SIGN LANGUAGE DETECTION USING DEEP LEARNING

Bharath Gajula
Department of ECE
B.M.S College of Engineering
Bangalore, India
bharath.ec18@bmsce.ac.in

***Abstract*— Many individuals who are not familiar with sign language find it difficult to communicate without an interpreter. Therefore, a device that transcribes sign language symbols into plain text can assist with real-time communication, and can also provide people with immersive training to learn sign language. Hand gestures are used in sign language to communicate meaning.**

A lot of research has been done in the American Sign Language (ASL) area. The key challenges that are present little research being done in ASL have been the lack of common datasets, occluded features and variance in the language with locality. Our project is an attempt to research the challenges of Indian Sign Language character classification (ASL) and to gather a dataset and then use different features.

The project aims to create a model of machine learning that will be able to identify the different hand gestures used in sign language for finger spelling. Classification machine learning algorithms are trained using a collection of image data in this user independent model and testing is performed on a set of data. On the datasets different algorithms are applied.

I. INTRODUCTION

American Sign Language is a predominant Sign Language Since the only disability Deaf and Dumb people have to communicate related and since they could not speak language. Communication is that method of exchange the thoughts and message in varied ways that like to speech, Signals, behavior and visuals. D&M folks create use of their hand to precise totally different gestures to precise their concepts with others as shown in the figure [1.1.1].

Gestures are the non-verbally changed message and these gestures to precise their concepts with others. Gestures are the non-verbally changed message and these gestures are understood with vision. Minimizing the articulation gap among D&M and Non- D&M folks turns into a need to foam a bound effective speech among all language translation is among the foremost growing lines of analysis and it perm it's the uttermost one among the foremost growing lines of analysis and its permits the uttermost natural manner of communication for those with hearing impairments.

A hand gesture recognition system offers a chance for deal folks to speak with vocal human while not the requirement of AN interpreter. The system is made for the machine-driven conversion of sign language into the matter content and its speech. The goal of this project to create a neural network ready using CNN to classify that letter of the sign which in the format of (ASL) alphabet is being signed, given a picture of language hand a attain table sign translator, which might take communication in sing and translate them into written and oral language. Such a translator would greatly lower the barrier and oral language. Mute people to be ready to higher communicate with other in day-to-day interactions. There are giant barriers that deeply have an effect on life quality steam from the communication disconnected between the deaf and therefore the hearing

The blind may suffer additional delay compared to a normal person because of the limited transportation choices. The most commonly used mode of transport for the visually impaired is public transport, which is considered one of the most important modes of transport in many countries.

Figure 1.1.1: Communication between the deaf people

II. METHADOLOGY AND IMPLEMENTATION

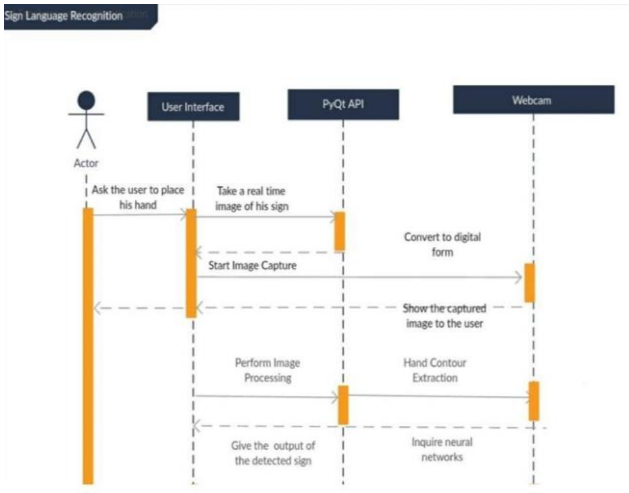


Figure 3.1.1: Sequence diagram of the model.

In the figure [3.1.1] the user is asked to place hand in front of the webcam. The user interface takes a real time image of the gesture shown and then it will be converted into digital form. The captured image is then shown to the user. Then image

preprocessing and hand contour extraction is performed. The given image image is tested with the trained dataset to give the output of the detected gesture.

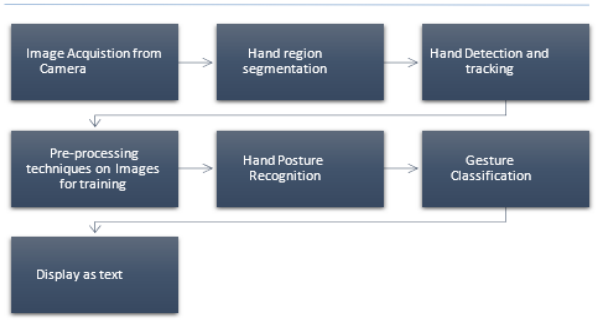


Figure 3.1.2: Flow chart of the model

The user is instructed to place their hand in front of the webcam as seen in figure [3.1.2]. The user interface captures a real-time image of the gesture being used, which is then digitally transformed. The user is then presented the captured image. Then, hand contour extraction and picture preprocessing are carried out. To determine the output of the gesture that was recognized, the provided image was evaluated against the training dataset.

III. PROJECT FLOW

A.Data Preprocessing

RGB to HSV

The technique used is converting RGB to HSV.

The hue, saturation, value (HSV) color space is designed to act close to the perception of a human eye. It is referred as an approximately uniform perceptual color space. The V channel represents luminance, and the other channels represent chrominance. The reason we use HSV color space for color detection/thresholding over RGB/ BGR is that HSV is more robust towards external lighting changes.

This means that in cases of minor changes in external lighting (such as pale shadows.) Hue values vary relatively lesser than RGB values. The R,G,B values are divided by 255 to change the range from 0..255 to 0..1: R' = R/255, G' = G/255, B' = B/255

Cmax = max(R', G', B'),

Cmin = min(R', G', B'),

Δ = Cmax – Cmin

$$H = \begin{cases} 0^\circ & \Delta = 0 \\ 60^\circ \times \left(\frac{G'-B'}{\Delta} mod 6\right) & , C_{max} = R' \\ 60^\circ \times \left(\frac{B'-R'}{\Delta} + 2\right) & , C_{max} = G' \\ 60^\circ \times \left(\frac{R'-G'}{\Delta} + 4\right) & , C_{max} = B' \end{cases} \qquad S = \begin{cases} 0 & , C_{max} = 0 \\ \frac{\Delta}{C_{max}} & , C_{max} \neq 0 \end{cases}$$ --------Equation 1

Hue calculation, Saturation calculation

Value calculation: V = Cmax

The recognition process includes segmentation as an important step, but the segmentation results are not greatly evaluated. These recognition methods are based on several approaches that could also be used. The approaches are distinguished as boundary enhancement, clustering, smoothening, edge detection.

This work employs an enhancement method with k means clustering method to eliminate the over segmentation and false edges during the segmentation process using sobel edge detection.

It is a widely used algorithm for image segmentation because of its ability to cluster huge data points very quickly. The function of clustering is to group image pixels where the related feature vectors produce the similar images.

Hence the next technique used is K-means clustering, following are the steps for this algorithm.

a) We first load the input images in step 1.

b) In step 2 we commute the RGB image into array of image for transformation.

c) In step 3 RGB images are combination of primary colours (Red, Green, Blue).

d) The next step RGB image feature Pixel Counting technique.

e) The RGB images space consists of a radiance layer, chromaticity-layer indicating where colour falls along the red-green axis and the images with high pixel are clustered to form a high contrast and clear image.

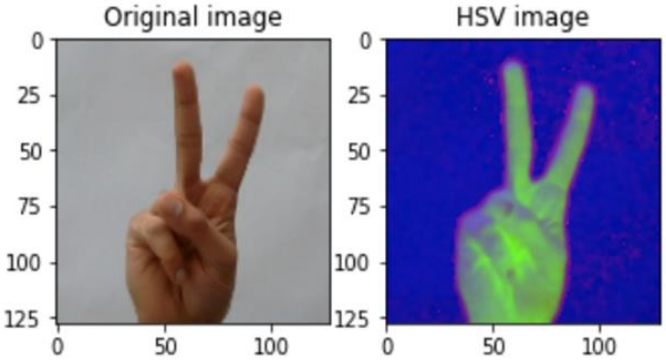


Figure 3.2.1.1: RGB to HSV converted image.

HSV Mask

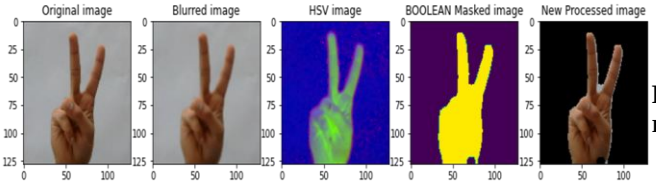
Pre-processing of input images is done to improve the quality of image and to remove the undesired distortion from the image. Clipping of the image is performed to get the interested image region and then image smoothing is done using the smoothing filter. To increase the contrast Image enhancement is also done. The mask segmentation process is based on various features found in the image. Some of these features may be colour information, boundaries or segments of an image.

We use Genetic algorithm for colour image segmentation. First, conversion of the RGB images into HSV colour space for segmentation is performed. After completion of this process, to generate a colour co-occurrence matrix, each pixel map is used, which results in three colour co-occurrence matrices, one for each of H, S, V.

Masking of pixels: Masking means that the pixel value in an image is set to zero or to some other value. In this technique, we computed a threshold value that is used for these pixels.

Then in the following way mostly green pixels are masked: if pixel intensity is less than the pre-computed threshold value then zero value is assigned to the red, green and blue components of this pixel. After masking, pixels with zero values are discarded. In masking, the portion of the Image is identified by H and S plane value and value of 1 which is allocated to a particular portion. Rest of the regions 0 value is given.

As a result, a binary image contains only zeros and ones. Thus, the image of hand sign can be extracted.



Figu re

3.2.1.2: HSV image segmentation on hand image.

Canny's edge detection technique:

Canny edge detection is a technique to extract useful structural information from different vision objects and dramatically reduce the amount of data to be processed. It has been widely applied in various computer vision systems.

The canny operator works in a multi-stage process. Then a simple 2-D first derivative operator is applied to the smoothed image to highlight regions of the image with high first spatial derivatives. Gradient is the first – order derivatives of image for each direction. Which is non maximal suppression. The gradient can be computed using central difference.

$\partial X(x,y)=[(x+1,y)-(x-1,y)]/2$ -------- Equation 2

$\partial Y(x,y)=[(x,y+1)-(x,y\_1)]/2$ -------- Equation 3

Magnitude of horizontal and vertical gradient is used for non-maximal suppression process. The magnitude can be computed by:

Magnitude= $(\partial X(x) * \partial X(y) + \partial Y(x) * \partial Y(y))$ -------- Equation 4

The following shows the canny edge detection algorithm steps:

The algorithm runs in 5 separate steps,

1. Smoothing: Blurring of the image to remove noise.

2. Finding gradients: The edges should be marked where the gradients of the image have large magnitudes.

3. Non-maximum suppression: Only local maxima should be marked as edges.

4. Double thresholding: Potential edges are determined by thresholding.

5. Edge tracking by hysteresis: Final edges are determined by suppressing all edges that are not connected to a very certain (strong) edge.

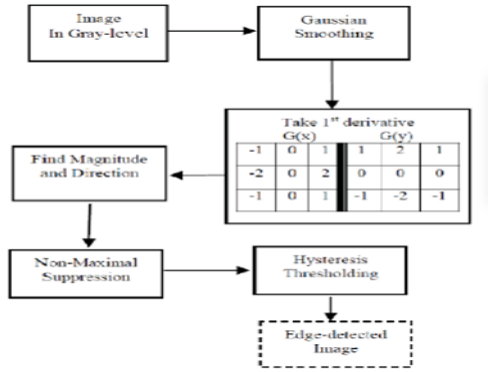


Figure 3.2.1.3: Flowchart of pre-processing.

Conversion from RGB to Grayscale Image

Three channel RGB image is then converted to one channel Gray Scale image as shown in Figure [3.2.1.4]. Also, the background noise (or cluster defining the background image) is removed by Histogram analysis of clustered image.

Three channel RGB image is then converted to one channel Gray Scale image as shown in Figure below.

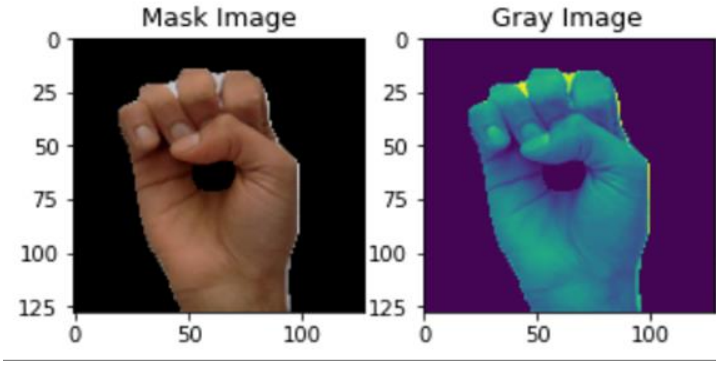


Figure 3.2.1.4: RGB converted to grey image.

Noise Reduction

Since the mathematics involved behind the scene are mainly based on derivatives (Gradient calculation), edge detection results are highly sensitive to image noise [3.2.1.4]. One way to get rid of the noise on the image, is by applying Gaussian blur to smooth it. To do so, image convolution technique is applied with a Gaussian Kernel (3x3, 5x5, 7x7 etc.…). The kernel size depends on the expected blurring effect. Basically, the smallest the kernel, the less visible is the blur. The equation for a Gaussian filter kernel of size (2k+1) × (2k+1) is given by:

$H = 1\ 2\pi\sigma 2\ \exp(-(i-(k+1))\ 2 + (j-(k+1))\ 2\ 2.\sigma 2\ )$; $1<=i, j<= (2k+1)$ --------Equation 5

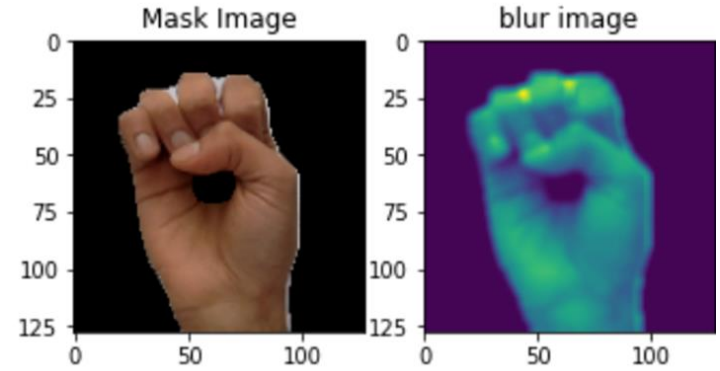


Figure 3.2.1.5: Gaussian blurred image.

Non-Maximum Suppression

Here we check basically if the pixels on the same direction are more or less intense than the ones being processed. When the pixel (i, j) is being processed, and the pixels on the same direction (i, j-1) and (i, j+1), If one those two pixels are more intense than the one being processed, then only the more intense one is kept.

If say Pixel (i, j-1) seems to be more intense, the intensity value of the current pixel (i, j) is set to 0. If there are no pixels in the edge direction having more intense values, then the value of the current pixel is kept as it is.

As evident from Figure sobel magnitude figure, Magnitude image consists of all strong and weak edges, Also, some of the edges are thicker while some are thinner which results into loss of clarity in important part of image.

Gradient calculation beside of magnitude also gives angle orientation of each pixel. We can use this information for thinning out the thicker ones in the above image.

Double Threshold

The double threshold step aims at identifying 3 kinds of pixels: strong, weak, and nonrelevant:

- Strong pixels are pixels that have an intensity so high that we are sure they contribute to the final edge.
- Weak pixels are pixels that have an intensity value that is not enough to be considered as strong ones, but yet not small enough to be considered as nonrelevant for the edge detection.
- Other pixels are considered as non-relevant for the edge.

So, the double thresholds hold for:

- High threshold is used to identify the strong pixels (intensity higher than the high threshold)
- Low threshold is used to identify the non-relevant pixels (intensity lower than the low threshold)
- All pixels having intensity between both thresholds are flagged as weak and the Hysteresis mechanism (next step) will help us identify the ones that could be considered as strong and the ones that are considered as non-relevant. The result of this step is an image with only 2-pixel intensity values (strong and weak).

From the Non-Maximum Separation. figure F, we notice that still some pixels are brighter than others. The result of double thresholding is an image with only two-pixel values (strong and weak). By edge tracking 16 by hysteresis weak pixels are transformed into strong edges if and only if it is surrounded by at least one strong pixel. It is evident from the Figure that method proposed in this project gives better edge detected image for further analysis by deep learning.

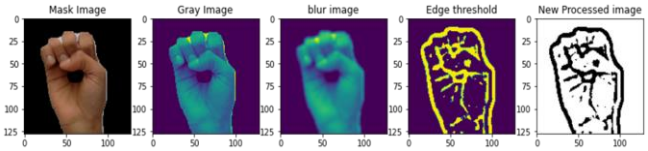


Figure 3.2.1.6: Threshold on images for edge detection.

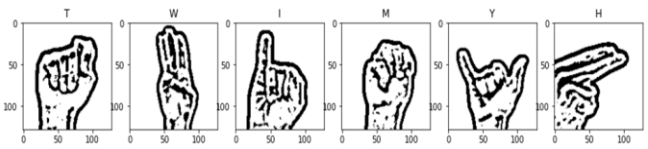


Figure 3.2.1.7: Pre-processed images after applying threshold.

Gaussian filter:

Gaussian filter is used as a pre-processing technique to make the image smooth and eliminate all the irrelevant noise. Intensity is analyzed and Non-Maximum suppression is implemented to remove false edges. For a better pre-processed image data, double Thresholding is implemented to consider only the strong edges in the images. All the weak edges are finally removed and only the strong edges are considered for the further phases. Non-Maximum suppression is implemented to remove false edges.

A Gaussian Filter is a low pass filter used for reducing noise (high frequency components) and blurring regions of an image. The filter is implemented as an Odd Sized Symmetric Kernel (DIP version of a Matrix) which is passed through each pixel of the Region of Interest to get the desired effect. The kernel is not hard towards drastic color changed (edges) due to it the pixels towards the center of the kernel having more weightage towards the final value then the periphery. A Gaussian Filter could be considered as an approximation of the Gaussian Function (mathematics).

B. Model Building

CNN Algorithm:

CNN algorithm is used for classification purposes.

CNNs are composed of four types of layers:

- Input layer
- Convolutional layer
- Pooling layer
- Fully-connected layers

When these layers are piled, a CNN architecture will be created. The input layer will contain the image's pixel values. The convolutional layer determines the scalar product of the weights of each pixel and determines the output neurons. Introduction to Convolutional Neural Networks aims to add the rectified linear unit (usually abbreviated to ReLu) to the activation output provided by the previous layer with an 'elementary' activation function such as sigmoid.The pooling layer applies a function so that all the negative values are replaced with zero. The fully connected layer is the layer where actual classification takes place.Class scores are generated and the shrinked image with class scores is converted into a list.

How CNN Works?

An input image may be deformed. These deformed images should also be classified by classifier because they are also the images to be predicted. In normal technique, when both the images are compared, the image classifier will not be able to predict the deformed image. A computer understands an image using numbers at each pixels. For example, in a binary image, black pixel is considered with a value 1 and while pixel will have -1 value. CNN compares these images piece by piece. By finding rough features in roughly the same positions in two images, CNN gets a lot better in seeing similarity than whole image matching schemes.

VGG 16:

VGG16 is a convolution neural net (CNN) architecture as shown in the figure [3.2.2.1] which was used to win ILSVR(ImageNet) competition in 2014. It is considered to be one of the excellent vision model architectures till date. Most unique thing about VGG16 is that instead of having a large number of hyper-parameters they focused on having convolution layers of 3x3 filter with a stride 1 and always used same padding and maxpool layer of 2x2 filter of stride 2. It follows this arrangement of convolution and max pool layers consistently throughout the whole architecture. In the end it has 2 FC(fully connected layers) followed by a softmax for output. The 16 in VGG16 refers to it has 16 layers that have weights. This network is a pretty large network and it has about 138 million (approx.) parameters.

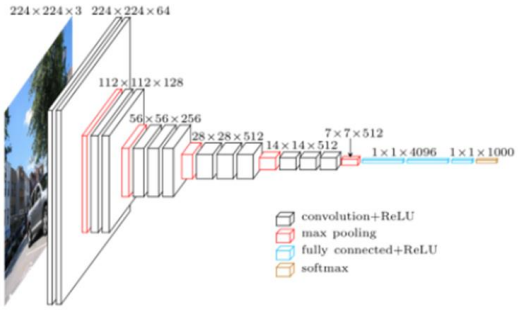


Figure 3.2.2.1: Architecture of VGG 16

ResNet50

ResNet-50 is a convolutional neural network that is 50 layers deep as shown in the figure [3.2.2.2]. ResNet, short for Residual Networks is a classic neural network used as a backbone for many computer vision tasks. The fundamental breakthrough with ResNet was it allowed us to train extremely deep neural networks with 150+layers.

Convolutional Neural Networks have a major disadvantage — 'Vanishing Gradient Problem'. During backpropagation, the value of gradient decreases significantly, thus hardly any change comes to weights. To overcome this, ResNet is used. It makes use of "SKIP CONNECTION".

SKIP CONNECTION is a direct connection that skips over some layers of the model. The output is not the same due to this skip connection. Without the skip connection, input 'X gets multiplied by the weights of the layer followed by adding a bias term.

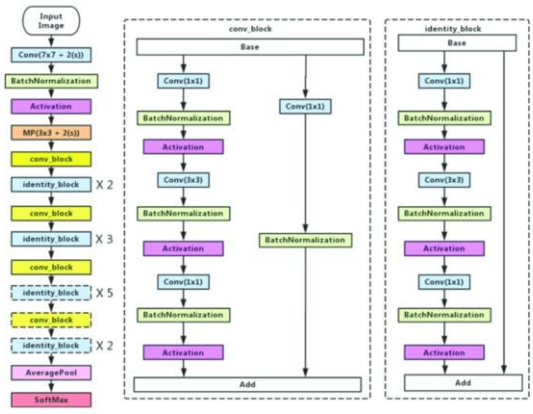


Figure 3.2.2.2: Architecture of ResNet50

YOLOV5 Architecture

YOLO v5 is a single-stage object detector as shown in the figure [3.2.2.3], it has three important parts like any other single stage object detector.

- Model Backbone
- Model Neck
- Model Head

Model Backbone is mainly used to extract important features from the given input image. In YOLO v5 the CSP (Cross Stage Partial Network) are used as a backbone to extract rich in informative features from an input image. Model Neck is mainly used to generate feature pyramids. Feature pyramids help models to generalized well on object scaling. It helps to identify the same object with different sizes and scales. Feature pyramids are very useful and help models to perform well on unseen data. There are other models that use different types of feature pyramid techniques like FPN, BiFPN etc.

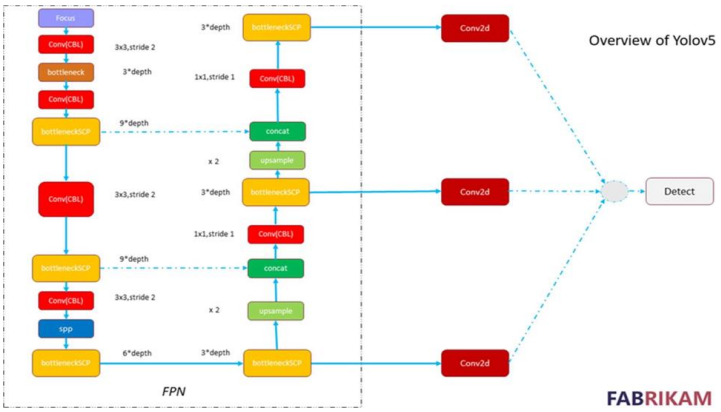


Figure 3.2.2.3: YOLOV5 architecture.

Transfer Learning

Transfer Learning is a machine learning technique where models are trained on (usually) larger data sets and refactored to fit more specific or niche data. This is done by recycling a portion of the weights from the pre-trained model and reinitializing or otherwise altering weights at shallower layers. The most basic example of this would be a fully trained network whose final classification layer weights have been reinitialized to be able to classify some new set of data. The primary benefits of such a technique are its less demanding time and data requirements. However, the challenge in transfer learning stems from the differences between the original data used to train and the new data being classified. Larger differences in these data sets often require re-initializing or increasing learning rates for deeper layers in the net.

Training:

For training a list of training images has to be built from the file system. The sub folders need to be analyzed in the image directory and split into stable training, testing, and validation sets. The next step is to return a data structure describing the lists of images for each label and their paths. The training step then creates a graph from the saved file and returns a Graph object holding the trained Inception network, and various tensors that will be manipulated. Then the model tar file is downloaded and extracted. If the pre trained model to be used doesn't already exist, it is downloaded from the TensorFlow.org website and unpacked it into a directory. The given list of floats is written to a binary file. Running the image through simple distortions like cropping, scaling, and flips during training, can improve the results. These reflect the kinds of variances expected in the actual world, and can help the model cope better with natural data. A network of operations has to be built to apply the specified parameters to an image in this step.

Cropping: Cropping is done by randomly placing a bounding box in the full image. The cropping parameter determines how big that box is in comparison to the input image. If it's 0, the box will be the same size as the input, with no cropping. The crop box will be half the width and height of the input if the value is 50%.

Scaling: Scaling is similar to cropping, only difference is that the bounding box is always centered and the size of the bounding box fluctuates randomly within the given range. If the scale

percentage is zero, for example, the bounding box will be the same size as the input and no scaling will be done. If you set it to 50%, the bounding box will be somewhere between half the width and height and full size.

The tensor flow library downloads the inception model. This model has a set of rules for machine learning. These include read input files, index them and convert them to a format such that machine learning can be applied. The captured image may vary in width and height. These are converted o a fixed height and width of small dimensions so that it can be stored and trained more easily. Gaussian filters are applied to reduce the noise in the dataset. Then it is passed to tensor flow, which is the image processing library. Seventy percent of the files present in the dataset are used for training.

Testing:

We need to retrain the top layer to recognize our new classes, so we write a function that adds the appropriate operations to the graph, as well as some variables to store the weights, and then sets up all of the gradients for the backward pass. We do a final test evaluation on some new images we haven't used before once all of the training is completed. Testing is done in a similar way with the same properties, but unlike training, it uses only one file and compares the values with the values of the model created. Thirty percent of the files present in the dataset are used for testing.

C: Real Time Detection:

Sign to text: Tensor Flow is the main library used in conversion from sign to text as shown in the figure [3.2.3.1]

T**ext to voice:** gTTS (Google Text to Speech) was used to convert the given text into voice. There are several APIs available to convert text to speech in Python. One of such APIs is the Google Text to Speech API commonly known as the gTTS API. gTTS is a very easy to use tool which converts the text entered, into audio which can be saved as a mp3 file. The gTTS API supports several languages including English, Hindi, Tamil, French, German and many more. The speech can be delivered in any one of the two available audio speeds, fast or slow. However, as of the latest update, it is not possible to change the voice of the generated audio.

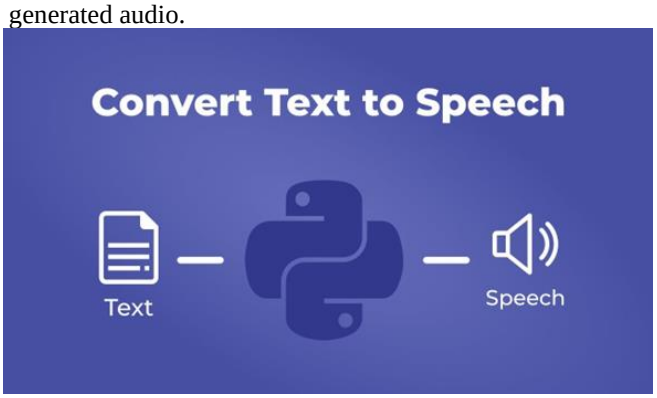


Figure 3.2.3.1: Google text to speech.

IV. SYSTEM ARCHITECTURE

In the figure [3.3.1], the user is told to hold their hand in front of the webcam. Real-time video of the gesture is recorded by the user interface and afterwards converted into digital form. The acquired image is then displayed to the user. Then, manual contour extraction and image preparation are completed. The submitted image is tested using the training dataset, which also generates the output of the recognized gesture.

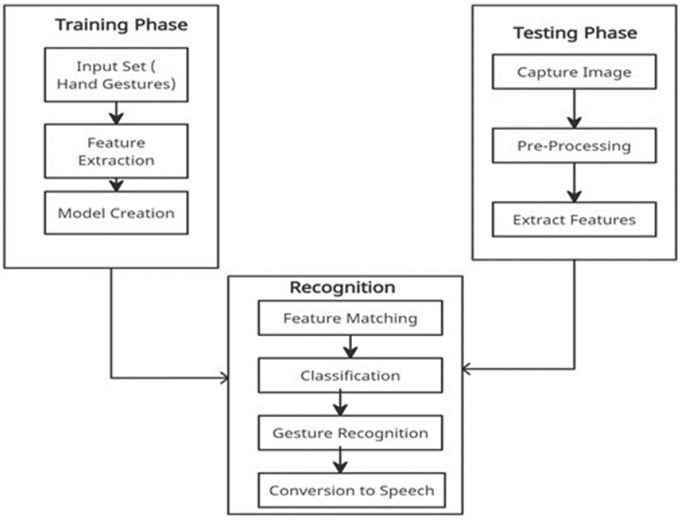


Figure 3.3.1: System architecture.

V.RESULTS AND DISCUSSIONS

CNN MODEL

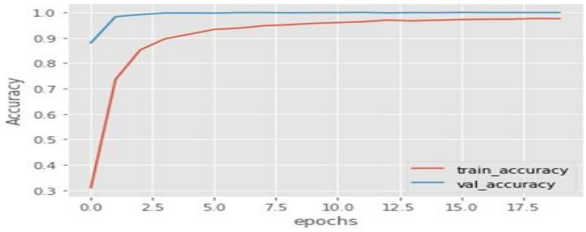


Figure 4.1.1: Accuracy Evaluation of CNN

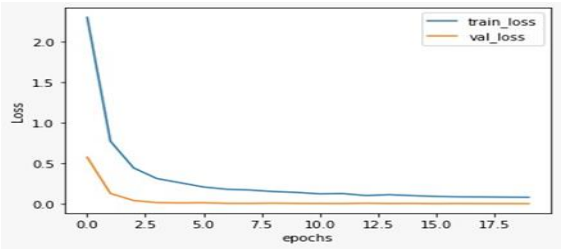


Figure 4.1.2: Loss Evaluation of CNN

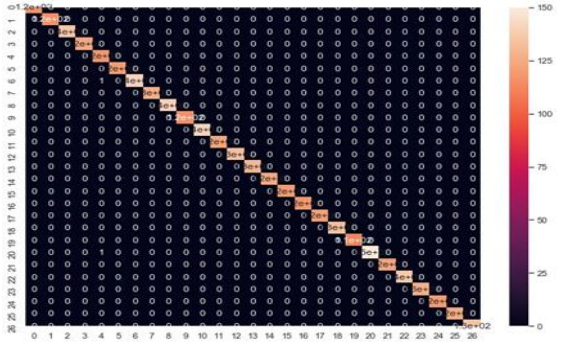


Figure 4.1.3: Confusion Matrix of CNN

As shown the above figure [4.1.1] and [4.1.2], as the epochs increase accuracy is increasing and loss is decreasing. At epoch 2.5 the accuracy knee bend has occurred and up to the epoch 5 curve is smooth and out. As we see in the figure [4.1.3] the confusion matrix is given. we can see the model CNN shows the predicted label vs actual label classification with accuracy of CNN this shows us the model is good with high score and accuracy was able to predict and classify alphabet label appropriately.

VGG 16

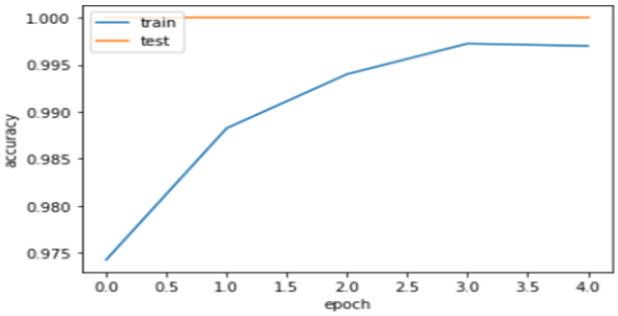


Figure 4.2.1: Accuracy Evaluation of VGG 16

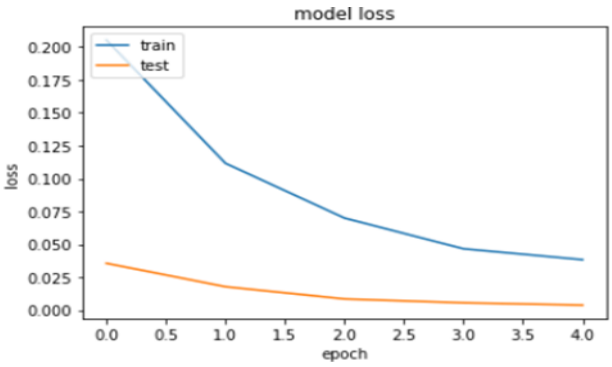


Figure 4.2.2: Loss Evaluation of VGG16

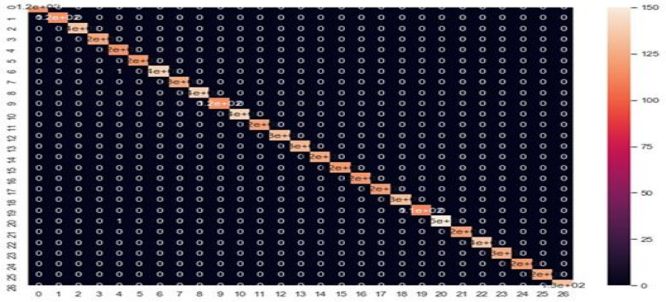


Figure 4.2.3: Confusion Matrix of VGG 16

As shown the above figure [4.2.1] and figure [4.2.2], as the epochs increase accuracy is increasing and loss is decreasing. At epoch 1.0 the accuracy knee bend has occurred and upto the epoch 3.0 curve is smooth and out. As we see in the figure [4.2.3] the confusion matrix is given. we can see the model VGG16 shows the predicted label vs actual label classification with accuracy of VG16 this shows us the model is good with high score and accuracy was able to predict and classify alphabet label appropriately.

RESNET 50

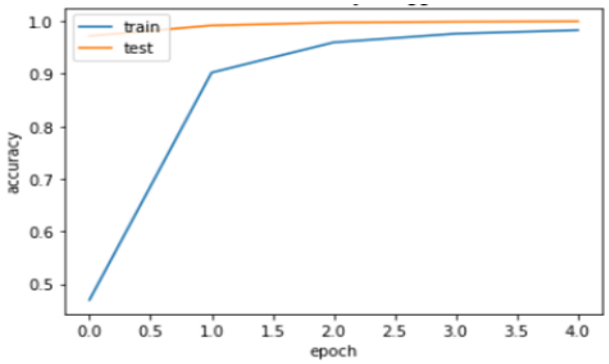


Figure 4.3.1: Accuracy Evaluation of ResNet50

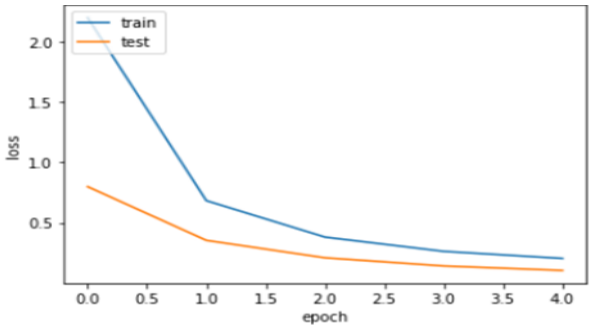


Figure 4.3.2: Loss Evaluation of ResNet50

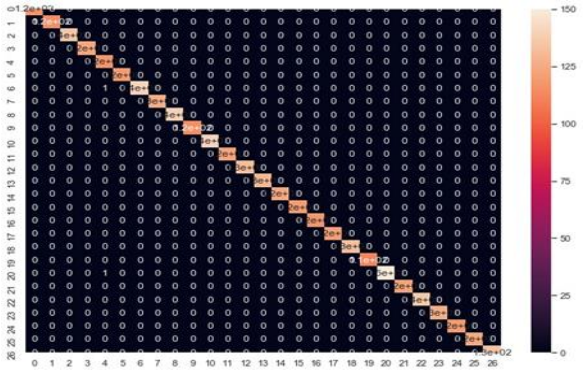


Figure 4.3.3: Confusion Matrix of ResNet50

As shown the above figure [4.3.1] and [4.3.2], as the epochs increase accuracy is increasing and loss is decreasing. At epoch 1.0 the accuracy knee bend has occurred and upto the epoch 2.0 curve is smooth and out. As we see in the figure [4.3.3] the confusion matrix is given. we can see the model ResNet50 shows the predicted label vs actual label classification with accuracy of ResNet50 this shows us the model is good with high score and accuracy was able to predict and classify alphabet label appropriately.

IV. CONCLUSION

Machine translation is a very hot research subject in the field of natural language processing at present. Machine learning helps to train a human brain-like translation system. CNN, VGG 16 and ResNet50 are capable of recognizing and translating sign language into text and speech. In text to voice processing, Google Text to Speech offers better performance. Sign language recognition is a crucial communication aid for the speaking and hearing impaired. This instrument can help bridge the gap between individuals who are natural and deaf/dumb. We use the several models to classify sign language characters, including alphabets and numerals, with exceptional precision.

Instead of only characters, we construct a real-time application that can define the sign language, including terms and phrases

REFERENCES

[1] Sanil Jain , K.V.Sameer Raja . “Indian Sign Language Character Recognition”.

[2] Muttu Mariappan H , Dr. Gomathi V . “Real Time Recognition of Indian Sign Language” Conference: 2019 International Conference on Computational Intelligence in Data Science (ICCIDS).

[3] Youngwook Kim, Brian Toomajian. “Hand Gesture Recognition Using Micro-Doppler Signatures With Convolutional Neural Network” California State University , Fresno , CA 93740, USA , November 18,2016.

[4] Yogeshwar I Rokade, Prashant M. Jadav “Indian Sign Language Recognition System”, International Journal of Engineering and Technology , July 2017

[5] Abhishek Dudhal,Heramb Mathkar, Abhishek Jain,Omkar Kadam and Mahesh Shirole ,“Hybrid SIFT feature Extraction Approach for Indian Sign Language

[6] Neel Kamal Bhagat, Vishnusai Y and Rathna .G N . “Indian Sign Language Gesture Recognition using Image Processing and Deep Learning”, Published August 2019.

[7] Kumud Tripathi ,Neha Baranwal and G.C.Nandi.”Continuous Indian Sign Language Gesture Recognition and Sentence Formation”. Indian Institute of Technology , Allahabad ,2015

[8] Joyeeta Singha, Karen Das.”Recognition of Indian Sign Language in Live Video”.DBCET ,Assam Don Bosco University , Guwahati , Assam

[9] Purva A.Nanivadekar, Dr.Vaishali Kulkarni.”Indian Sign Language Recognition:Database Creation,Hand Tracking and Segmentation”. MPSTME,NMIMS ,Mumbai,India.

[10] Archana S.Ghotkar, Gajanan K.Kharate.”Dynamic Hand Gesture Recognition and Novel Sentence Interpretation Algorithm for Indian Sign Language Using Microsoft Kinect Sensor".Journal Of Pattern Recognition, July 10,2015.

[11] Zhi-hua Chen, Jung-Tae Kim, Jianning Liang, Jing Zhang, and Yu-Bo Yuan.”Real-Time Hand Gesture Recognition Using Finger Segmentation”.Published 25 June 2014.

[12] Purva C.Badhe,Vaishali Kulkarni.”Indian Sign Language Translator Using Gesture Recognition Algorithm”.2015 IEEE International Conference.

[13] Munir Oudah, Ali Al-Naji and Javaan Chahl , “Hand Gesture Recognition Based on Computer Vision : A Review TEchniques”, Journal of Imaging, Published 23 July 2020.

[14] H.S. Nagendraswamy and B.M.Chethan Kumara , “LBVP for Recognition of Sign Language at Sentence Level: An Approach BAsed on Symbolic Representation”

[15] Mahesh Kumar N B. “Conversion of Sign Language into Text”. International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Published Number 9 2018.

[16] Bhargav Hegde, Dayananda P, Mahesh Hegde, Chetan C, “Deep Learning Technique for Detecting NSCLC”, International Journal of Recent Technology and Engineering (IJRTE), Volume-8 Issue-3, September 2017

[17] Sakshi Goyal, Ishita Sharma, Shanu Sharma .“Sign Language Recognition System For Deaf And Dumb People International “Journal of Engineering Research & Technology (IJERT), April – 2013

[18] Geetha M, Manjusha Uc.”A Vision Based Recognition of Indian Sign Language Alphabets and Numerals Using B-Spline Approximation”Amrita Vishwa Vidyapeetham. March 2012.