Real-Time Sign Language Recognition using Transfer Learning

Lipika Goel
Department of Computer Science and Engineering
Gokaraju Rangaraju Institute of Engineering and Technology
Hyderabad, India
lipika.bose@gmail.com

NVS Pavan Karthik
Department of Computer Science and Engineering
Gokaraju Rangaraju Institute of Engineering and Technology
Hyderabad, India
navuluruvspavankarthik@gmail.com

Madasu Jashwanth Naidu
Department of Computer Science and Engineering
Gokaraju Rangaraju Institute of Engineering and Technology
Hyderabad, India
jashuwanth2001@gmail.com

Pradyumna Sinha
Department of Computer Science and Engineering
Gokaraju Rangaraju Institute of Engineering and Technology
Hyderabad, India
pradyumnasinha01@gmail.com

Anand Thota
Department of Computer Science and Engineering
Gokaraju Rangaraju Institute of Engineering and Technology
Hyderabad, India
s.v.k.anand.t@gmail.com

***Abstract*—For decades, many researchers have been working on finding efficient solutions to implement sign language recognition systems through various technologies. Sign languages are visual languages that communicate through the hand, facial expressions, as well as body movements. Sign language is the bridge that connects us to the world of those who have impaired hearing or verbal ability. It allows them to understand the world around them through visual descriptions and, as a result, contribute to society. Thus, It strikes our mind to bridge the gap between the hearing impaired and normal people & make communication easier. Many researchers faced many challenges such as long training times to train a deep neural network model such as CNN or RNN, huge datasets of millions of images, high computational power requirements, the usage of expensive sensory devices, etc. Hence, the proposed method in this paper uses a technique known as "Transfer learning" where we use a high-accuracy pre-trained model and fine-tune its final layer parameters according to our task that is, Sign Language recognition which significantly reduces the training time and requires a small set of data of less than 20 images per gesture. Thus in this paper, we created a custom dataset generated using the system webcam, labeled data using the LabelImg tool, used Tensorflow Object detection API and downloaded a pre-trained model named "SSD MobinetNet V2 " and applied Transfer Learning. Challenges faced in the project were issues with the correct labeling of images, a few installation errors, and lighting and background issues in a few detections. But overall, the method proposed is highly efficient in saving a lot of training time, providing high accuracy, and using fewer images for training.**

Keywords—SignLanguageRecognition,Transfer Learning,OpenCV, TensorFlow,DeepLearning

I. INTRODUCTION

Sign language is a visual-oriented language used by the Deaf and mutes involving hand gestures as a source of communication, instead of spoken language producing articulate sounds. Vocal and sign languages both contain words. Words in vocal languages often consist of a small number of vowels, consonants, and tones. Whereas, sign languages are constructed from a small collection of forms, orientations, places, hand movements, and frequently facial expressions. While sign language is similarly governed by complicated grammar, spoken language uses rules to create full statements.

Although there are commonalities between many sign languages, sign languages are not universal and typically cannot be understood by one another because of different sign languages produced by different countries/continent.

Table-1: Sign Languages in different Countries

S.No.	Sign Language	Country
1	Indian Sign Language	India
2	American Sign Language	U.S.A
3	Japanese Sign Language	Japan
4	Australian Sign Language	Australia

As a result, signs are not complete gestures but rather can be analyzed as a collection of linguistically important elements. SLs are made up of the following indivisible features, just as spoken languages:

- Manual Features
- Non-Manual Features

Therefore, translating sign language into words using an algorithm or model can aid in closing the communication gap between those who have hearing or speech disabilities and the rest of society.

Using a Transfer Learning methodology we are introducing a Real-Time Sign Language Detector. In this, the code in Jupyter notebook, and by using our webcam we will be capturing the hand gesture images, will be detecting as well as displaying its correct sign language meaning in real-time (i.e., in the live video). It plays a significant role in improving communication between the general people ,deaf and mute people

Sign Language is used by more than 350 million people across the world. Hence, by introducing our model, Mute people can use gestures and we will be able to detect their language and make them understand for communication.

II. Review of Related Works

Sign Language recognition has been studied from years and researchers are trying to improve the model using various methodologies. In [1], the authors have listed down all the techniques used for sign language from 2011 - 2022. We have taken references from research papers from 2014 to 2022.

In the [5], the researchers discussed various kinds of specially designed data acquisition sensory devices to acquire input signs.CyberGlove, Sensor Glove, and Polhemus FASTRAK are the input devices used. After acquiring the videos, the authors used a Fourier analysis approach to detect periodicity and the VQPCA clustering method to recognise the trajectory. VQPCA is a hybrid clustering and local PCA method that finds a locally linear representation of the data while making no assumptions about the underlying data distribution. Through the above methods, sign language was detected in the video. However, the limitations of using such devices are they are highly expensive and also make the user feel uncomfortable since they need to be always attached to their bodies while they are carrying out their work. [2014].

The publication [3] focuses on using the ST-GCN approach to overcome the problem. In this methodology, there are spatial networks used whose topology changes with time. The authors classify the body components based on their eccentric movements. It consists of a root node as its first component, the centripetal group consisting of the neighbourhood nodes closest to the skeleton's centre of gravity. Finally, there is the centrifugal group, which consists of the nodes farthest from the gravitational centre. The centre of gravity is assumed to be the average coordinate of all skeletal joints in a single frame. Lexicon of American Sign Language The dataset that was used was Video Dataset (ASLLVD).In around 10,000 samples, there are 2,745 different indications exhibited. The first step consists of obtaining the videos based on the metadata file, then the videos are segmented, and based on the segmented videos the estimation of skeletons is done. Then the key points are filtered. In this case, they use 27 of 130 estimated key points. The dataset is divided into smaller subsets in this split, training samples make up 80% while test samples make up 20%. Finally, normalize and serialize samples that are compatible with ST-GCN input. These are the steps for using the ST-GCN method to recognise sign language. When compared to other relevant strategies, the suggested strategy produced results with less expressive performance. It is important to look for methods that can enhance the knowledge about the motions of the approximated coordinates, particularly those of the hands and fingers, which, despite being very delicate, are crucial for the expression and interpretation of the signs. Note: ST-GCN stands for Spatio-Temporal Graph Convolution Networks. [2020].

The authors of [2] present a source of communication for the Dumb and Deaf using the ANN and CNN algorithms which involved a huge dataset of American Sign Languages (ASL). The images are then converted to greyscale and a Gaussian Blur filter is applied to remove noise. To extract the hand from the background and resize the image to 128*128 pixels, an Adaptive Threshold value is used. The processed image is passed to the model and is trained thoroughly. The letter is displayed if it is found in more than 50 frames. A KERAS model is built and implemented to design an efficient, user-friendly sign language recognition system. This method has some restrictions when photographs are taken indoors because it performs poorly in sunlight and the size of the images must be raised to improve accuracy.

The paper [4] emphasizes using the approach of KNN. The authors used this method to create two databases, one for single-handed gestures and the other for double-handed gestures, in which the image is divided into two parts for pixel balancing using the Hierarchal Centroid Technique. The image matrix is used to generate a Feature Vector, which is scaled down to 40*50 pixels and a one-dimensional array with 2000 elements for each gesture. Then using those vectors, a KNN classification model is generated. After that, the KNN classifier is used to classify unknown instances through association with the known instances using distance functions that operate using feature vectors. But this approach has shortcomings as its high sensitivity leads to noise, the classifiers lack robustness, and the sample set used is of insufficient size.

J. Bhattacharya [6] utilized the PCBR algorithm where the images for the dataset were taken using a camera, and their sizes were reduced to 120x120. The skin colour model is used to segment and detect hands, and the Principle Curvature Based Region Detector (PCBR) is then used to extract features. Instead of looking for points, the PCBR focuses on identifying regions. Hence, it collects local structural cues like edges and curvilinear shapes as they are more robust to variation in colour, pose, and intensity. It accomplishes this via the watershed algorithm. Wave packet decomposition is then used to reduce the extracted feature's dimensionality from 192 to 112. The necessary feature vectors are then created using the extracted features. The generated vectors are then classified using a multi-class non-linear support vector machine. This method was created to reduce inter-class ambiguity among ISL alphabets. However, due to the less stable nature of the PCBR features, accuracy may suffer. Performance may suffer if the features overlap.

The authors of [7] demonstrated the SIFT algorithm, in which images are captured using an integrated webcam and saved in a directory, and the captured images are compared to images in the database using the SIFT Algorithm. After that, the SIFT method transforms an image into a large collection of local feature vectors. The input of hand movement can take any form, and the gesture is converted into its recognisable character by passing out a single-dimensional array of 26 characters corresponding to the alphabet. This method has a limitation in that it fails to classify if the images are shape-like.

The researchers in [9] performed classification with and without background using Deep Learning by two different datasets, One contains different hand gestures, the other contains hand segmentation models and a pre-trained CNN model is involved. Images in both the data sets are resized to 331 x331 resolution and online augmentation techniques are implemented during training to reduce overfitting in terms of background and then the gesture is detected. In terms of classifying without background, after training the hand segmentation model the best hand detection model is segmented and output sign language is recognized. Both these approaches have a frequent change in accuracy on the skin colour and hand movement.

The model was created by the authors of [10] by dividing it into five stages. They collected sign images for classification using various image-acquisition devices (webcam, hand glove, Kinect, and so on). After partitioning the image into meaningful segments to obtain the region of interest, pre-processing techniques were applied to the input images to remove noise and improve quality. The compact feature vector is extracted to improve learning accuracy and the visibility of the result. The feature extraction stage's output assists the classification stage by looking for features that can effectively distinguish between classes and aid in achieving high recognition accuracy. Colour, texture, and shape features characterise the extracted features from the interested region, and the sign has been determined after several iterations. Problems arose as a result of the various environments in which the images were captured, illuminations, computational time, and gesture tracking.

Image frame acquisition is completed first in [11], followed by hand segmentation. The features are then extracted after finishing the hand tracking. After feature extraction, categorization is completed. They eventually learn the output gesture. SVM is the algorithm in use here. This supervised machine learning model and related learning model analyses data from regression analysis and categorization. Only static ISL numeric signs can be used with this system. Because ISL alphabets, words, and sentences must be included for complete sign language recognition, the ISL recogniser system cannot be considered complete.

Before pre-processing, which involves noise removal and compression, the input is first captured in the 2D or 3D position of the hand making the sign . Then, extract the polygonal description and trajectory features. The classifying process is now complete. Ensemble subspace KNN is the classifier with the best performance. They employ the K folds cross-validation algorithm. In this instance, K=5. More measurements that are unrelated to the hand trajectory are used by the authors in [12] for this process. The dimensionality and simplicity are decreased by these qualities.

Sentences in sign language that have been recorded on video are used as input in [13] by using LSTM and RNN approach. In order to streamline the input data, the input images were resized to 224 × 224 pixels, and any words with videos with over 45 frames or less than five frames were eliminated. Every video sequence was passed through a CNN to extract features from the images. They have all been padded to ensure that each of these feature tensor sequences was the same length. The generated tensors were cached and used as input for two different detection systems. A multilayer RNN from PyTorch, as well as an LSTM. LSTM performed better than RNN in comparison. However, the drawback of using accuracy is that each prediction is either 0 or 1 — there is no probability in between.

A Kinect device is used by the authors in [14] , which captures the video of the signer. The video would be divided into frames, resulting in a raw image sequence. The boundaries will then be determined by processing this sequence. The head and hands are the two main subparts of the body that the camera is capturing. The head's postures, movements, and facial expressions are extracted, as well as the hands' gestures and poses. The WLASL Dataset will then match all the data for categorization. The generated words won't follow grammar conventions. Thus, Google's T5 model will produce semantically valid statements. An audio generator will be used to create speech from this output. Kinect's drawbacks include its short range for depth detection, inability to capture motion in a frame that lasts less than 0.5 seconds, and sensitivity to sunlight.

The authors of [15] extracted features from the image using the HOG (Histogram of Oriented Gradients) technique and used image processing techniques such as edge detection as a pre-processing step. Then they applied Support Vector Machine (SVM classifier) to classify the gesture. They used a dataset that consists of 6000 images which have all 26 English alphabets. The dataset was divided into a training set of 4800 images and a testing set of 1200 images and then used to train the SVM model.

The method proposed in [16], uses the CNN approach in an environment of Tensorflow to provide highly accurate results. They have used the CIFAR-10 dataset to train the CNN model. They have performed reshaping and transposing the images in CIFAR-10 to make them suitable for the Tensorflow environment. They have displayed the workflow in the Tensorflow interface and how to train the model. The major challenge faced by this method was its high computational cost. Without GPU installed in the system, it would be very difficult to train a CNN model, and is also time-consuming.

The method used by the authors in [8] is using the Deep Forest technique. There are mainly three key processes: recognition of the human skeleton, Colour based skin segmentation model and detection. In this study, the American Sign Language Lexicon Video Dataset as their input and after they have extracted the key points of the frame which are thought of as the joints of the skeleton. Then, the dataset is divided into training and testing sets with a ratio of 80:20.Finally, The deep-based forest-based classifier is used to detect the sign languages with the help of DTW and HMM

III. Proposed Methodology

The proposed method is explicitly split into two major phases

- Pre-processing phase
- Detection phase

<u>Pre-Processing phase:</u> This phase consists of three steps: Collect images using Webcam, Labelling images using the LabelImg interface and Split the collected into training and test images.

Acquiring Images - By using the system's webcam 16 images for each label were collected. In total there are 6 labels in the project. Hence the total number of collected images is 96.

Labelling images using the LabelImg interface - By using a labelling computer vision software named "LabelImg" interface as shown in the fig-1, the images were labelled according to their classification. An XML file is generated for each image consisting of details such as the class type, bounding box configuration and no. of pixels. Using this software labelling of images has been made easy and it automatically provides all the labelled images of the same size which is suitable for TFOD API.

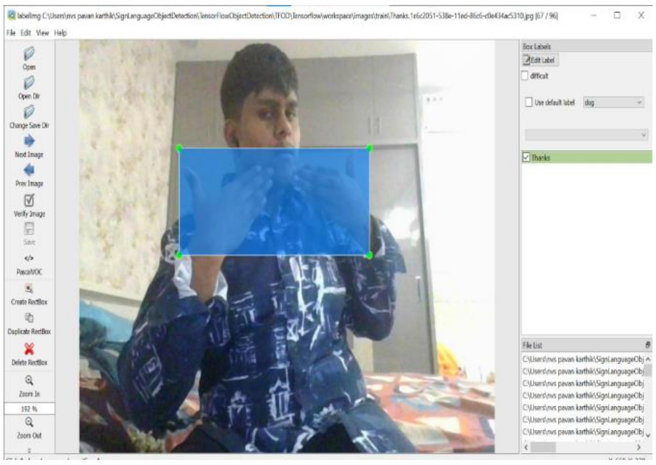


Fig-1 : Labelling images using LabelImg Interface

Splitting images into testing and training – The custom-acquired dataset is then divided into training and testing datasets with a split ratio of 75:25 respectively. The 12 images from each label are considered in the training dataset and the rest 4 images are part of the testing dataset.

<u>Detection Phase:</u> This is an important phase in the model and is constituted into eight different steps starting from the setup paths step.

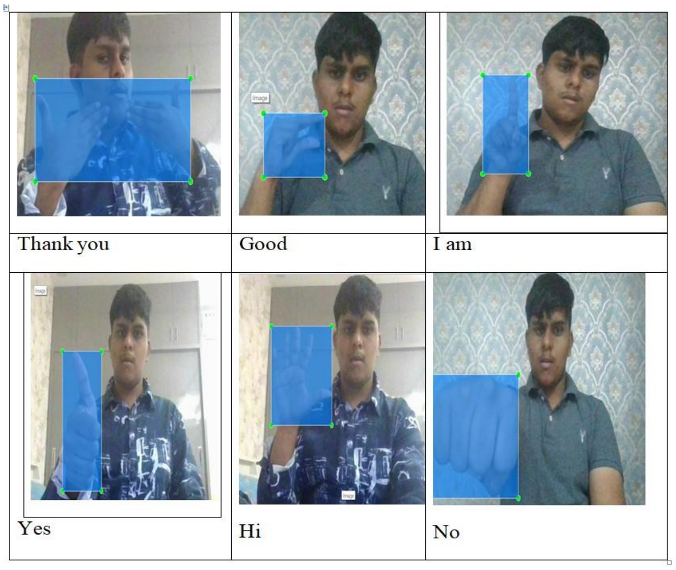


Fig-2 : Labelling all the six classes

Setup paths – In this step, all the required directories of the files and folders are defined. These paths are used to access required models and files in the project. These are defined for the Windows Operating System(O.S).

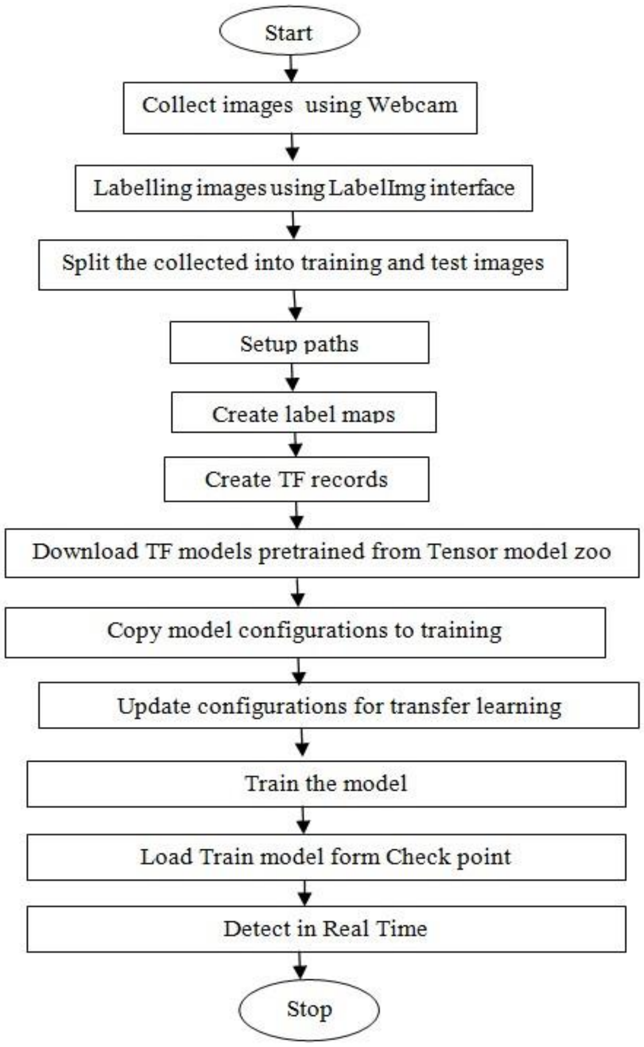

Start
Collect images using Webcam
Labelling images using LabelImg interface
Split the collected into training and test images
Setup paths
Create label maps
Create TF records
Download TF models pretrained from Tensor model zoo
Copy model configurations to training
Update configurations for transfer learning
Train the model
Load Train model form Check point
Detect in Real Time
Stop

Fig-3 : Steps involved in the proposed method

Create label maps – The labels of sign languages are defined using a python dictionary known as label maps. It is file which is stored as “labelmap.pbtxt” extension in the annotation path. The six labels used in the project are: Hi,I_am, Good, Yes, No, and Thank You.

Create Tf Records – To work with TensorFlow Object Detection API, a binary file is created which is known as TF Record. Here, two TF Records are generated by running a python script “generate_tfrecord.py” in the command prompt. One TF Record is for training purposes and the other is for testing.

Download TF models pre-trained from Tensor model zoo – Since the approach used in this paper is transfer learning, a pre-trained model is downloaded from the official Tensor Flow model zoo library. The pre-trained model used in the method is “SSD_MobNetV2”.

Copy model configurations to training – After the above step, the model configuration i.e., “pipeline. config” is copied into the training folder for modification in the copied model and to avoid using the original model.

Copy model configurations to training – After the above step, the model configuration i.e., “pipeline. config” is copied into the training folder for modification in the copied model and to avoid using the original model.

Update configurations for transfer learning – As illustrated in the figure-4, the last layers of the pre-trained neural network model are fine-tuned according to the custom labels data. The modifications performed on the pre-trained model configuration using the python code were :

1. Updating the number of classes to six
2. modifying the paths defined in the model
3. providing our custom-generated dataset of training and testing images.

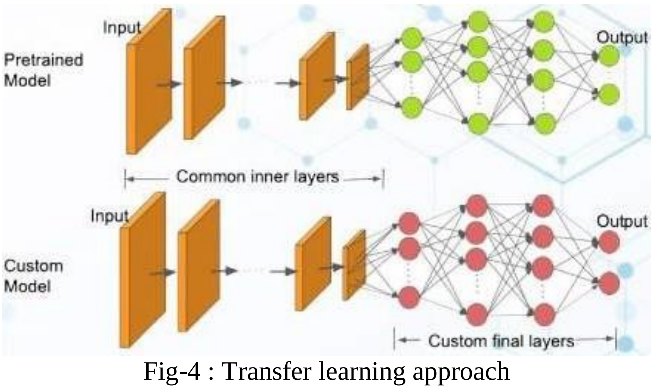


Fig-4 : Transfer learning approach

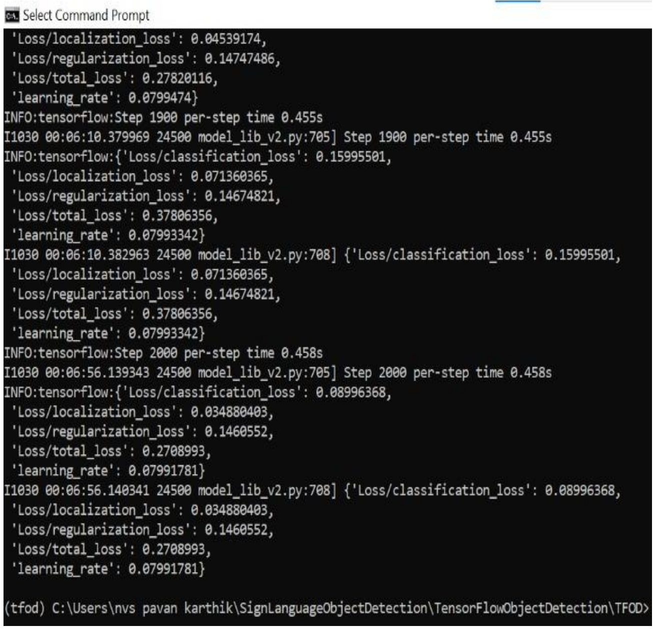
Train the model – After fine-tuning the final layers of the pre-trained model, the training step is carried out by running the python script "model main tf2.py," which uses the model's updated configuration "Checkpoint 0".The training process is done for 2000 epochs and three checkpoints were obtained and are stored in the model's folder. The “Checkpoint3” configuration is taken into consideration due to its high accuracy score.

Load train model from checkpoint – As discussed in the previous step the “Checkpoint3” configuration file has high accuracy score and thus, it is loaded for evaluating the sign language detection.

Detect in real-time – In this final step, the testing image is taken as input for the model and is converted into a 2D matrix which is called as Tensor. This Tensor is provided to the “detect_fn” which is a TensorFlow function defined by the official TensorFlow Object Detection API and is used to visualize the bounding boxes, identify the type of class in the image and display the prediction score of the class on the test image

IV. Results and Discussions

The model has been trained for 2000 epochs and it obtained the loss of 0.08% as shown in the figure-5, which is very low and hence it clearly demonstrates that the model is estimated to be highly accurate in terms of sign language detection.



```
 'Loss/localization_loss': 0.04539174,
 'Loss/regularization_loss': 0.14747486,
 'Loss/total_loss': 0.27820116,
 'learning_rate': 0.0799474}
INFO:tensorflow:Step 1900 per-step time 0.455s
I1030 00:06:10.379969 24500 model_lib_v2.py:705] Step 1900 per-step time 0.455s
INFO:tensorflow:{'Loss/classification_loss': 0.15995501,
 'Loss/localization_loss': 0.071360365,
 'Loss/regularization_loss': 0.14674821,
 'Loss/total_loss': 0.37806356,
 'learning_rate': 0.07993342}
I1030 00:06:10.382963 24500 model_lib_v2.py:708] {'Loss/classification_loss': 0.15995501,
 'Loss/localization_loss': 0.071360365,
 'Loss/regularization_loss': 0.14674821,
 'Loss/total_loss': 0.37806356,
 'learning_rate': 0.07993342}
INFO:tensorflow:Step 2000 per-step time 0.458s
I1030 00:06:56.139343 24500 model_lib_v2.py:705] Step 2000 per-step time 0.458s
INFO:tensorflow:{'Loss/classification_loss': 0.08996368,
 'Loss/localization_loss': 0.034880403,
 'Loss/regularization_loss': 0.1460552,
 'Loss/total_loss': 0.2708993,
 'learning_rate': 0.07991781}
I1030 00:06:56.140341 24500 model_lib_v2.py:708] {'Loss/classification_loss': 0.08996368,
 'Loss/localization_loss': 0.034880403,
 'Loss/regularization_loss': 0.1460552,
 'Loss/total_loss': 0.2708993,
 'learning_rate': 0.07991781}

(tfod) C:\Users\nvs pavan karthik\SignLanguageObjectDetection\TensorFlowObjectDetection\TFOD>
```

Fig-5: Training the model for 2000 steps in CMD Prompt

The training of model involved the usage of collection of training and testing images and also the label map file which contains the different types of classes defined. In TensorFlow training process, saved TensorFlow models are obtained after a fixed number of iterations which are known as checkpoints. After training the model for 2000 epochs as seen in the figure - 4, three checkpoints were created. In general, the last obtained checkpoint has the highest accuracy of detection when compared to the previous checkpoints. Hence, after using the Checkpoint3 configuration model which was obtained at step 2000 on the testing image, the accuracy of detecting the sign language was found to be around 90% which tells that the model is performing well and can be further improved by adding more images to the class which displayed low accuracy.

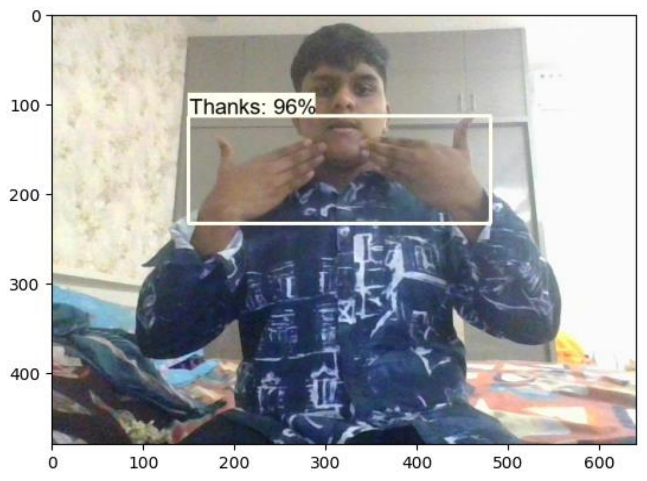


Fig-6 : The above figure illustrates the accuracy of 96% for the class "THANKS"

The model is not only restricted to detecting the sign languages that we have trained but also it can be used to train and detect various kinds of gestures. This approach we used can be applied to various kinds of computer vision use cases such as Face mask Detection and other sign language detections. The pros of the approach discussed in this paper are :

- Less number of images are required
- The model has trained very accurately with less training.
- Time taken for model training is relatively very less

V. Conclusion

This paper proposed an efficient solution to implement Sign Language Recognition in a real-time application using the Transfer Learning approach. Transfer Learning is a technique that refers to using the pertinent components of a machine learning model that has already been trained to solve a different but related issue. For example, the knowledge acquired from the model which was used to detect cars can be used in the model being trained to detect bikes or other vehicles. Numerous researchers have been performing various kinds of techniques for many years. Most of their vision-based approaches had the disadvantage of taking more time for training the model and using huge datasets. Along with that, their computational costs were high in the systems in which they run the code. When it came to sensor-based approaches, many of the devices used were expensive, and the final product made was requiring lots of sensory devices to achieve high accuracy. In addition to that, those products are uncomfortable for the customers to wear and use.

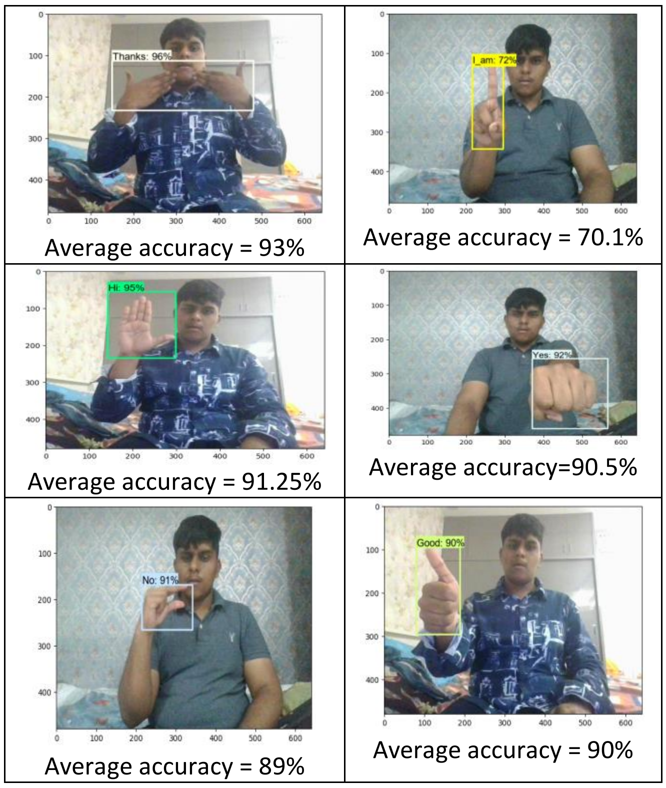


Figure-7 : Average Accurancy of each class detection

Hence, we chose a vision-based approach rather than a sensory-based approach which just uses a laptop device and would be the least expensive method to implement. The proposed approach in this paper utilizes majorly two phases. The first phase is the pre-processing phase which mainly covers the installation of all the required python and TensorFlow environments. It also makes the labeled data ready for the model to implement the Transfer Learning approach and to use the Official TensorFlow Object Detection API. It is simple to create, train, and deploy object detection models due to the TensorFlow Object Detection API, an open-source framework built on top of TensorFlow. It has various kinds of pre-trained models which help computer vision enthusiasts to save time and apply a transfer learning approach in their work. Hence, this tool helped us to take less time to train the sign language recognition model and achieved high average accuracies for all the kinds of sign languages defined in the paper and are above 80% accurately detected. To conclude, in this paper we could achieve great accurate results by just using a very small custom-made dataset of 180 images of 6 sign Language variations where 20 images were collected for each kind and training for 2000 epochs. Such high accuracies with small data were possible mainly due to the application of the transfer learning technique in the project which is highly effective. This recognition model defined in the paper can be extended to detect any number of sign languages and is not restricted to six. To achieve greater accuracies, we can increase the size of the dataset and also increase the training steps (aka epochs). In future work, by adding all the techniques to improve accuracy, the proposed method can be implemented and embedded into various kinds of online video meetings such as Zoom, and Google Meet, which can help the deaf and mute people interact without any communication barrier and solve the communication problems faced by them.

References

[1] Boban Joksimoski, Eftim Zdravevski, Petre Lameski, Ivan Miguel Pires, Francisco José Melero, Tomás Puebla Martinez “Technological Solutions for Sign Language Recognition : A Scoping Review of Research Trends, Challenges, and Opportunities ”,vol. 4,2016

[2] Reddygari Sandhya Rani, R Rumana and R. Prema, " A Review Paper on Sign Language Recognition for The Deaf and Dumb",vol. 10,2021.

[3] Cleison Correia de Amorim, David Macedo and Cleber Zanchettin, "Spatial-Temporal Graph Convolutional Networks for Sign Language Recognition",2020

[4] Ashok kumar sahoo and kiran kumar Ravulakollu, "Vision based Indian Sign Language Character Recognition " , vol. 67,2014

[5] Ashok K Sahoo, Gouri Sankar Mishra and Kiran Kumar Ravulakollu, "SIGN LANGUAGE RECOGNITION: STATE OF THE ART",vol. 9,2014

[6] J. Rekha, J. Bhattacharya, and S. Majumder, "Shape. Texture and Local Movement Hand Gesture Features for Indian Sign Language Recognition",pp. 30 – 35,2011

[7] Sakshi Goya, Ishita Sharma, Shanu Sharma, "Sign Language Recognition System For Deaf And Dumb People",vol. 2,2013

[8] Qifan Xue, Xuanpeng Li, Dong Wang and Weigong Zhang,"Deep Forest-based Monocular Visual Sign Language Recognition",vol. 9,2019

[9] Kanchon Kanti Podder, Muhammad E. H. Chowdhury, Anas M. Tahir, Zaid Bin Mahbub, Amith Khandakar and Muhammad Abdul Kadir, "Bangla Sign Language (BdSL) Alphabets and Numerals Classification Using a Deep Learning Model",2022

[10] I.A. Adeyanju, O.O. Bello and M.A. Adegboye, "Machine learning methods for sign language recognition: A critical review and analysis", 2021.

[11] Radha S.Shirbhate, Vedant D.Shinde, Sanam A.Metkari, Pooja U.Borkar,Mayuri A.Khandge" Sign Language Recognition using Machine Learning Algorithm",2020

[12] Ala addin I.Sidig, Sabri A. Mahamoud, "Trajectory Based Arabic Sign Language Recognition",2018

[13] Nihar Joshi, Ryan Gudal, Tianyu Jiang, Samantha Clark, “American Sign Language Recognition Using Computer Vision”,2021

[14] Ayushi N. Patani, Varun S. Gawande, Jash V. Gujarathi, Vedant K. Puranik, Tushar A .Rane, ”Methodologies for Sign Language Recognition: A Survey”,2021

[15] Omkar Vedak, Prasad Zavre, Abhijeet Todkar, Manoj Patil, "Sign Language Interpreter using Image Processing and Machine Learning",2019

[16] Priyal Jawale, Hitiksha Patel Chaudhary, Nivedita Rajput, "Real-Time Object Detection using TensorFlow",2020