SIGN LANGUAGE AND HAND GESTURE RECOGNITION SYSTEM USING CONVOLUTIONAL NEURAL NETWORK

A project report for the disciplinary project

PDPM
IIITDM
Jabalpur

Pandit Dwarka Prasad Mishra, Indian Institute of Information Technology, Design and Manufacturing, Jabalpur

AN INSTITUTE OF NATIONAL IMPORTANCE ESTABLISHED BY THE MINISTRY OF HOME AND RESOURCE MANAGEMENT

Submitted by:

1) Rajat Jain	21BEC084
2) Krish	21BEC062
3) Arnav Puri	21BEC023

UNDER THE GUIDANCE OF

DR. IRSHAD AHMAD ANSARI and **DR.VARUN BAJAJ**

DEPARTMENT OF ELECTRONICS AND COMMUNICATION

Acknowledgment

We would like to express our deep gratitude towards **Dr. Irshad Ahmad Ansari and Dr. Varun Bajaj**, Department of Electronics and Communication, Pandit Dwarka Prasad Mishra Indian Institute of Information Technology, Design and Manufacturing, Jabalpur, for their guidance with unsurpassed knowledge and immense encouragement.

We would like to thank **Dr. Matadeen Bansal, Head of Department, Department of Electronics and Communication, PDPM IIITDMJ**, for giving us this opportunity to work on this project which helped us develop our skills in the field of Artificial Intelligence and Machine Learning and contribute towards society as well.

We would like to thank all the teaching and non-teaching staff of the Department of Electronics and Communication, PDPM IIITDMJ, for their immense support and guidance through reviews of the system.

Lastly, we would like to thank our parents, friends, and classmates for their immense support and encouragement during our project. We would like to thank everyone who supported us directly or indirectly during this time.

TABLE OF CONTENTS

ABSTRACT

Sign language is the only tool of communication for a person who is not able to speak and hear anything. Sign language is a boon for physically challenged people to express their thoughts and emotion. This work proposes a novel scheme of sign language recognition 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.

Introduction

The goal of this project was to build a neural network able to classify which letter of the Indian Sign Language alphabet is being signed, given an image of a signing hand. This project is a first step towards building a possible sign language translator, which can take communications in sign language and translate them into written and oral language. Such a translator would greatly lower the barrier for many deaf and mute individuals to be able to better communicate with others in day-to-day interactions.

This goal is further motivated by the isolation that is felt within the deaf community. Loneliness and depression exist at higher rates among the deaf population, especially when they are immersed in a hearing world. Large barriers that profoundly affect life quality stem from the communication disconnect between the deaf and the hearing. Some examples are information deprivation, limitation of social connections, and difficulty integrating into society.

Most research implementations for this task have used depth maps generated by depth cameras and high-resolution images. The objective of this project was to see if neural networks are able to classify signed ISL letters using simple images of hands taken with a personal device such as a laptop webcam. This is in alignment with the motivation as this would make a future implementation of a real-time ISL-to-oral/written language translator practical in an everyday situation.

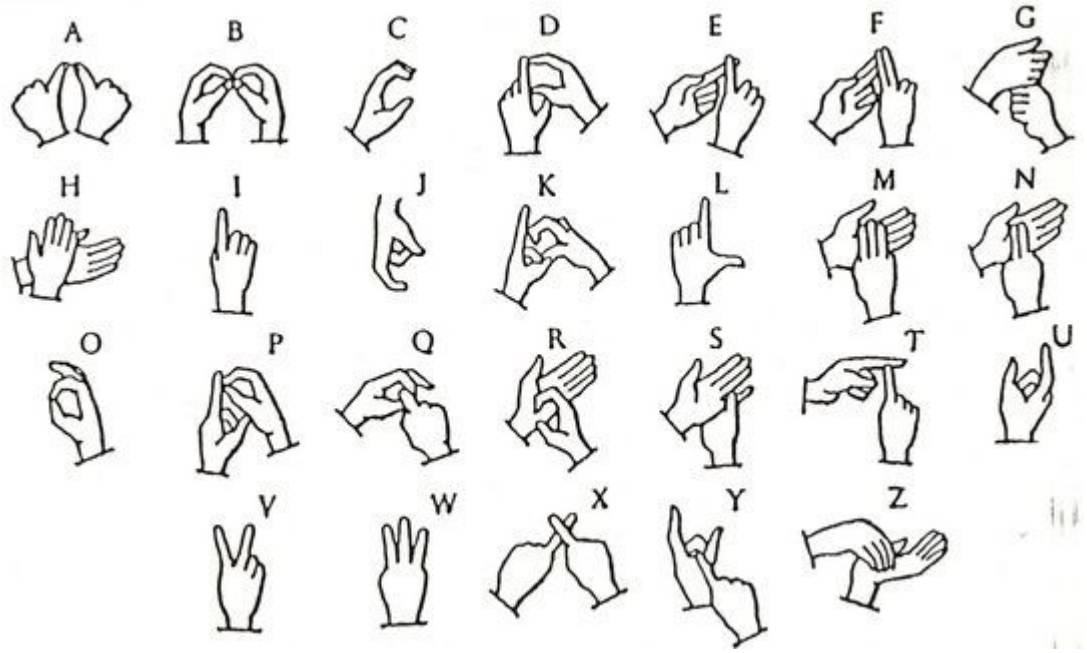
2.1 IMAGE PROCESSING

Image processing is a method to perform some operations on an image, in order to get an enhanced image or to extract some useful information from it. It is a type of signal processing in which the input is an image and the output may be an image or characteristics/features associated with that image.

Nowadays, image processing is among the rapidly growing technologies. It forms a core research area within engineering and computer science disciplines too. Image processing basically includes the following three steps:

• Importing the image via image acquisition tools.
• Analysing and manipulating the image.
• Output in which the result can be altered image or report that is based on image analysis.

2.2 SIGN LANGUAGE



It is a language that includes gestures made with the hands and other body parts, including facial expressions and postures of the body. It is used primarily by people who are deaf and dumb. There are many different sign languages as, British, Indian, and American

sign languages. British sign language (BSL) is not easily intelligible to users of American Sign Language (ASL) and vice versa.

2.3 SIGN LANGUAGE AND HAND GESTURE RECOGNITION

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 are 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.

2.4 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 percent of the 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 the sign language of impaired people. Our project hence is aimed at converting sign language gestures into text that is readable for normal people.

2.5 PROBLEM STATEMENT

Speech-impaired people use hand signs and gestures to communicate. Normal people face difficulty in understanding their language. Hence there is a need for a system that recognizes the different signs, and gestures and conveys the information to normal

people. It bridges the gap between physically challenged people and normal people.

Keywords and Definitions

3.1 INTRODUCTION

The domain analysis that we have done for the project mainly involved understanding the neural networks

Feature Extraction and Representation:

The representation of an image as a 3D matrix having dimensions as of height and width of the image and the value of each pixel as depth (1 in case of Grayscale and 3 in case of RGB). Further, these pixel values are used for extracting useful features using CNN.

Artificial Neural Networks:

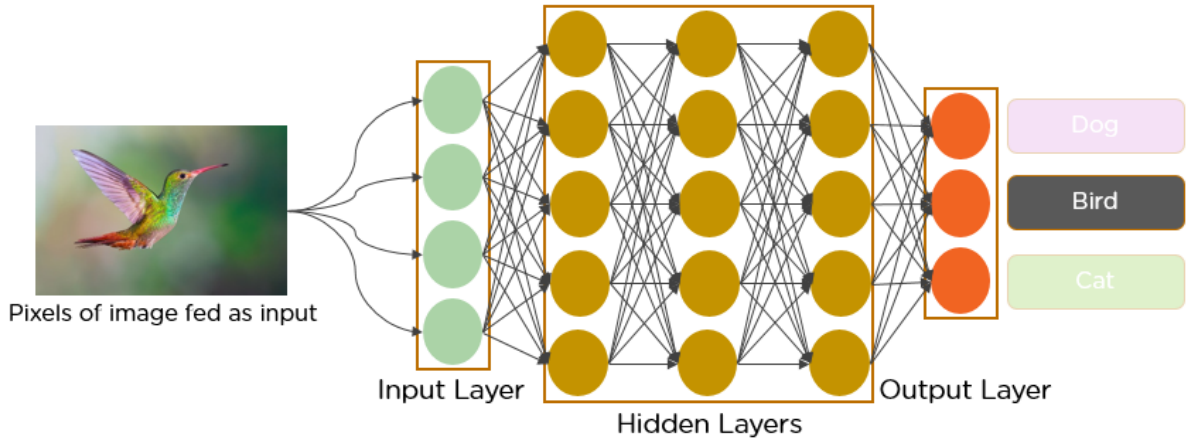
Artificial Neural Network is a connection of neurons, that replicates the structure of the human brain. Each connection of a neuron transfers information to another neuron. Inputs are fed into the first layer of neurons which processes it and transfers it to another layer of neurons called hidden layers. After processing information through multiple layers of hidden layers, information is passed to the final output layer(36). They are capable of learning and they have to be trained. There are different learning strategies:

1. Unsupervised Learning
2. Supervised Learning
3. Reinforcement Learning

Convolution Neural Network:

Unlike regular Neural Networks, in the layers of CNN, the neurons are arranged in 3 dimensions: width, height, and depth. The neurons in a layer will only be connected to a small region of the layer (window size) before it, instead of all of the neurons in a fully-connected manner. Moreover, the final output layer would have dimensions (number of classes), because by the end of the CNN

architecture, we will reduce the full image into a single vector of class scores.



1. **Convolution Layer**: In the convolution layer we take a small window size [typically of length 5*5] that extends to the depth of the input matrix. The layer consists of learnable filters of window size. During every iteration, we slide the window by stride size [typically 1], and compute the dot product of filter entries and input values at a given position. As we continue this process will create a 2-Dimensional activation matrix that gives the response of that matrix at every spatial position. That is, the network will learn filters that activate when they see some type of visual feature such as an edge of some orientation or a blotch of some colour.

2. Pooling Layer: We use the pooling layer to decrease the size of the activation matrix and ultimately reduce the learnable parameters. There are two types of pooling:

a) Max Pooling: In max pooling, we take a window size [for example window of size 2*2], and only take the maximum of 4 values. Well, lid this window and continue this process, so we'll finally get an activation matrix half of its original Size.

b) Average Pooling: In average pooling, we take an average of all values in a window.

3. Fully Connected Layer: In the convolution layer neurons are connected only to a local region, while in a fully connected region, well connect all the inputs to neurons.

4. Final Output Layer: After getting values from a fully connected layer, we will connect them to the final layer of neurons[having a count equal to a total number of classes], which will predict the probability of each image being in different classes.

3.2 KERAS API

These are the building blocks of the neural network in Keras. Consists of a tensor-in-tensor out computational function (layer's call method) and some state held in the Tensor Flow variable(the layer's weights)

We have used Convolution2D, MaxPooling2D, Flatten, Dense, and Dropout layers of Keras API.

3.3 NUMPY

Numpy is a library in Python.

3.4 OPENCV:

OpenCV (Open Source Computer Vision Library) is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage, then Itseez (which was later acquired by Intel).

The library is cross-platform and free for use under the open-source BSD licence.

3.5: Grad-CAM Visualization:

Grad-CAM visualisation is a technique used to explain the predictions of a convolutional neural network (CNN). It works by creating a heatmap that highlights the regions of an image that are most important for CNN's prediction.

The heatmap is created by backpropagating the gradients of the CNN's output with respect to the input image. The gradients are then averaged over the channels of the final convolutional layer. The resulting heatmap shows the relative importance of each pixel in the input image for CNN's prediction.

Methodology

We have used Convolutional Neural Networks and Artificial Neural Networks for this project

4.1 DATA COLLECTION

We have used a dataset available online on Kaggle.com.
Since gathering and creating a dataset is an expensive as well as a time-consuming process, we trained our model with the limited resources we had.

All the images in our data set are in binary scale. Since we have tested our model using binary scale only, our model also takes input images in RGB and converts it into BINARY SCALE for Prediction.

We used two convolutional layers and the dropout function to reduce overfitting so that we could reduce the competition by using kernels.

4.2 DATA PREPROCESSING

For preprocessing of data, we used data augmentation. Data augmentation is a technique used to artificially increase the size of a dataset by creating new data points from existing data. This is done by applying a set of transformations to the existing data, such as cropping, flipping, rotating, and adding noise. The goal of data augmentation is to make the training dataset more diverse and representative of the real world, which can help to improve the performance of machine learning models.

4.3 THE ALGORITHM

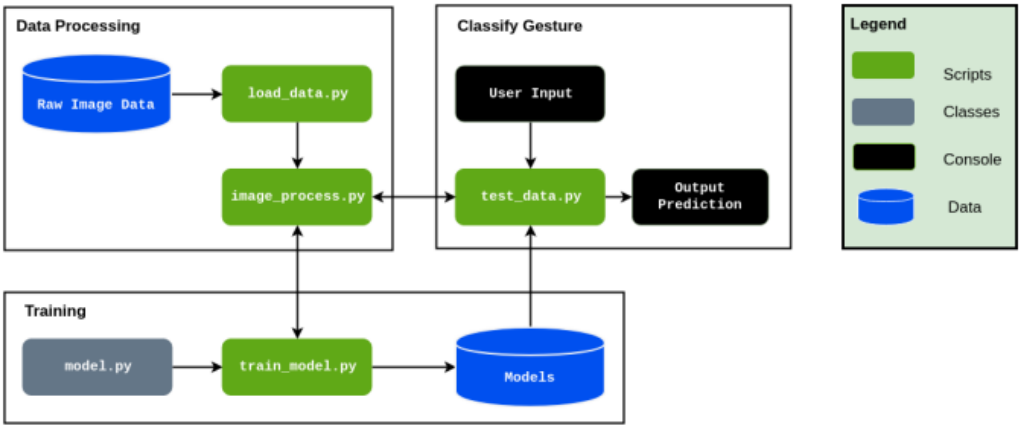


Fig 4.1 Diagrammatic representation of the algorithm

Our algorithm uses two convolutional layers. With the use of the Keras layers, we performed the Max Pooling operation.
We also used the Dropout operation so that we could tackle the problem of overfitting in our data
We used data augmentation so that our model even takes images that have slight noise in them or are shown from a different angle.
We used 80% of our data set for training our model and the rest for validation purposes which is the best-fit ratio **(80:20)**.
Lastly, we put the results of the convolutional layer over the ANN layer to train our model.

Training and Testing

5.1 TRAINING

We gathered our data set from Kaggle.com.
We used 80% of the data set for training and the rest for validation purposes through various tests and trials which, as per the research done in the field, is the best-fit ratio for a CNN model.

We had 36 classes in total, 26 for the alphabet from A to Z and 10 for the numbers from 0 to 9.

Optimizer: The optimizer is to be used during training. It is responsible for updating the weights of the model during training.
Our model is trained using Adam optimizer. Adam Optimizer is known for its performance and quick convergence.

Loss function: The loss function is used to measure the error between the model's predictions and the actual ground label. We have used the Cross Entropy function which is popularly used for classification problems(more than two classes).

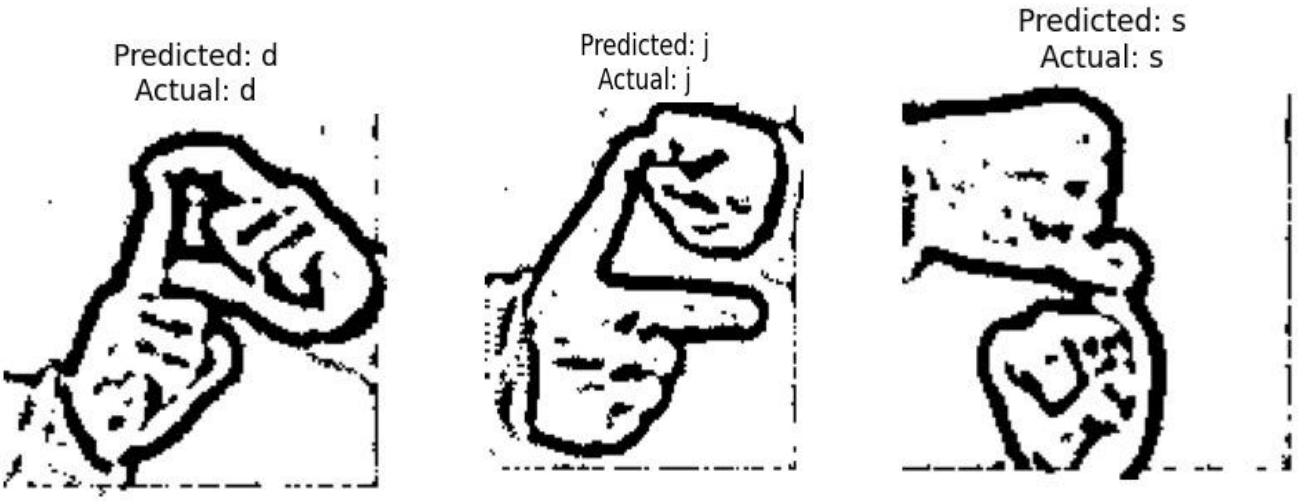
As we have mentioned earlier, we used data augmentation so that our model is able to see different angles of our training images. So that if there is any noise in our image it is able to predict what the sign in the image is showing.

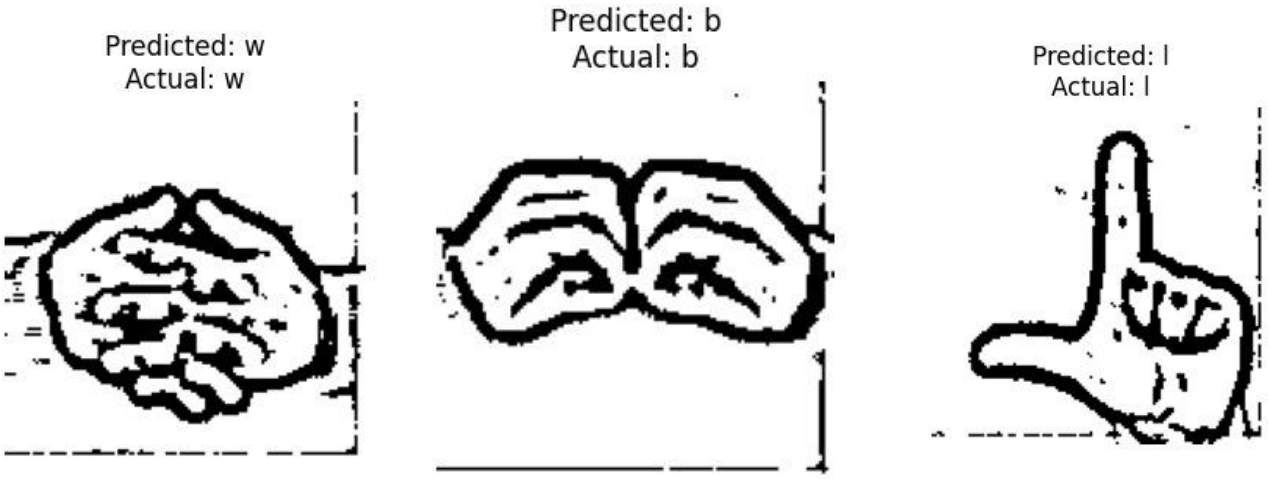
The batch size for our training data set is 64 images per batch and the number of epochs done is 40.
Since all the images were of different sizes, we used a standard size of 128 pixels by 128 pixels and accordingly we resized them.

5.2 TESTING

To determine whether our preprocessing of images actually results in a more robust model, we verified on a test set composed of images from the test dataset.

The screenshots from our testing are hereby attached below.**(TERMINAL PICS)**





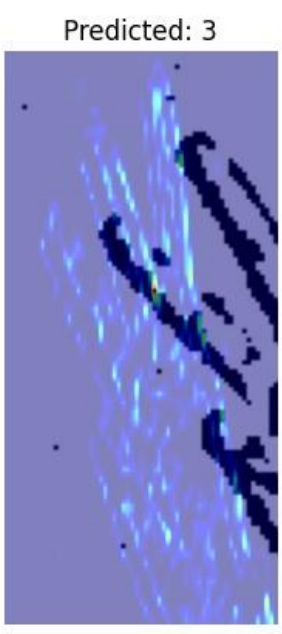
For better testing of our model, we also used the Grad -CAM VISUALISATION technique.

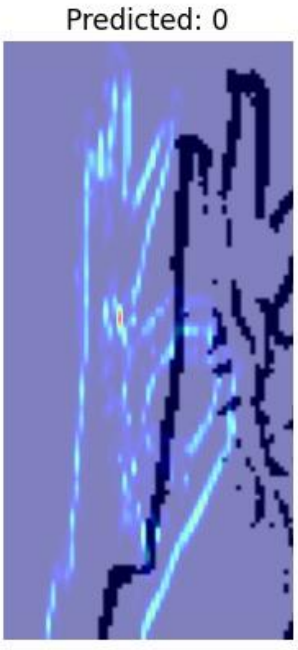
We used the Grad-CAM technique to generate a heatmap that highlights the most relevant regions of the input image for the model's prediction. The heatmap indicates which parts of the image contributed the most to the model's decision. Brighter areas on the heatmap correspond to regions that the model focused on.

Overlaying Heatmap: We overlaid the heatmap on top of the original grayscale image using the matplotlib library. This overlay visually demonstrates where the model was looking to make its prediction. The heatmap was colour-coded using the "jet" colormap to indicate the intensity of attention.

Interpretation: By examining the heatmap overlay, we can interpret which parts of the hand gesture are most significant for the model's classification decision. For instance, if the heatmap focuses on the fingers or specific hand movements, it suggests that those features strongly contribute to the model's understanding of the gesture.

The results of testing through this technique are shown below.





Challenges

During the journey of creating this model we faced a bunch of challenges.

1) Since this was a totally new concept for us, we had to start everything from scratch.
2) During the early stage of creating our model, we did not use the dropout function which resulted in over-fitting of the data. As soon as we learned of our mistake, we corrected it.
3) First we used all the training data for validation purposes, but then we used 80% of our data set for training our model and 20% for testing from the given training data which is the best-fit ratio.

Key Learnings

We did face a lot of challenges, but with the guidance and knowledge of our mentors, we overcome those challenges with ease. There were a lot of learnings during this whole journey and we would like to document a few.

1) The concept of artificial intelligence was a totally new concept for us. This project helped us not only learn about this wonderful thing but also to hone our skills in the field.
2) During this project we learned about various functions in the domains and their implementations.
3) While working on this project we also learned a lot about the hardships which the Deaf and Dumb people face. This made us realise the value of what we have been given as a person and be grateful for it.

Results

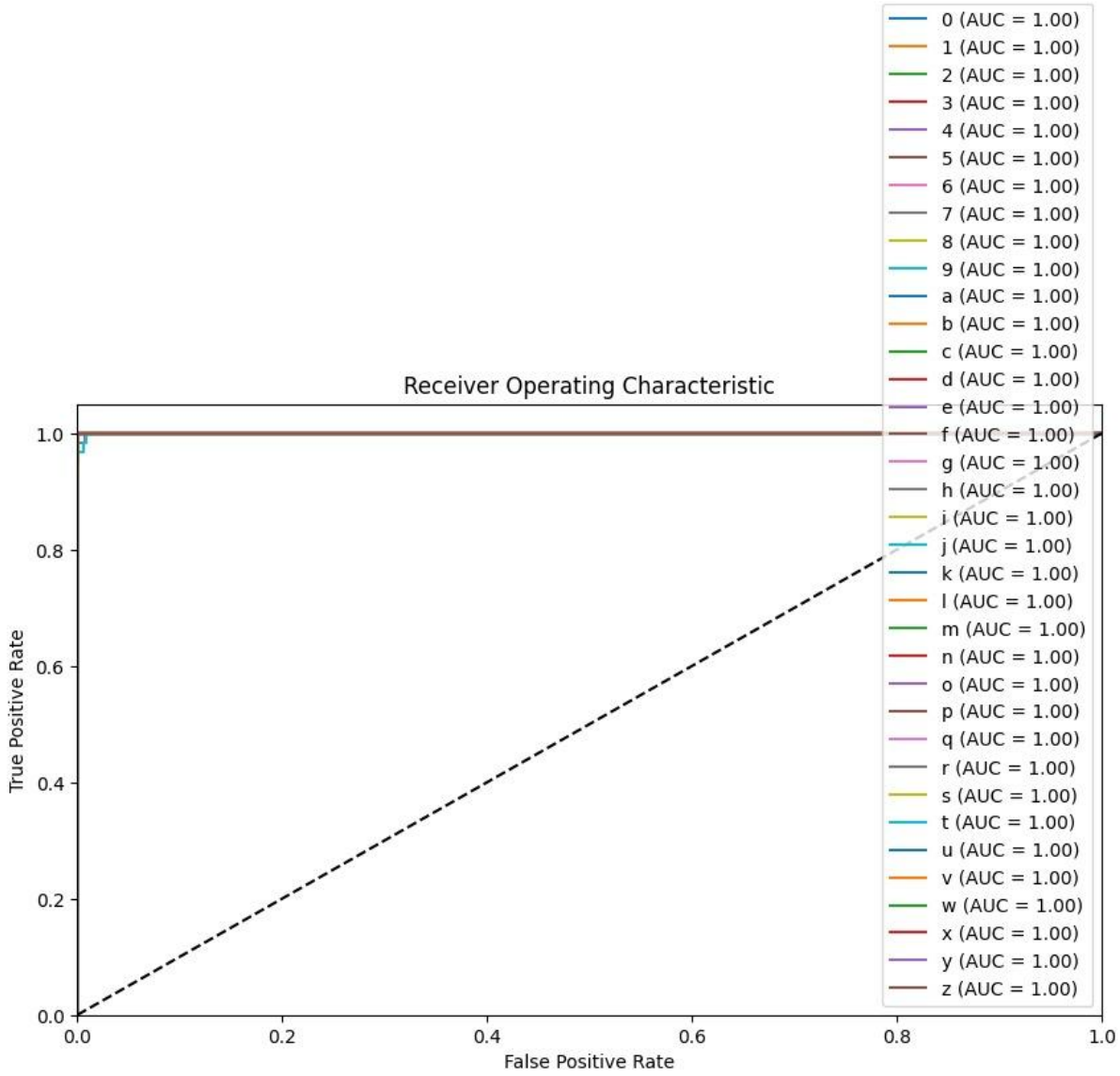
The results after all our training and testing are shown below.

```
244/244 [==============================] - 176s 719ms/step - loss: 0.0661 - accuracy: 0.9793 - val_loss: 0.0184 - val_accuracy:
0.9946
Epoch 36/40
244/244 [==============================] - 175s 718ms/step - loss: 0.0669 - accuracy: 0.9795 - val_loss: 0.0244 - val_accuracy:
0.9908
Epoch 37/40
244/244 [==============================] - 174s 713ms/step - loss: 0.0704 - accuracy: 0.9783 - val_loss: 0.0263 - val_accuracy:
0.9928
Epoch 38/40
244/244 [==============================] - 174s 712ms/step - loss: 0.0613 - accuracy: 0.9795 - val_loss: 0.0197 - val_accuracy:
0.9926
Epoch 39/40
244/244 [==============================] - 173s 708ms/step - loss: 0.0638 - accuracy: 0.9797 - val_loss: 0.0151 - val_accuracy:
0.9961
Epoch 40/40
244/244 [==============================] - 174s 712ms/step - loss: 0.0625 - accuracy: 0.9797 - val_loss: 0.0273 - val_accuracy:
0.9920
Model Saved
Weights saved
```

Classification Report:

	precision	recall	f1-score	support
0	0.03	0.03	0.03	131
1	0.02	0.02	0.02	60
2	0.02	0.02	0.02	60
3	0.00	0.00	0.00	60
4	0.00	0.00	0.00	60
5	0.00	0.00	0.00	60
6	0.02	0.02	0.02	60
7	0.02	0.02	0.02	60
8	0.00	0.00	0.00	60
9	0.00	0.00	0.00	60
a	0.02	0.02	0.02	60
b	0.04	0.04	0.04	181
c	0.00	0.00	0.00	60
d	0.05	0.05	0.05	240
e	0.03	0.03	0.03	60
f	0.02	0.02	0.02	60
g	0.04	0.04	0.04	240
h	0.03	0.03	0.03	240
i	0.07	0.07	0.07	240
j	0.05	0.05	0.05	240
k	0.03	0.03	0.03	240
l	0.06	0.06	0.06	240
m	0.00	0.00	0.00	60
n	0.02	0.02	0.02	60
o	0.02	0.02	0.02	60
p	0.00	0.00	0.00	60
q	0.00	0.00	0.00	60
r	0.03	0.03	0.03	60
s	0.07	0.07	0.07	240
t	0.06	0.06	0.06	240
u	0.00	0.00	0.00	60
v	0.08	0.08	0.08	240
w	0.03	0.03	0.03	240
x	0.06	0.06	0.06	240
y	0.04	0.04	0.04	240
z	0.06	0.06	0.06	240
accuracy			0.04	4872
macro avg	0.03	0.03	0.03	4872
weighted avg	0.04	0.04	0.04	4872

The receiver operating characteristic curve (ROC curve) which is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied for our model is as follows



Our model showed an average accuracy of 97.9 % with an average value-loss of 0.0273 and an average validation accuracy of 99.20%. The results show that our model is in comparison with the already existing models in this domain and even better than some of them.

Future Scope

We are planning to use this model for real-time video detection of hand gestures which will be a step up towards our goal of supporting the deaf and dumb people and creating a better world for them.

We hope to design an Artificial Intelligence model in the future which will detect hand gestures in real time as well as be able to read videos.

With new innovations taking place in the field of Artificial Intelligence every day, we dream of a world where people with disabilities, people for whom the normal day-to-day tasks are way harder than one can think of and people who face so many hardships on a daily basis can live a peaceful and fulfilling life.

SOURCES USED

For our project, we used the following sources:

1)Kaggle. Com

https://www.kaggle.com
We used Kaggle for the collection of our dataset.

2) Stackoverflow

https://stackoverflow.com
We referred to StackOverflow for solutions to some of the problems that we faced during this project.