Deep Learning Approach to Recognize Sign Language Gestures

Vijayalakshmi K
Computer Science and Business Systems
Rajalakshmi Engineering College

***Abstract*—Hand gesture recognition is a very useful technological advancement as an alternative user interface for providing real-time data to a computer. Classical interaction tools like mouse, keyboard limit the way we interact with our system. Also, this can be very helpful for the deaf and dumb people in interacting and communicating with other people and also computer systems. It can also be used as a sign language translator for the deaf and dumb people. The aim of this project is thus to create a sign recognition detector using Deep Learning and CNN that can detect basic patterns like numbers or signs based on the trained data set being used. Existing gesture recognition software make use of a hardware system design with motion sensors that are not effecient. By implementing a CNN model using Tensor flow, this system aims to make use of just the device camera to live capture and detect the sign automatically. The data set can be freshly trained based on the user requirement. The future scope of the project will be to train the model to store and convert the text into program input or voice which may require more complex algorithms and understanding of Neural Networks and Deep Learning Technologies.**

Keywords—Hand gesture recognition, Deep Learning, Tensorflow, sign language, CNN model

I INTRODUCTION

GENERAL:

The sign language is a very important way of communication for deaf-dumb people.In sign language each gesture has a specific meaning. Sign language is a gesture-based language for communication of deaf and dumb people. It is a non-verbal language used by them to communicate more effectively with other people. Unfortunately, there are not many technologies which help in connecting this social group to the rest of the world. Understanding sign language is one of the primary enablers in helping users of sign language communicate with the rest of the society.

Existing System:

Currently the main sign language recognition approach used is sensor-based sign detection. But sensor-based devices are not convenient as hand gloves, helmet etc. are required by the user.Hardware devices like kinetic sensors (by Microsoft) develops a 3D model of the hand and observes the hand movements and their orientations. A glove-based approach was another technique wherein the user was required to wear a special glove that recognized the position and orientation of the hand.

Limitations In Existing System:

- High initial setup cost and less practical feasibility
- User requires adequate knowledge of technology to use the hardware system
- Considerable E-waste due to large number of sensors used

Our statement of contribution in this paper is to develop a system that can effectively recognize sign language using deep learning techniques. The proposed system intends to help computers recognize sign language, which could then be interpreted by other people. Convolutional neural networks have been employed to recognize sign language gestures.The image dataset used consists of dynamic sign language gestures captured on a system attached camera. Preprocessing was performed on the images, which then served as the cleaned input. The results are obtained by retraining and testing this sign language gestures dataset on a convolutional neural network model.

The paper is structured as follows: In section II.I, the technical terms used and system prerequisites are

discussed.In the following sections II.II and II.III, the user profile attributes are described and the motivation behind the work is established.Section III discusses the prior work and related researches previously done in this area. Section IV presents the research methodology. The hypothesis and the implementation of the proposed method is described in detail in Section V and the implemented system is evaluated in Section VI. Finally, Section VII concludes the paper by highlighting the research contributions,research limitations and future plans to extend this work.

II.I PRELIMINARIES

1)***Tensorflow-*** TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications.

2)*Convolutional Neural Networks-* A Convolutional Neural Network (CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other.

3)Gesture Recognition- Gesture recognition is a computing process that attempts to recognize and interpret human gestures through the use of mathematical algorithms. Gesture recognition technology that is vision based uses a camera and motion sensor to track user movements and translate them in real time.

4)***Transfer Learning-*** Transfer learning (TL) is a research problem in machine learning (ML) hat focuses on storing knowledge gained while solving one problem and applying it to a different but related problem. It is a popular approach where pre-trained models are used as the starting point on computer vision and natural language processing tasks given the vast compute and time resources required to develop neural network models on these problems.

5) ***SSD Mobilenet***: The mobile-ssd models is a Single-Shot multibox detection network intended to perform detection. It works parallely with a convolutional neural network in Object Detection models.

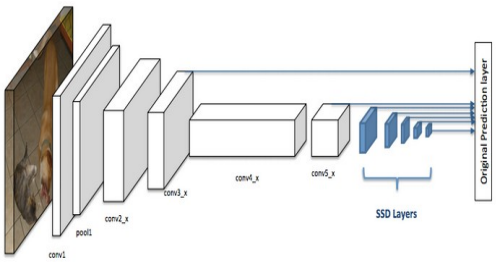

Fig.1: Architecture of a convolutional neural network with a SSD detector

II.II SYSTEM PRE-REQUISITES

The project is done on Windows 7(64-bit RAM). Python version 3.7 is used to implement the code. Object detection API and tensorflow is pre-installed.Microsoft Visual C++ Redistributable for Visual Studio 2015, 2017 and 2019 is needed for installation of Object detection API.Mobilenet-ssd model is used as Single-Shot multibox Detection (SSD) network intended to perform object detection.

Metric	Value
Type	Detection
GFLOPs	2.316
MParams	5.783
Source framework	Caffe*

Table 1: SSD Model Specification

A GPU setup is required to train the images using cudart to achieve better accuracy.GPU support requires a CUDA®-enabled card in Windows and Ubuntu systems. Tensorflow GPU has the following requisites:

Prerequisites
Nvidia GPU (GTX 650 or newer)
CUDA Toolkit v10.0
CuDNN 7.6.5
Anaconda with Python 3.7 (Optional)

II.III USER PROFILE ATTRIBUTES

In our work, we consider the following user profile attributes for incorporating into the proposed system. It is intended in a way to cater to all levels of user by considering the following aspects:

1)**Skill level-** This includes current competency level of the learner and the target skill level that is to be achieved. At level 1, it consists of basic primer knowledge on the subject User Level 2 is the ability to participate in and understand the working process. Level 3 are profiles that have the skill to contribute or alter the process of implementation. The system takes into consideration all the levels to create a system that is easy to use,flexible and also allows dynamic user

experience to help level 3 users train and customize the system as they require.

2) **Target User**- The system is mainly designed to cater to the deaf and dumb people who are unable to communicate with other people who do not understand sign language. It can be used as a feature in online conferencing systems to automatically display sign language as text.

II.IV MOTIVATION

Communication is one of the basic requirements for survival in society.Deaf and dumb people communicate among themselves using sign language but normal people find it difficult to understand their language.Sign language recognition is a problem that has been addressed in research for years. However, we are still far from finding a complete solution available in our society.Also, lack of datasets along with variance in sign language with locality has resulted in restrained efforts in such hand sign gesture detection. Our project aims at taking the basic step in bridging the communication gap between normal people and deaf and dumb people using a simple and effective deep learning vision based model to recognize sign languages in real time.Effective extension of this project to words and common expressions may not only make the deaf and dumb people communicate faster and easier with the outer world, but also provide a boost in developing autonomous systems for understanding and aiding them.

III PRIOR WORK

This section briefly presents the prior work and related searches carried out in the context of sign language conversion. In 2015, Taner Arsen published his research in the International Journal of Computer Science &Engineering Survey on converting sign language to text using a motion capture sensor. In 2019, Surejya Suresh published an IEEE paper titled Sign Language Recognition system using Deep Neural Networks. The main focus of this work is to create a vision based system, a Convolutional Neural Network (CNN) model, to identify six different sign languages from the images captured. G. A. Rao, K. Syamala, P. V. V. Kishore and A. S. C. S. Sastry presented their research titled "Deep convolutional neural networks for sign language recognition" in the 2018 Conference on Signal Processing And Communication Engineering Systems.

A paper was published by C. J. L. Flores, A. E. G. Cutipa and R. L. Enciso, "Application of convolutional neural networks for static hand gestures recognition under different invariant features" in the 2017 IEEE XXIV International Conference on Electronics Electrical Engineering and Computing. All these cited works use signal sensing and processing techniques instead of image recognition using labels and machine learning models. Since it is a relatively budding and new area, not much work has been done in applying machine learning to develop a simplistic approach to leverage the user tagged parameters.

IV RESEARCH METHODOLOGY

Sign language is a major form of communication used by almost 70 million people throughout the world. The problem arises as there is no standard mode of sign language practised. Different people use different signs in different places. Hence, a a methodical solution that can train itself based on one-time user input is the foundation of the research. The experimentive dataset is a primary dataset collected by recording the images of certain basic signs used worldwide. On the collected dataset, we divided our approach to tackle the classification problem into three stages.

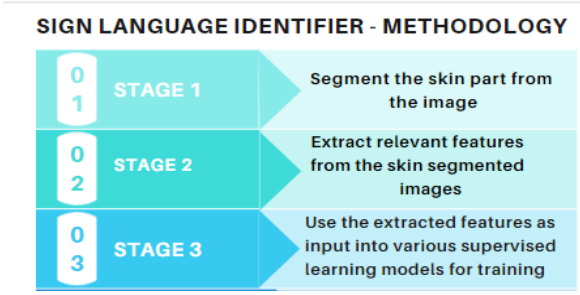

Fig.1 : Stages of Implemented Methodology

The first stage is to segment the skin part from the image, as the remaining part can be regarded as noise. The second stage is to extract relevant features from the skin segmented images which can prove significant for the next stage i.e. learning and classification. The third stage as mentioned above is to use the extracted features as input into various supervised learning models for training and then finally use the trained models for classification.

The use of simple machine learning techniques in this research is done to eliminate the ue of sensors or motion capture signal processors by training a model to detect live images based on a set of trained parameters. The research method extends to work on creating a system that allows user to train certain signs at any time as per user requirement.

V IMPLEMENTATION OF PROPOSED WORK

The implementation of the proposed system is done in 5 major steps:

STEP1: A separate sub-directory is created for the IMAGES_PATH where the collected images and their labels will be stored. The labels are initialised and live images are captured and collected using opencv and stored in the respective label folder.

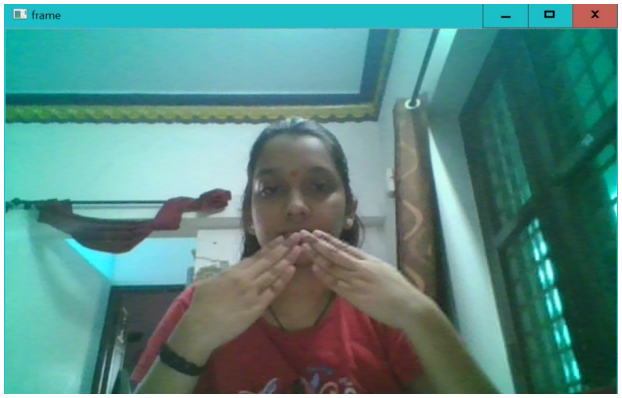

Fig:2 Live capturing sign images for dataset

STEP2: Each of these images are labelled using Labellmg tool to graphically label the images and also remove the skin part and unwanted background from the image for training. The attributes and coordinates of the labelled image is stored in an XML file.

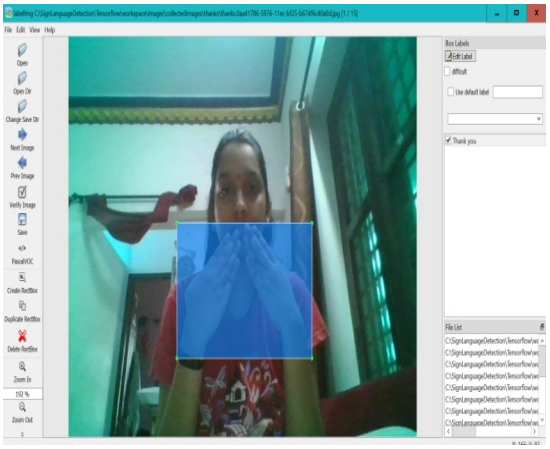

Fig.3: Label and clean images using Labellmg

STEP 3: Tensorflow Object Detection pipeline configuration is setup. The labels are converted and saved in a pbtxt file that contains text graph definition in protobuf format.

STEP 4: The configurations are pipelined from the config file and trained using the SD Mobilenet model in Object Detection in python. 5000 train steps are used to train the model. Transfer Learning is used to train a deep learning model and load the trained model from checkpoint.

```
Out[9]: {'model': ssd {
          num_classes: 90
          image_resizer {
            fixed_shape_resizer {
              height: 320
              width: 320
            }
          }
          feature_extractor {
            type: "ssd_mobilenet_v2_fpn_keras"
            depth_multiplier: 1.0
            min_depth: 16
            conv_hyperparams {
              regularizer {
                l2_regularizer {
                  weight: 3.9999998989515007e-05
                }
              }
```

*Fig.4: Generated SSD model_*

STEP 5: The category boxes are used to visualize and label the boxes. Normalised coordinates are used for the visualised box images.OpenCV is used to detect the sign language in real time.

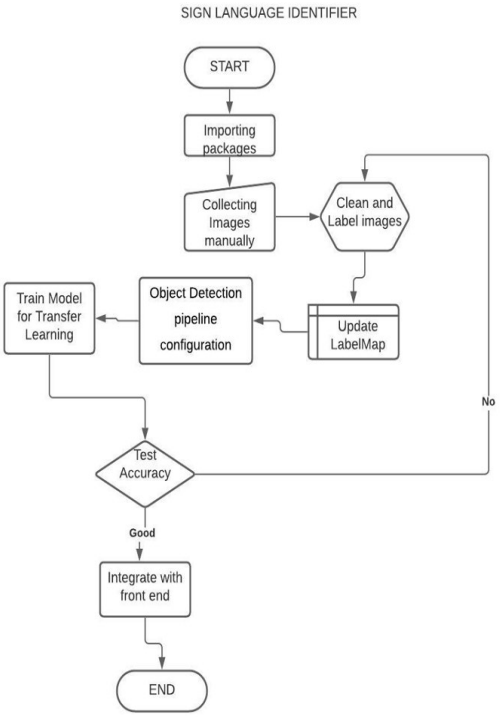

Fig.5 *Implementation Workflow*

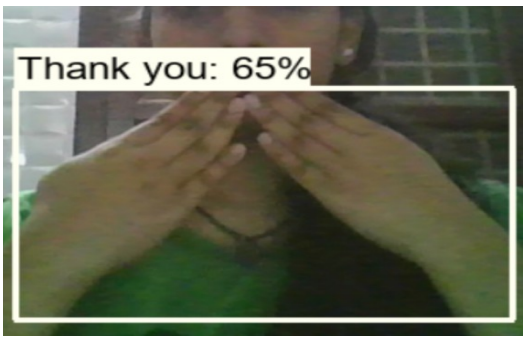

Fig.6: Live Detection of signs

VI EVALUATION OF IMPLEMENTED WORK

The usage of deep learning model is proven to be more efficient than machine learning algorithms in terms of the computation speed and power of the model.

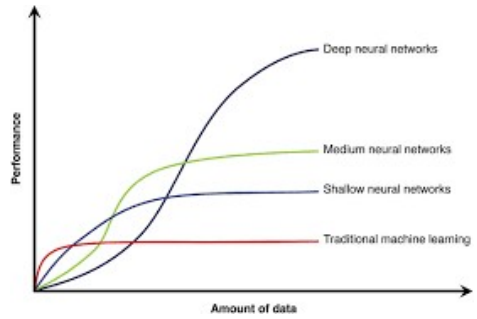

Fig.7 :Average precision and recall values

The training and validation accuracy of the deep learning model is plotted. An 80% training accuracy and average of 60% validation accuracy is obtained as represented in Fig.8.

Fig.8: Training and Validation Accuracy

The model is successfully trained and run with no compilation errors. An average accuracy of 60-80% is achieved in the detection.

The detection occurs within a proximity of 40-50 metre from the system camera.

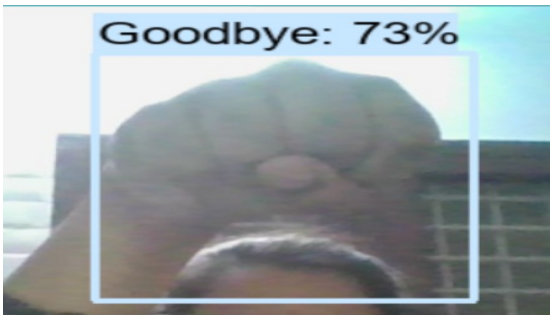

Fig.9: Live detection of signs

The detection box is dynamic and expands its dimensions based on the size of the input image processed. AP (Average precision) is a popular metric in measuring the accuracy of object detectors like the SSD model. Average precision computes the average precision value for recall value over 0 to 1. The average precision and recall values are enlisted in Fig.10.

```
Average Precision (AP):
  AP                % AP at IoU=.50:.05:.95 (primary challenge metric)
  AP^IoU=.50        % AP at IoU=.50 (PASCAL VOC metric)
  AP^IoU=.75        % AP at IoU=.75 (strict metric)
AP Across Scales:
  AP^small          % AP for small objects: area < 32^2
  AP^medium         % AP for medium objects: 32^2 < area < 96^2
  AP^large          % AP for large objects: area > 96^2
Average Recall (AR):
  AR^max=1          % AR given 1 detection per image
  AR^max=10         % AR given 10 detections per image
  AR^max=100        % AR given 100 detections per image
AR Across Scales:
  AR^small          % AR for small objects: area < 32^2
  AR^medium         % AR for medium objects: 32^2 < area < 96^2
  AR^large          % AR for large objects: area > 96^2
```

Fig.10: Average precision and recall values

A second run through was carried out to increase the accuracy by feeding more images as input and re-training the model. The accuracy improved by a margin of 5-10% indicating that with more number of samples the accuracy of the model is bound to inrease.

Fig.11 :Improved Accuracy of Detection

VII.I LIMITATIONS OF RESEARCH WORK

Live capturing of the sign images is a tedious and time-consuming process. Training images with large size is directly proportional to the computational power of the system. Since the system RAM is limited to 12 GB, the optimal image size trained was 93X63 pixels decreasing the quality and accuracy of detection.Since the live images captured correspond to a lot of memory space, only a limited number of signs are trained. Also, the system used does not have a GPU setup machine and hence the acuracy and predictions are subject to a marginal error and opacity.

VII.II FUTURE ENHANCEMENTS

Since the project is done on a low scale with few resources and limited capabilities, there are many future enhancements that can be done to make it a real world application for use. A GPU machine can be used to achieve more accuracy and trained with more sign images. The model can be fed and used with an external camera for capturing and detecting sign languages from a longer distance.

1.a We can develop a model for ISL word and sentence level recognition. This will require a system that can detect changes with respect to the temporal space.

1.b The model can also be programmed with teleconferencing software like Google Meet, Zoom as a functionality extension to enable deaf and dumb people to interact in the virtual medium using the sign detection model.

1.c We can develop a complete product that will help the speech and hearing impaired people, and thereby reduce the communication gap.

ACKNOWLEDGMENT

This work was supported by the Department of Computer Science and Business Systems of Rajalakshmi Engineering College. We are grateful to Mr.Bhuvaneswaran for providing the necessary guidance during the course of this project.

REFERENCES

1 *https://www.researchgate.net/publication/282839736_Sign_Language_Converter*

2 http://sersc.org/journals/index.php/IJAST/article/view/20937/10563

3 https://tensorflow-object-detection-api-tutorial.readthedocs.io/en/latest/

4 https://academic.oup.com/jdsde/article/11/4/421/411839

5 https://link.springer.com/chapter/10.1007/978-3-642-02707-9_3

6 https://www.sciencedirect.com/science/article/pii/S1877050918321331

7 https://www.ijert.org/sign-language-to-text-and-speech-translation-in-real-time-using-convolutional-neural-network

8 *https://www.ripublication.com/ijaer18/ijaerv13n9_90.pdf*

9 *https://www.researchgate.net/figure/Comparison-accuracy-Faster-R-CNN-R-FCN-SSD-and-YOLO-models*

10 *http://reports.ias.ac.in/report/19049/real-time-indian-sign-language-recognition*