Sign Language Detection

A Synopsis Submitted

in Partial Fulfillment of the Requirements for the Degree of

# Bachelor In Technology

in

# Computer Science and Engineering

By

# Aniket Kumar 1903480100018

# Ankit Srivastava 1903480100020

# Himanshu Singh 1903480100054

# Marshal 1903480100063

Computer Science & Engineering

PSIT COLLEGE OF ENGINEERING AKTU, LUCKNOW, INDIA

# April, 2021-22

Synopsis

# Introduction

Sign language is the only tool of communication for the person who is not able to speak and hear anything. Sign language is a boon for the physically challenged people to express their thoughts and emotion. In this work, a novel scheme of sign language recognition has been proposed for identifying the alphabets and gestures in sign language. With the help of computer vision and neural networks we can detect the signs and give the respective text output.

The process of converting the signs and gestures shown by the user into text is called sign language recognition. It bridges the communication gap between people who cannot speak and the general public. Image processing algorithms along with neural networks is used to map the gesture to appropriate text in the training data and hence raw images/videos are converted into respective text that can be read and understood.

The base model will be able to recognize the sign language, which is used by deaf and dumb in order to express or convey their thoughts to others. The other layer of this model will be able to take input from user of new hand gestures in real time minimum of 5-8 inputs and then it will be able to recognize that new gesture created explicitly by the user, using this we will be able to create our own sign language and train the model on it with ease. After the model creation it will be deployed into an android application.

# Motivation

The 2011 Indian census cites roughly 1.3 million people with “hearing impairment”. In contrast to that numbers from India’s National Association of the Deaf estimates that 18 million people –roughly 1 per cent of Indian population are deaf. These statistics formed the motivation for our project. As these speech impairment and deaf people need a proper channel to communicate with normal people there is a need for a system. Not all normal people can understand sign language of impaired people. Our project hence is aimed at converting the sign language gestures into text that is readable for normal people.

# Related work

**3.1 A Survey of Hand Gesture Recognition Methods in Sign Language Recognition**

Sign Language Recognition (SLR) system, which is required to recognize sign languages, has been widely studied for years. The studies are based on various input sensors, gesture segmentation, extraction of features and classification methods. This paper aims to analyze and compare the methods employed in the SLR systems, classifications methods that have been used, and suggests the most promising method for future research. Due to recent advancement in classification methods, many of the recent proposed works mainly contribute on the classification methods, such as hybrid method and Deep Learning. This paper focuses on the classification methods used in prior Sign Language Recognition system. Based on our review, HMM based approaches have been explored extensively in prior research, including its modifications. This study is based on various input sensors, gesture segmentation, extraction of features and classification methods. This paper aims to analyze and compare the methods employed in the SLR systems, classifications methods that have been used, and suggests the most reliable method for future research. Due to recent advancement in classification methods, many of the recently proposed works mainly contribute to the classification methods, such as hybrid method and Deep Learning. Based on our review, HMM-based approaches have been explored extensively in prior research, including its modifications. Hybrid CNN-HMM and fully Deep Learning approaches have shown promising results and offer opportunities for further exploration.

**3.2 A System for Recognition of Indian Sign Language for Deaf People using Otsu’s Algorithm**

In this paper we proposed some methods, through which the recognition of the signs becomes easy for peoples while communication. And the result of those symbols signs will be converted into the text. In this project, we are capturing hand gestures through webcam and convert this image into gray scale image. The segmentation of gray scale image of a hand gesture is performed using Otsu thresholding algorithm. Total image level is divided into two classes one is hand and other is background. The optimal threshold value is determined by computing the ratio between class variance and total class variance. To find the boundary of hand gesture in image Canny edge detection technique is used. In Canny edge detection we used edge based segmentation and threshold based segmentation. Then Otsu’s algorithm is used because of its simple calculation and stability. This algorithm fails, when the global distribution of the target and background vary widely.

**3.3 Intelligent Sign Language Recognition Using Image Processing**

Computer recognition of sign language is an important research problem for enabling communication with hearing impaired people. This project introduces an efficient and fast algorithm for identification of the number of fingers opened in a gesture representing an alphabet of the Binary Sign Language. The system does not require the hand to be perfectly aligned to the camera. The project uses image processing system to identify, especially English alphabetic sign language used by the deaf people to communicate. The basic objective of this project is to develop a computer based intelligent system that will enable dumb people significantly to communicate with all other people using their natural hand gestures. The idea consisted of designing and building up an intelligent system using image processing, machine learning and artificial intelligence concepts to take visual inputs of sign language’s hand gestures and generate easily recognizable form of outputs. Hence the objective of this project is to develop an intelligent system which can act as a translator between the sign language and the spoken language dynamically and can make the communication between people with hearing impairment and normal people both effective and efficient. The system is we are implementing for Binary sign language but it can detect any sign language with prior image processing.

# Proposed Method

# Our proposed system is sign language recognition system using convolution neural networks which recognizes various hand gestures by capturing video and converting it into frames. Then the hand pixels are segmented and the image it obtained and sent for comparison to the trained model. Thus our system is more robust in getting exact text labels of letters.

# 4.1 WORKING ARCHITECTURE

# 

# Methodology

**5.1 Training Module:**

**Supervised machine learning:** It is one of the ways of machine learning where the model is trained by input data and expected output data. Тo create such model, it is necessary to go through the following phases:

1. model construction

2. model training

3. model testing

4. model evaluation

**Model construction:**

It depends on machine learning algorithms. In this projects case, it was neural networks. Such an algorithm looks like:

1. begin with its object: model = Sequential()

2. then consist of layers with their types: model.add(type\_of\_layer())

3. after adding a sufficient number of layers the model is compiled. At this moment Keras communicates with TensorFlow for construction of the model. During model compilation it is important to write a loss function and an optimizer algorithm. It looks like: model.comile(loss= ‘name\_of\_loss\_function’, optimizer= ‘name\_of\_opimazer\_alg’ )

The loss function shows the accuracy of each prediction made by the model.

**Model training:**

After model construction it is time for model training. In this phase, the model is trained using training data and expected output for this data. It’s look this way: model.fit(training\_data, expected\_output). Progress is visible on the console when the script runs. At the end it will report the final accuracy of the model.

**Model Testing:**

During this phase a second set of data is loaded. This data set has never been seen by the model and therefore it’s true accuracy will be verified. After the model training is complete, and it is understood that the model shows the right result, it can be saved by: model.save(“name\_of\_file.h5”). Finally, the saved model can be used in the real world. The name of this phase is model evaluation. This means that the model can be used to evaluate new data.

**5.2 Preprocessing:**

**Understanding aspect ratios:** An aspect ratio is a proportional relationship between an image's width and height. Essentially, it describes an image's shape. Aspect ratios are written as a formula of width to height, like this: For example, a square image has an aspect ratio of 1:1, since the height and width are the same. The image could be 500px × 500px, or 1500px × 1500px, and the aspect ratio would still be 1:1. As another example, a portrait-style image might have a ratio of 2:3. With this aspect ratio, the height is 1.5 times longer than the width. So the image could be 500px × 750px, 1500px × 2250px, etc.

**Cropping to an aspect ratio:** Aside from using built in site style options , you may want to manually crop an image to a certain aspect ratio. For example, if you use product images that have same aspect ratio, they'll all crop the same way on your site.

**Option 1** - Crop to a pre-set shape Use the built-in Image Editor to crop images to a specific shape. After opening the editor, use the crop tool to choose from preset aspect ratios.

**Option 2** - Custom dimensions To crop images to a custom aspect ratio not offered by our built-in Image Editor, use a third-party editor. Since images don’t need to have the same dimensions to have the same aspect ratio, it’s better to crop them to a specific ratio than to try to match their exact dimensions. For best results, crop the shorter side based on the longer side.

• For instance, if your image is 1500px × 1200px, and you want an aspect ratio of 3:1, crop the shorter side to make the image 1500px × 500px.

• Don't scale up the longer side; this can make your image blurry.

**Image scaling:**

• In computer graphics and digital imaging , image scaling refers to the resizing of a digital image. In video technology, the magnification of digital material is known as upscaling or resolution enhancement .

• When scaling a vector graphic image, the graphic primitives that make up the image can be scaled using geometric transformations, with no loss of image quality. When scaling a raster graphics image, a new image with a higher or lower number of pixels must be generated. In the case of decreasing the pixel number (scaling down) this usually results in a visible quality loss. From the standpoint of digital signal processing, the scaling of raster graphics is a two dimensional example of sample-rate conversion, the conversion of a discrete signal from a sampling rate (in this case the local sampling rate) to another.

**5.3 Algorithm**

**Back Propagation:** Back-propagation is the essence of neural net training. It is the method of fine-tuning the weights of a neural net based on the error rate obtained in the previous epoch (i.e., iteration). Proper tuning of the weights allows you to reduce error rates and to make the model reliable by increasing its generalization. Backpropagation is a short form for "backward propagation of errors." It is a standard method of training artificial neural networks. This method helps to calculate the gradient of a loss function with respects to all the weights in the network.

**Optimizer(Adam):** Adam can be looked at as a combination of RMSprop and Stochastic Gradient Descent with momentum. It uses the squared gradients to scale the learning rate like RMSprop and it takes advantage of momentum by using moving average of the gradient instead of gradient itself like SGD with momentum. Adam is an adaptive learning rate method, which means, it computes individual learning rates for different parameters. Its name is derived from adaptive moment estimation, and the reason it’s called that is because Adam uses estimations of first and second moments of gradient to adapt the learning rate for each weight of the neural network. Now, what is moment ? N-th moment of a random variable is defined as the expected value of that variable to the power of n.

**Loss Function(categorical cross entropy):** Categorical cross entropy is a loss function that is used for single label categorization. This is when only one category is applicable for each data point. In other words, an example can belong to one class only. Note. The block before the Target block must use the activation function Softmax.

**Convolutional Neural Network:** Image classification is the process of taking an input(like a picture) and outputting its class or probability that the input is a particular class. Neural networks are applied in the following steps:

1)One hot encode the data: A one-hot encoding can be applied to the integer representation. This is where the integer encoded variable is removed and a new binary variable is added for each unique integer value.

2)Define the model: A model said in a very simplified form is nothing but a function that is used to take in certain input, perform certain operation to its best on the given input (learning and then predicting/classifying) and produce the suitable output.

3)Compile the model: The optimizer controls the learning rate. We will be using ‘adam’ as our optimizer. Adam is generally a good optimizer to use for many cases. The adam optimizer adjusts the learning rate throughout training. The learning rate determines how fast the optimal weights for the model are calculated. A smaller learning rate may lead to more accurate weights (up to a certain point), but the time it takes to compute the weights will be longer.

4)Train the model: Training a model simply means learning (determining) good values for all the weights and the bias from labeled examples. In supervised learning, a machine learning algorithm builds a model by examining many examples and attempting to find a model that minimizes loss; this process is called empirical risk minimization.

5)Test the model A convolutional neural network convolves learned featured with input data and uses 2D convolution layers

**Operations of convolution neural network are as follows:**

1.Convolution Operation

2.ReLu Layer

3.Pooling Layer

4.Fully Connected Layer

# Plan of work

Our model will be developed in different phases.

**First-Phase**: In this phase we will develop the base model for sign language detection. It will be in development phase from 16 April 2022 to 10 May 2022 (approximately).

**Second-Phase**: In this phase we will improve our base model by advancing or configuring it in such a way that it can take the input in dynamic fashion. It will be in development phase from 11 May 2022 to 12 June 2022.

**Third-Phase**: In this phase we will transform or model in such a way that it can dynamically add new gestures to the existing dataset which will be assigned as the user desires and also able to recognize them in real time, it will need at least 5-8 input images of new gestures for its training part. It will be in development phase from 13 June 2022 to 14 July 2022

**Final Phase**: In this our developed model will be deployed as android application using Flutter which is a Google developed application which uses Dart language to develop android apps. It will be in development phase from 15 July 2022 to 10 August 2022.

# References

1. Aurélien-Géron-Hands-On-Machine-Learning-with-Scikit-Learn-Keras-and-Tensorflow\_-Concepts-Tools-and-Techniques-to-Build-Intelligent-Systems-O’Reilly-Media
2. Research paper on convolutional neural network by S Albawi.
3. Udemy Machine Learning course by kiril.
4. Full Flutter course by FreeCodeCamp on YouTube.