SIGN LANGUAGE TRANSLATOR

BY:

ABHISHEK KUMAR (18JE0027)

RAHUL MISHRA (18JE0649)

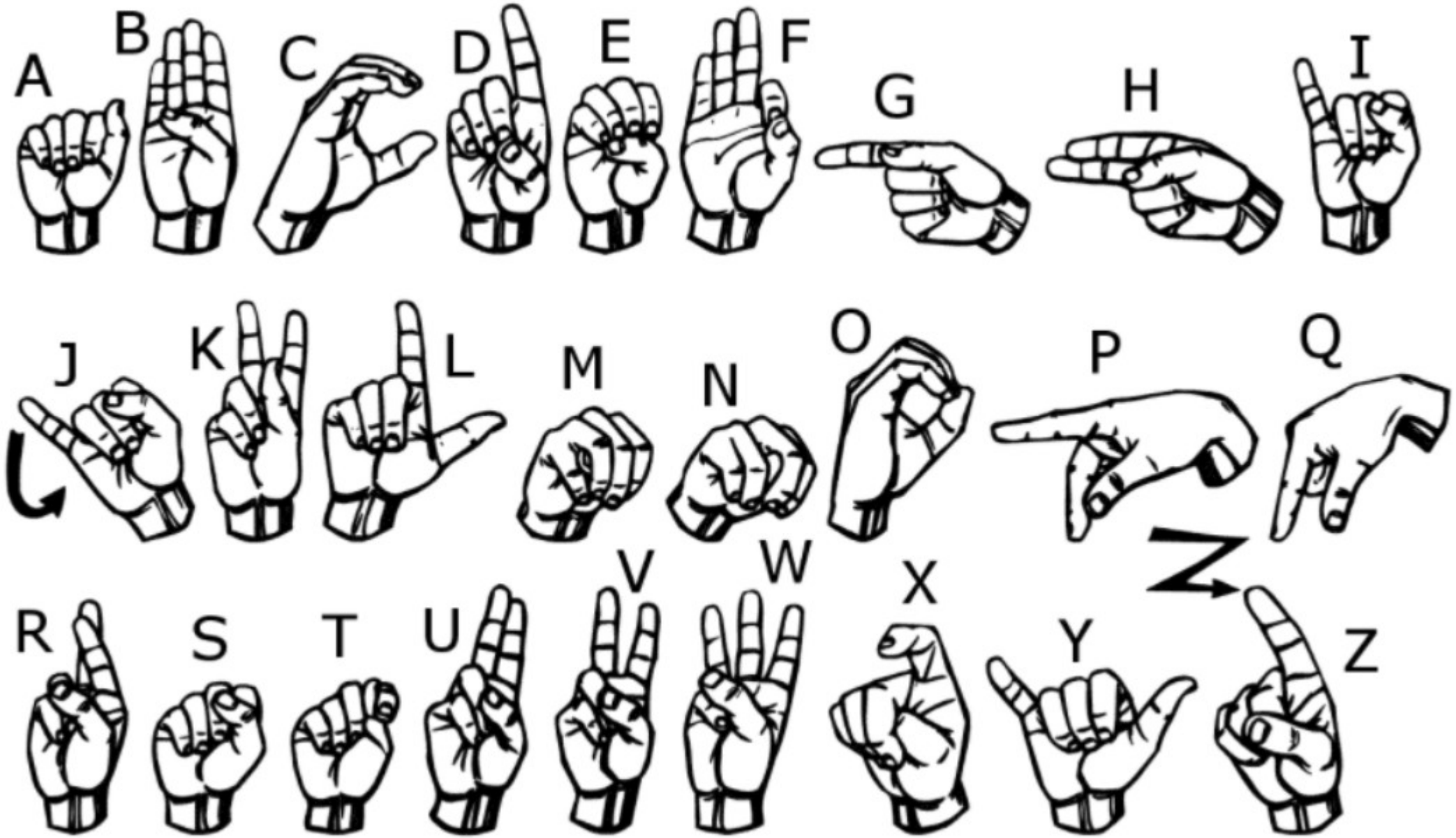
Table Of Contents

AIM

- To make a robust, user friendly interface which can translate sign language to text and further speak the inner voice of the mute people on their behalf.

INTRODUCTION

- Sign languages are visual-based natural languages used by the mute people for their communication.
- Most of the hearing people cannot understand sign language. Thus, sign language translation (SLT) has become an important application to bridge the communication gap between deaf and hearing people.


A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z

OUR SOLUTION

- To build the communication gap and assist mute people, we have come up with the idea of sign language translation.
- Sign Language Translator will take the sign from mute people and then will interpret that sign and provide the output in the form of English alphabetical letter.
- To achieve this we need a mechanism which can interpret the alphabetical and numerical meaning of the sign.
- This mechanism can be achieved by a Deep Learning model trained over sign image dataset.

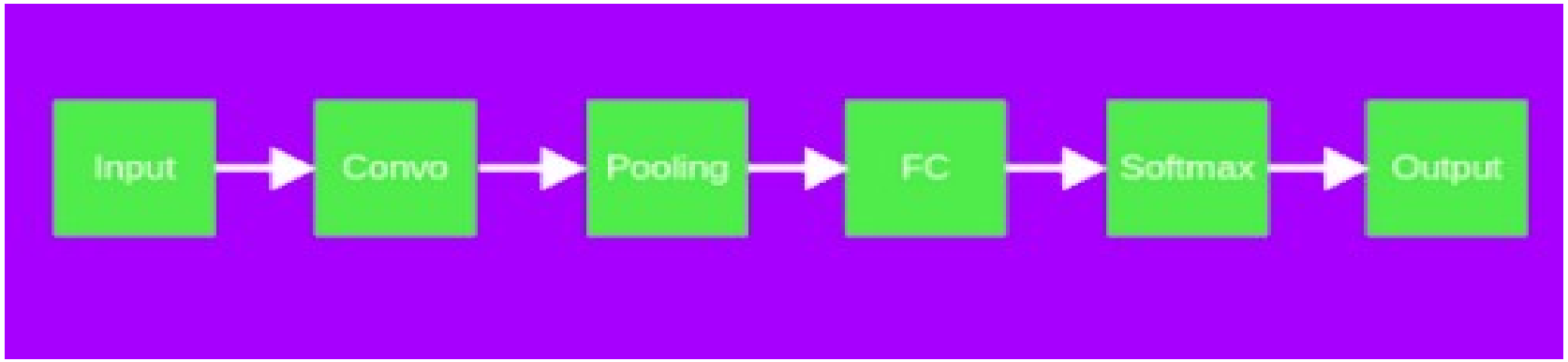
IMPLEMENTATION

- We designed a convolutional neural network which takes sign image as input.
- Our CNN architecture consist of 5 blocks of convolutional layer followed by Max Pooling layer with pool size of 2.
- To avoid overfitting we also have used dropout layers with a dropout ratio of 0.5.
- Softmax Classifier is used at last to predict the probability scores of all 26 English Alphabets.

Layers in CNN

- The first layer of the CNN usually detects basic features such as horizontal, vertical, and diagonal edges.

- The output of the first layer is fed as input of the next layer, which extracts more complex features, such as corners and combinations of edges. As we move deeper into the convolutional neural network, the layers start detecting higher-level features such as objects, faces, and more.



- Technologies and Tools Used:

Python
Convolutional Neural Network
Keras
Android Studio
Flask

Three Level Architecture:

User End : Android App
Backend : Deep Learning Model
Communicator : Flask

```
keras.layers.Conv2D(filters=64, kernel_size=3, input_shape=[28, 28, 1]),
keras.layers.MaxPooling2D(pool_size=2),
keras.layers.Conv2D(filters=128, kernel_size=3, activation='relu', padding='same'),
keras.layers.Conv2D(filters=128, kernel_size=3, activation='relu', padding='same'),
keras.layers.MaxPooling2D(pool_size=2),
keras.layers.Conv2D(filters=128, kernel_size=3, activation='relu', padding='same'),
keras.layers.Conv2D(filters=128, kernel_size=3, activation='relu', padding='same'),
keras.layers.MaxPooling2D(pool_size=2),
```

DEPLOYMENT

- To make the life easier of mute people, there should be an easily accessible platform where sign translation can be done in real time.
- To achieve this we have chosen an android application through which the hand gestures sign can be converted into alphabet with the help of our trained model.
- To communicate our android application with server we have made a server using flask.
- User will be able to take picture of a particular sign. The picture will be sent over the server where image will be fed in our model and the predicted alphabet will be returned again to the android application.

IMPROVEMENTS WE CAN HAVE IN FUTURE

- Increase the vocabulary of our model. Add more and more sign languages to the model.
- Incorporate feedback mechanism to make the model more robust.
- Model accuracy can be further improved by increasing the layers our model.

CONCLUSION

- Sign Language Recognition is a breakthrough for helping deaf-mute people and has been researched for many years.
- This technique removes the barrier of the world by which mute people cannot directly interact with people around them and helps them to overcome their disability.