Real Time Sign Language Detection

Sahil Rawal[1], Dhara Sampat[2], Priyank Sanghvi[3]

[1,2,3]*Students, Department of Electronics and Telecommunication Engineering, K. J. Somaiya Institute of Engineering and Information Technology, Mumbai, Maharashtra, India.*

---***---

Abstract - *Throughout the long term, correspondence has played an indispensable job in return of data and sentiments in one's day to day existence. Sign language is the main medium through which deaf and mute individuals can interact with rest of the world through various hand motions. With the advances in machine learning, it is possible to detect sign language in real time. We have utilized the OpenCv python library, Tensorflow Object Detection pipeline and transfer learning to train a deep learning model that detects sign languages in Real time.*

***Key Words*: American Sign Language (ASL), Sign Language Detection, Convolution, Neural Network (CNN), Transfer learning, Tensorflow, OpenCV**

1. INTRODUCTION

The disabled are the main users of sign language, and just a few others, such as families, campaigners, and teachers, comprehend it. The natural cue is a manual (hand-handed) expression agreed upon by the user (conventionally), recognised as limited in a certain group (esoteric), and utilised by a deaf person as a substitute for words (as opposed to body language). A formal gesture is a cue that is established consciously and has the same language structure as the spoken language of the society. More than 360 million of world population suffers from hearing and speech impairments. Sign language detection is a project implementation for designing a model in which web camera is used for capturing images of hand gestures which is done by open cv. After capturing images, labelling of images are required and then pre trained model SSD Mobile net v2 is used for sign recognition. Thus, an effective path of communication is developed between deaf and normal audience. Three steps must be completed in real time to solve our problem: 1. Obtaining footage of the user signing is step one (input). 2. Classifying each frame in the video to a sign. 3. Reconstructing and displaying the most likely Sign from classification scores (output). People with hearing impairments are left behind in online conferences, office sessions, schools. They usually use basic text chat to converse — a method less than optimal. With the growing adoption of telehealth, deaf people need to be able to communicate naturally with their healthcare network, colleagues and peers regardless of whether the second person knows sign language. Being able to achieve a uniform sign language translation machine is not a simple task, however, there are two common methods used to address this problem namely sensor based sign language recognition and Vision-based sign language recognition. Sensor based sign language recognition uses designs such as the robotic arm with a sensor, smart glove, golden glove for the conversion of ASL Sign language to speech. But the issue is that many people do not use it. Also, one must spend money to purchase such a glove, which is not easily available.

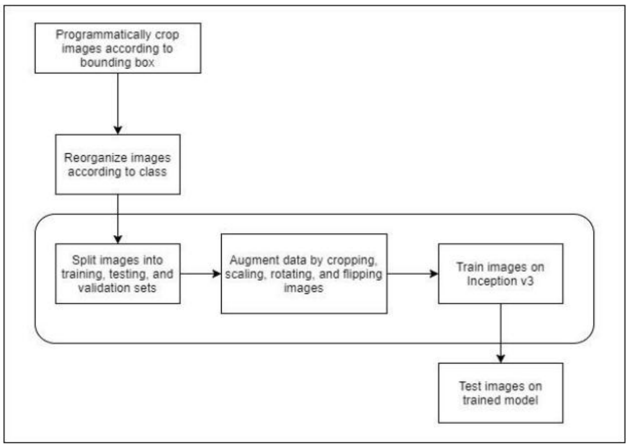
2. LITERATURE SURVEY

ASL recognition is not a new computer vision problem. Over the past two decades, researchers have used classifiers from a variety of categories that we grouped roughly into linear classifiers, neural networks and Bayesian networks. The first approach in relation to sign language recognition was by Bergh in 2011 [2]. Haar wavelets and database searching were employed to build a hand gesture recognition system. Although this system gives good results, it only considers six classes of gestures. Many types of research have been carried out on different sign languages from different countries. For example, a BSL recognition model, which understands finger-spelled signs from a video, was built [3]. As Initial, a histogram of gradients (HOG) was used to recognize letters, and then, the system used hidden Markov models (HMM) to recognize words. In another paper, a system was built to recognize sentences made of 3-5 words. Each word ought to be one of 19 signs in their thesaurus. Hidden Markov models have also been used on extracted features [4]. In 2011, a real time American Sign Language recognition model was proposed utilizing Gabor filter and random forest [5]. A dataset of color and depth images for 24 different alphabets was created. An accuracy of 75% was achieved utilizing both color and complexity images, and 69% using depth images only. Depth images were only used due to changes in the illumination and differences in the skin pigment. In 2013, a multilayered random forest was also used to build a real time ASL model. The system recognizes signs through applying random forest classifiers to the combined angle vector. An accuracy of 90% was achieved by testing one of the training images, and an accuracy of 70% was achieved for a new image. An American Sign Language alphabet recognition system was first built by localizing hand joint gestures using a hierarchical mode seeking and random forest method. An accuracy of 87% was achieved for the training, and accuracy of 57% when testing new images. In 2013, the Karhunen-Loeve Transform was used to classify gesture images of one hand into 10 classes [6]. These were translated and the axes were rotated to distinguish a modern coordinate model by applying edge detection, hand cropping, and skin filter techniques. (4) Another approach is to use deep learning techniques. This approach was used to build a model that recognizes hand gestures in a continual video stream utilizing DBN models [7]. An accuracy of over 99% was achieved. Another research used a deep learning technique, whereby a feed forward neural network was used to classify a sign. Many image pre-processing methods have been used, such as background subtraction, image normalization, image segmentation and contrast adjustment. All the works discussed above depended on the extraction of the hand before it is fed to a network. Kang, Tripathi & Nguyen built a real time sign finger spelling recognition system using CNNs from the depth map [8]. The authors collected 31,000 depth maps with 1,000 images for each class by using the Creative Senz3D camera. They had 31 various hand signs from 5 various persons. They utilized hand segmentation and assumed that the hand had to be near to the camera, which helped to make the bounding boxes in the same input size of 256, and 256. This work utilized only the depth map method by using the (Creative-Senz3D) camera, which is expensive and not available to everyone. Therefore, the proposed system cannot be implemented in a normal PC camera. This project differs from previous works in several ways. First, we have created our own dataset of 557 images. The dataset includes 5 gestures that are “Hello”, “Yes”, “No”, “Thank you” and “I Love You” and all the alphabets excluding J and Z because they are dynamic in nature. Second, we have created several user-friendly ways like website, QR code for users to access the system in order to communicate effectively.

3. PROPOSED MODEL

1. Collect images for deep learning using webcam and Opencv
2. Label images for sign language detection using Labelmg
3. Setup Tensorflow Object Detection pipeline configuration
4. Use transfer learning to train a deep learning model
5. Detect sign language in real time using OpenCV

In this project, we have first collected images using webcam and OpenCV. After collecting the images, the second task is labeling those images for Sign Language Detection. We labeled the collected images using LabelImg. After labeling the images, it is mandatory to set up Tensorflow Object Detection pipeline configuration. Now, after setting up the Tensorflow Object Detection configuration, we will use Transfer Learning to train a deep learning model. Now, we are able to detect sign language in real time using OpenCV.


Programmatically crop images according to bounding box
Reorganize images according to class
Split images into training, testing, and validation sets
Augment data by cropping, scaling, rotating, and flipping images
Train images on Inception v3
Test images on trained model

4. IMPLEMENTATION:

Techniques like recognising hand motion trajectories for individual signals and segmenting hands from the backdrop to predict and thread them into semantically acceptable and intelligible phrases are employed in sign language recognition. Furthermore, there are challenges in gesture recognition, motion modelling, motion analysis, pattern identification, and machine learning. Handcrafted parameters or parameters that are not manually specified are used in SLR models. The model's ability to categorise is impacted by the model's backdrop and environment, such as the room's illumination and the pace at which the motions are made. Because of the variations in views, the gesture seems unique in 2D space.

Systems that recognise gestures are divided into two categories: sensor-based and vision-based systems. Sensor-equipped devices acquire data in the sensor-based method.

The trajectory, position, and velocity of the hand are all parameters to consider. Vision-based approaches, on the other hand, leverage graphics from hand gesture video recordings. The phases involved in achieving sign language recognition are as follows: The camera for the sign language recognition system is as follows: The suggested sign language recognition system is based on a web camera collected frame on a laptop or PC. Image processing is carried out with the help of the OpenCV Python computer library. Taking Photographs: Under order to gain improved accuracy through a huge dataset, several photographs of distinct sign language signals were collected from various angles and in variable light conditions.

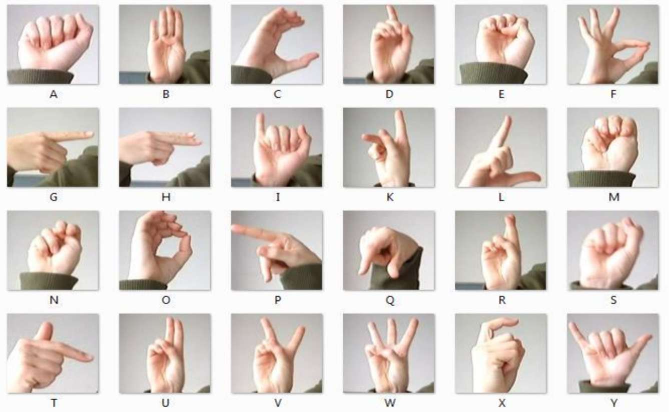
Segmentation: After the capturing portion is completed, a specific section from the full image is picked that contains the sign language symbol to be predicted. For the sign to be detected, bounding boxes are used. These boxes should be tightly packed around the picture region to be detected. The hand movements that were labelled were given specific names. The labelling was done with the LabelImg tool. Image selection for training and testing purposes

TF Record Creation: Multiple training and testing photos were used to produce record files.

There are two types of machine learning methods: supervised and unsupervised. Supervised machine learning is a method of teaching a system to recognise patterns in incoming data so that it predicts future data. Supervised machine learning applies a collection of known training data to labelled training data to infer a function.

Dataset: For this project, a user defined dataset is used. It is a collection of over 557 images. This dataset contains a

total of 5 symbols i.e.,Hello, Yes, No, I Love You and Thank You, which is quite useful while dealing with the real time application.



Convolutional Neural Network:

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning system that takes an input picture and give priority (learnable weights and biases) to various aspects/objects while also identifying them. Other classification algorithms need significantly more pre-processing than a ConvNet. ConvNets learns these filters/characteristics with adequate training, whereas simple techniques require filter hand-engineering.

ConvNets are multilayer artificial neural networks that handles input in two dimensions or three dimensions. Every layer in the network is made up of several planes, which can be 2D or 3D, and each plane is made up of a huge number of independent neurons, with layer neurons from neighboring layers linked but not from the same layer neurons.

A ConvNet captures the Spatial and Temporal aspects of an image by applying appropriate filters.

Furthermore, reducing the number of parameters involved and reusing weights resulted in the architecture performing better fitting to the picture collection. ConvNet's major goal is to make image processing easier by extracting relevant

Characteristics from images while preserving crucial information that is must for making accurate predictions. This is highly useful for developing an architecture that is not just capable of collecting and learning characteristics but also capable of handling massive volumes of data.

5. TOOLS USED:

TensorFlow is an open-source artificial intelligence programme that uses data flow graphs to generate models. It allows programmers to create large-scale neural networks with multiple layers. TensorFlow is mostly used for classification, perception, comprehension, discovery, prediction, and creation.

Object Detection API:It is an open source TensorFlow API to locate objects in an image and identify it.

Open CV:OpenCV is an open-source, highly optimised Python library targeted at tackling computer vision issues. It is primarily focused on real-time applications that provide computational efficiency for managing massive volumes of data. It processes photos and movies to recognise items, people, and even human handwriting

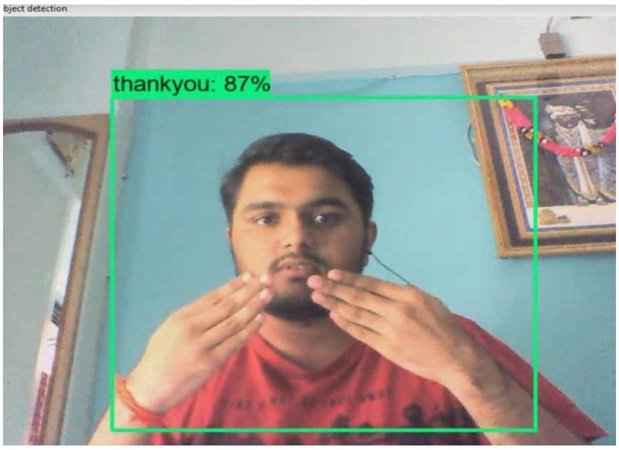
LabelImg: LabelImg is a graphical image annotation tool that labels the bounding boxes of objects in pictures.

6. MODEL ANALYSIS AND RESULT

The model was developed using a pre-trained model SSDmobile net v2 using a transfer learning technique.

The practice of applying a model that has been trained on one problem to a second, similar problem in some way is known as transfer learning. Transfer learning is a deep learning technique that involves training a neural network model on a similar problem to the one being addressed before being applied to the present problem. After that, one or more layers from the learnt model are used to train a new model on the problem of interest.

SSD Mobile net V2: The Mobile Net SSD model is a single-shot multibox detection (SSD) network that identifies objects by scanning the pixels of an image that are inside the bounding box coordinates and class probabilities. In contrast to standard residual models, the model's design is built on the concept of inverted residual structure, in which the residual block's input and output are narrow bottleneck layers. Intermediate layer nonlinearities are also removed, and lightweight depthwise convolution is used. This model is included in the TensorFlow object detection API.



7. APPLICATION AND FUTURE SCOPE:

Application: The collection can easily be enlarged and updated to match the user's needs, and it might be a big step in bridging the communication gap between the deaf and the dumb. Meetings held on a worldwide scale can become simpler to grasp for disabled people, and the worth of their hard work can be communicated utilising the sign detection technique. The concept may be used by anybody with a little grasp of technology, making it accessible to all. This technique might be implemented in elementary schools to teach children sign language at a young age.

Future scope: The adaptation of our concept to other sign languages, such as Indian and American sign languages. To improve the neural network's ability to recognise symbols, it will be further trained. Enhancement of the model's ability to recognise facial emotions.

8. CONCLUSION

The fundamental goal of a sign language detecting system is to provide a practical mechanism for normal and deaf individuals to communicate through hand gestures. The suggested method may be used with a webcam or any other in-built camera that detects and processes indicators for recognition. We may deduce from the model's findings that the suggested system produces reliable results under conditions of regulated light and intensity. Furthermore, new motions may be simply incorporated, and more photographs captured from various angles and frames will supply the model with greater accuracy. As a result, by expanding the dataset, the model may simply be scaled up to a vast size. Environmental issues such as low light intensity and an unmanaged backdrop are some of the model's limitations cause decrease in the accuracy of the detection. Therefore, we'll work next to overcome these flaws and also increase the dataset for more accurate results.

9. REFERENCES

[1]. Garcia, B., & Viesca, S. A. (2016). Real-time American sign language recognition with convolutional neural networks. Convolutional Neural Networks for Visual Recognition, 2, 225-232.

[2]. Van den Bergh, M., & Van Gool, L. (2011, January). Combining RGB and ToF cameras for real-time 3D hand gesture interaction. In 2011 IEEE workshop on applications of computer vision (WACV) (pp. 66-72). IEEE.

[3]. Liwicki, S., & Everingham, M. (2009, June). Automatic recognition of fingerspelled words in british sign language. In 2009 IEEE computer society conference on computer vision and pattern recognition workshops (pp. 50-57). IEEE.

[4]. Zafrulla, Z., Brashear, H., Starner, T., Hamilton, H., & Presti, P. (2011, November). American sign language recognition with the kinect. In Proceedings of the 13th international conference on multimodal interfaces (pp. 279-286).

[5]. Pugeault, N., & Bowden, R. (2011, November). Spelling it out: Real-time ASL fingerspelling recognition. In 2011 IEEE International conference on computer vision workshops (ICCV workshops) (pp. 1114-1119). IEEE.

[6]. Suk, H. I., Sin, B. K., & Lee, S. W. (2010). Hand gesture recognition based on dynamic Bayesian network framework. Pattern recognition, 43(9), 3059-3072.

[7]. Kang, B., Tripathi, S., & Nguyen, T. Q. (2015, November). Real-time sign language fingerspelling recognition using convolutional neural networks from depth map. In 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR) (pp. 136-140). IEEE.

[8]. Singha, J., & Das, K. (2013). Hand gesture recognition based on Karhunen-Loeve transform. arXiv preprint arXiv:1306.2599.