')
                f.write('\tname:\'{}\'\n'.format(label['name']))
                f.write('\tid:{}\n'.format(label['id']))
                f.write('}\n')
```

Figure 4: Create a label map

The steps involved in training the model are given below.

To create a label Map: Labels are created for the dataset present. Each dataset is given an ID and a name.Each label is mapped to the ID as shown in Figure 4.

To create TF records: The TF Record format is a simple format for storing a sequence of binary records. Protocol buffers are a cross-platform, cross-language library for efficient serialization of structured data. Protocol messages are defined by .proto files, these are often the easiest way to understand a message type.

Downloading the pre-trained models: There are already pre-trained models in a framework which is referred to as Model Zoo. This includes a collection of pre-trained models trained on the COCO dataset, the KITTI dataset, and the Open Images Dataset. These models can be used for inference if we are interested in categories only in this dataset. For this work, the pre-trained model used is ssd_moilenet_v2_fpnlit_320x320_coco17_tpu-8 which is taken from the TensorFlow model zoo.

Update the config file for transfer learning: The config file, which acts as the control hub to the entire code, consists of set instructions and commands. The config file must be updated, with the necessary instructions that are needed to set the model up to begin transfer learning and to train the model.

Train the model: Code is executed to train the model. The accuracy with which the model predicts the information in real-time is highly influenced by the number of steps in which it is trained. Statistically, the higher the number of steps used to train the model, the greater the accuracy. The drawback of training the model using many steps is the large amount of processing power used by the CPU and GPU, which technically cannot be achieved in every system. The number of steps used in this case is 5000.

Load trained models: The checkpoints are set during training of the model and it is saved. The program, while executing, loads the checkpoints in the trained model, to use for producing the output needed.

Detection in real-time: This part of the code uses every aspect of the trained model and uses OpenCV to detect the sign language gestures in real-time.

3. Experimental Results

Sign language, a form of communication for speech-impaired individuals, faces challenges when an individual who is not well versed in it, attempts to communicate with a speech-impaired person. To overcome that hassle, this work aims to bridge the gap between speech impediment and hearing impediment individuals by using AI-ML techniques to understand and process the visual sign language and present it in an understandable form for those who find it difficult to understand it. Considering that most of the population is not well versed in communicating using sign language, this working model can solve the issue well. This working model can be efficiently used for communication between two individuals where the difficulty in interpretation of sign language persists. The model detects sign language in real-time after executing the code corresponding to the above steps. Once the code is executed, a small window opens, which uses the device's camera and implements the model in the real-time camera vision, using the training, it detects the gestures performed by the person sitting in front of the camera. The code uses OpenCV to use the camera and implement the model in it for detection. The accuracy of the output depends on the size of the dataset and the number of training steps. This model can detect and recognize gestures for the numbers 1-5 and alphabets A to E with an accuracy of > 80%. The accuracy can be improved by increasing the training dataset. Each gesture has been trained for 60,000 steps. Table 1 shows the individual accuracies trained for each case.

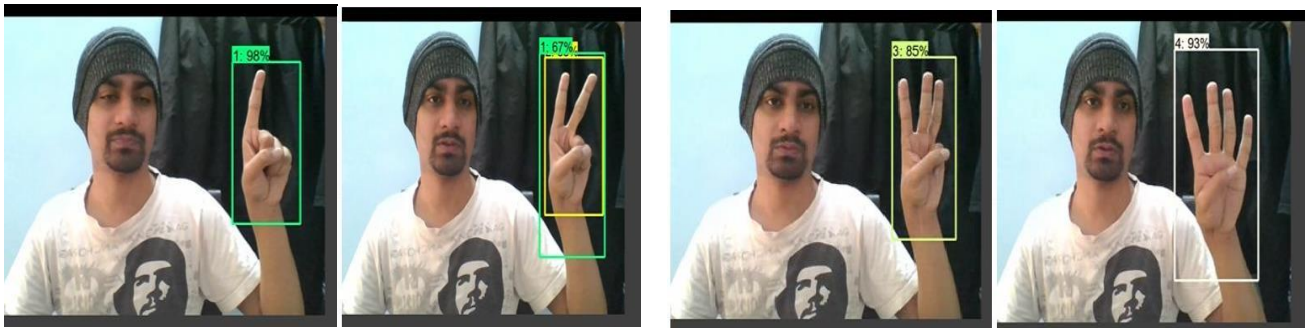

Figure 5: Detection of digits 1, 2, 3 & 4

Table 1: Accuracy obtained for different gestures

Gesture	Accuracy in recognition (%)
1	87
2	83
3	85
4	98
5	100
A	79
B	83
C	99
D	84
E	87

4. CONCLUSION

As the times are progressing, and as technology is getting more advanced, the way of living is getting much simpler. The development in technology would have meaning only if it is used in various fields to reduce human effort in as many ways as possible. The current technology should be used to its maximum potential for the positive development of society. This work aims to develop a device that makes life easier for speech-impaired individuals and people communicating with them. A machine learning model is trained in python, using two major python libraries: OpenCV and TensorFlow. The video is analyzed by using OpenCV for gestures performed in it using the model that was trained using transfer learning in Tensorflow, and the sign language gestures performed in the video are output, post which, this process is made possible in real-time by developing a portable device that translates sign language into speech. The main portable hardware component comprises a Raspberry pi-4 processor, in which the machine learning model is deployed for real-time conversion of sign language to text and then audio. The Raspberry Pi camera that is compatible with the processor is used for capturing the video in which sign language is performed for further processing. Post-processing the output which is text is converted into audio using another program coded into the processor. This audio is then played by a speaker connected to the Raspberry processor. This model helps detect the gestures of Indian Sign Language and acts as an interface for better communication.

ACKNOWLEDGMENT

We are grateful to the students and authorities of BNM Institute of Technology, Bangalore, India for their encouragement and support extended to carry out this work.

REFERENCES

[1] S. Kadam, A. Ghodke and S. Sadhukhan, "Hand Gesture Recognition Software Based on Indian Sign Language," 2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT), 2019, pp. 1-6, doi: 10.1109/ICIICT1.2019.8741512.

[2] Rawan A. Al Rashid Agha, Muhammed N. Sefer, and Polla Fattah, "A comprehensive study on sign languages recognition systems using (SVM, KNN, CNN and ANN)," In Proceedings of the First International Conference on Data Science, E-learning and Information Systems (DATA '18). Association for Computing Machinery, New York, NY, USA, 2018, Article 28, 1–6. https://doi.org/10.1145/3279996.3280024

[3] Masood, S., Srivastava, A., Thuwal, H.C., Ahmad, M, "Real-Time Sign Language Gesture (Word) Recognition from Video Sequences Using CNN and RNN," In: Bhateja, V., Coello Coello, C., Satapathy, S., Pattnaik, P. (eds) Intelligent Engineering Informatics. Advances in Intelligent Systems and Computing, vol 695. Springer, Singapore, 2018 https://doi.org/10.1007/978-981-10-7566-7_63

[4] Tse, Rita. "Detection and Recognition of Sign Language Protocol using Motion Sensing Device." 2018.

[5] H. Hosoe, S. Sako and B. Kwolek, "Recognition of JSL finger spelling using convolutional neural networks," 2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA), 2017, pp. 85-88, doi: 10.23919/MVA.2017.7986796.

[6] S.Gollapudi "Learn Computer Vision Using OpenCV: With Deep Learning CNNs andRNNs" 1st ed 2019 Publisher: Apress ch:5 Object Detection and Recognition 978-1-4842-4260- 5;978-1-4842-4261-2

[7] Hernandez V, Suzuki T, Venture G, "Convolutional and recurrent neural network for human activity recognition: Application on American sign language," PLoS ONE 15(2): e0228869, 2020. https://doi.org/10.1371/journal.pone.0228869