UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

Automated Sign Language Recognition based on Deep Learning

Student: **Denisa-Mariana FILIP**

Supervisor: **Prof. Dr. Eng. Florin ONIGA**

INTRODUCTION

1.5 billion

people are affected by hearing loss worldwide

2.5 billion

people are estimated to suffer from some degree of hearing loss by 2050 [1]

Hearing-impaired individuals can feel isolated from society and are more likely to experience depression and other negative health outcomes [2]

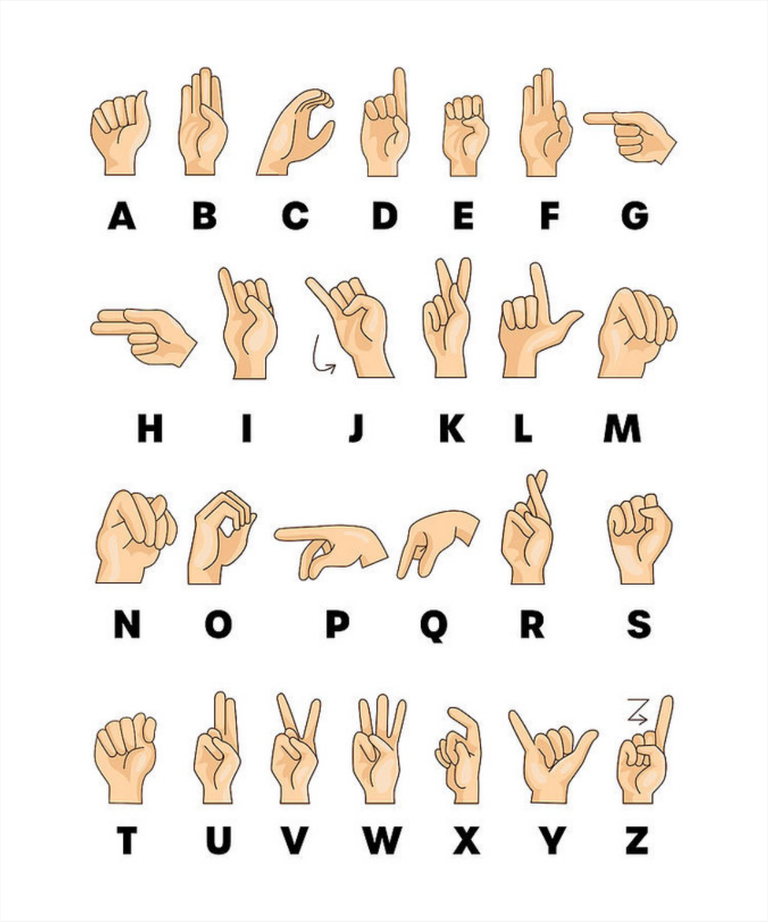
[1] Jennifer Wirth, “Deafness And Hearing Loss Statistics”
[2] Peter R. Dixon and David Feeny and George Tomlinson, “Health-Related Quality of Life Changes Associated With Hearing Loss”

AMERICAN SIGN LANGUAGE

American Sign Language is the most widely spread sign language across the globe and it has the most native signers.

It is a visual language that uses hand gestures, facial expressions and body language.

With its 26 letters, all words can be formed.

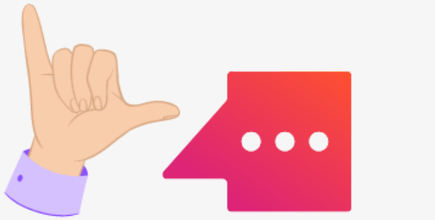


PROJECT PROPOSAL

A communication bridge between the hearing-impaired community and those unfamiliar with sign language.

Objectives:

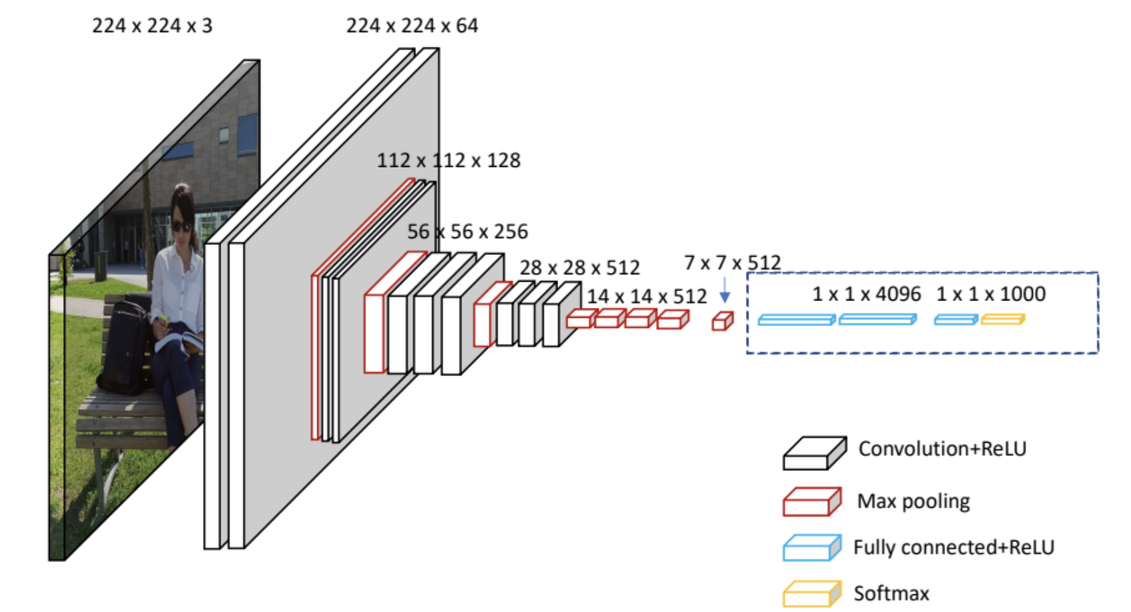
- **Sign Language Recognition (SLR)** deep learning model that recognizes and classifies sign language gestures from the American Sign Language;
- its integration into the Angular frontend of the proposed web application.



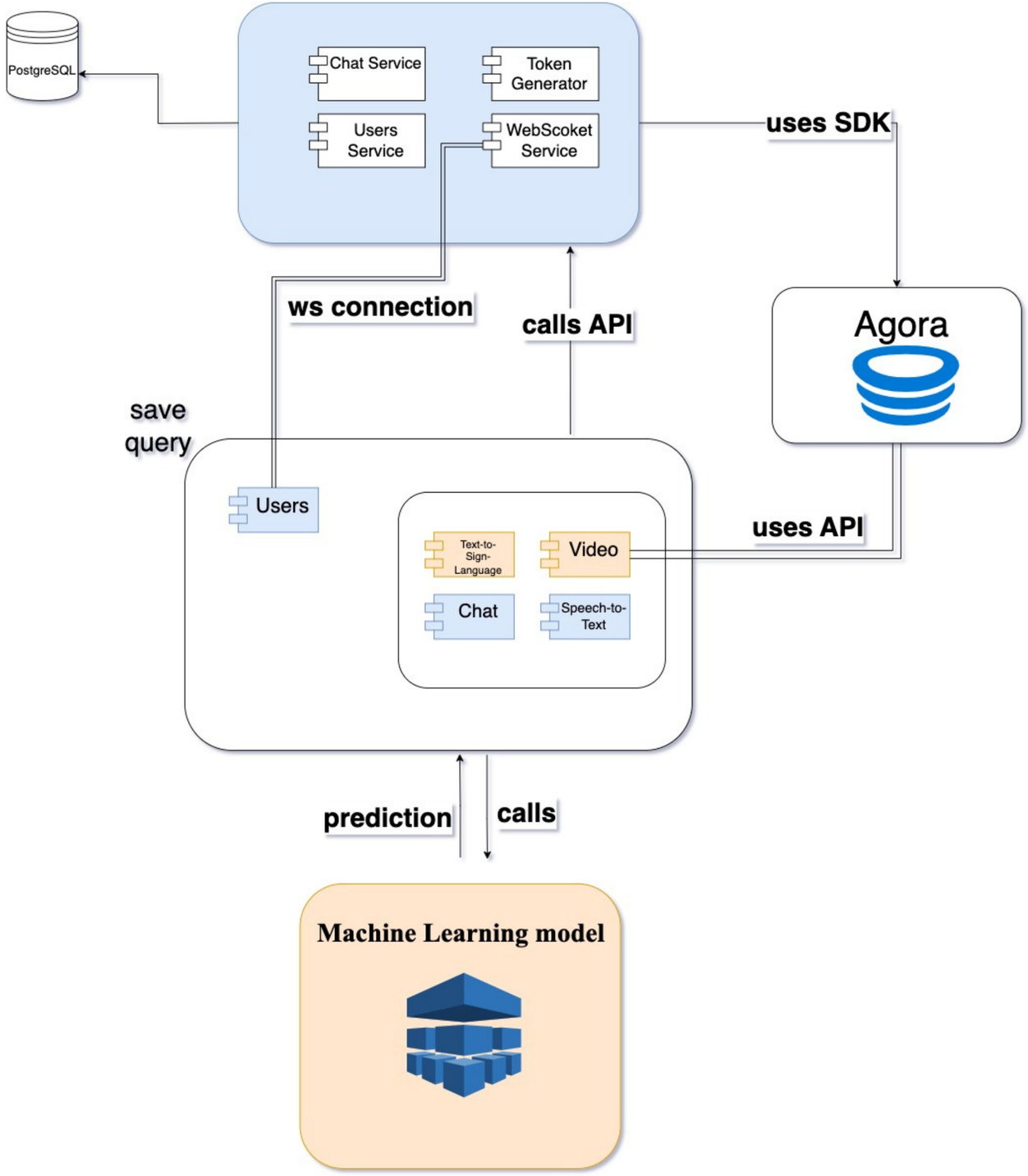
RELATED WORK

BrightSign Glove

CNN Architectures



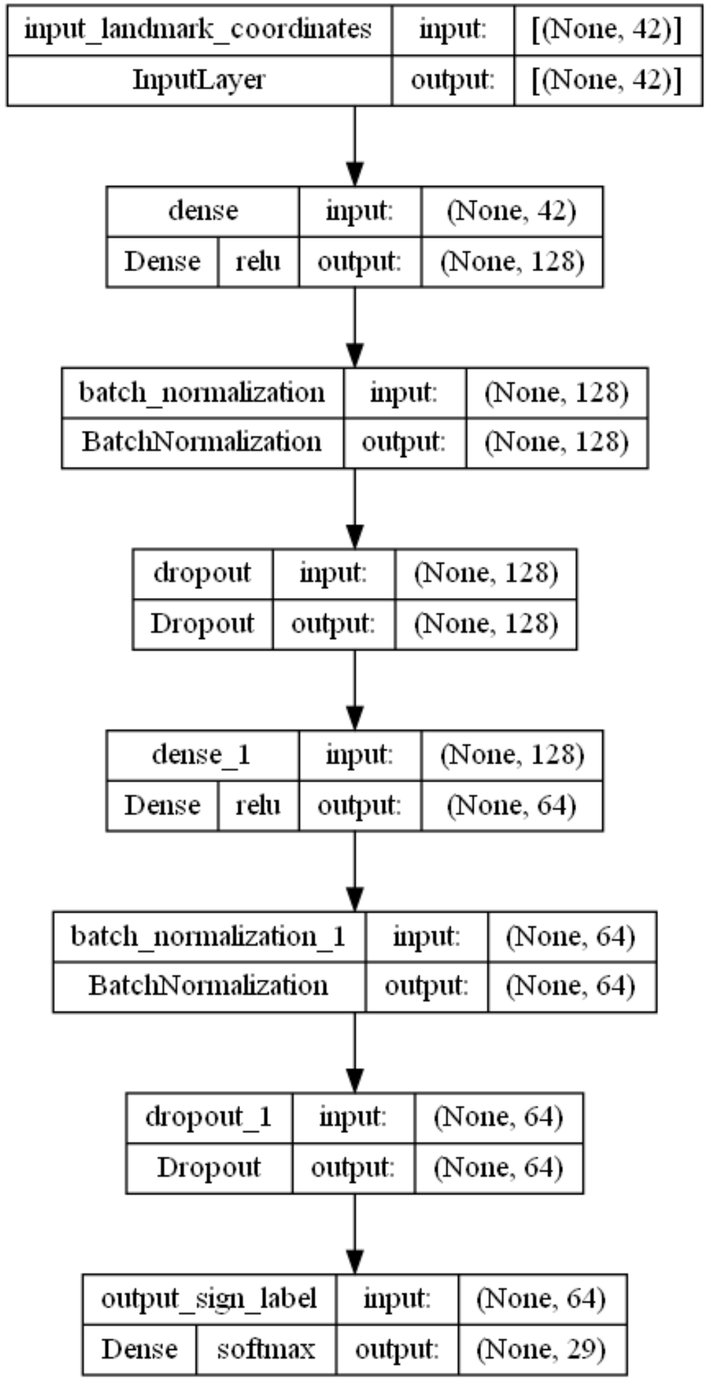
SOLUTION


PostgreSQL
Chat Service
Token Generator
Users Service
WebScoket Service
uses SDK
ws connection
calls API
Agora
save query
Users
Text-to-Sign-Language
Video
Chat
Speech-to-Text
uses API
prediction
calls
Machine Learning model

SLR MODEL

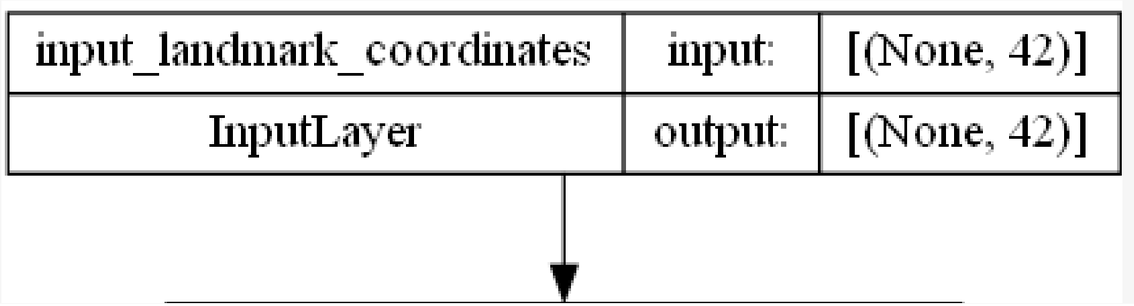
A deep learning model for the Sign Language Recognition (SLR) task, that is a fully connected multi-layer neural network, having:

- 1 input layer;
- 6 hidden layers (2 groups of Dense, BatchNormalization and Dropout layers);
- 1 output layer.



INPUT LAYER

- first layer of the model
- receives 42 x and y coordinates, corresponding to the landmarks that describe the hand position
- does not process the input data any further, sending it to the subsequent layer without applying any other transformations



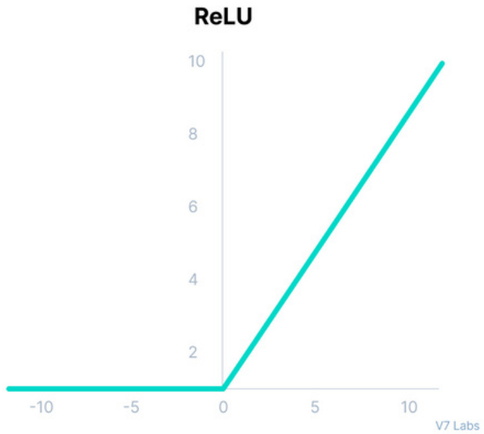
HIDDEN LAYERS

Dense Layer (128 units, ReLU activation)

A fully connected layer with 128 neurons, where each node receives input from all the nodes of the previous layer.

It applies the **ReLU activation function** to its outputs, which transforms any negative value to zero, helping introduce non-linearity in the model.

dense		input:	(None, 42)
Dense	relu	output:	(None, 128)

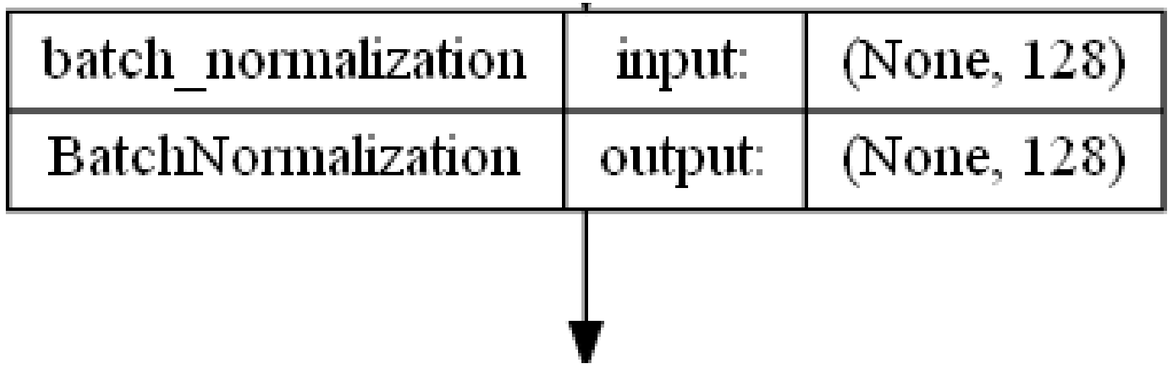


HIDDEN LAYERS

Batch Normalization Layer

Normalizes the inputs of the batch, so they have a mean of 0 and a standard deviation of 1.

This layer was added to **increase** the **training speed** of the model and to make it **less sensitive** to the initial weights of the architecture.

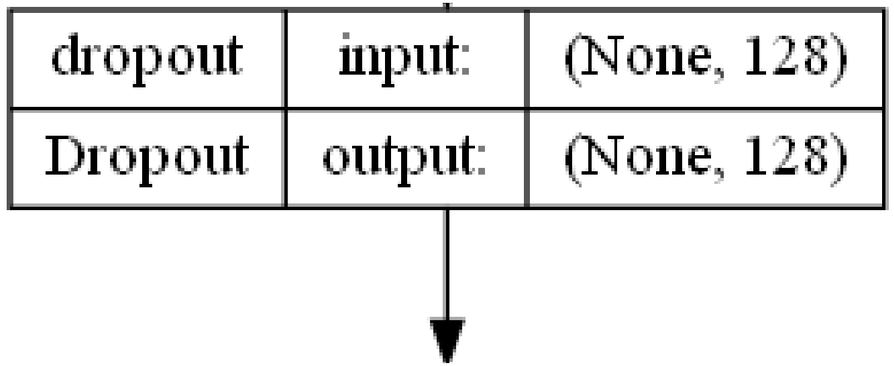


HIDDEN LAYERS

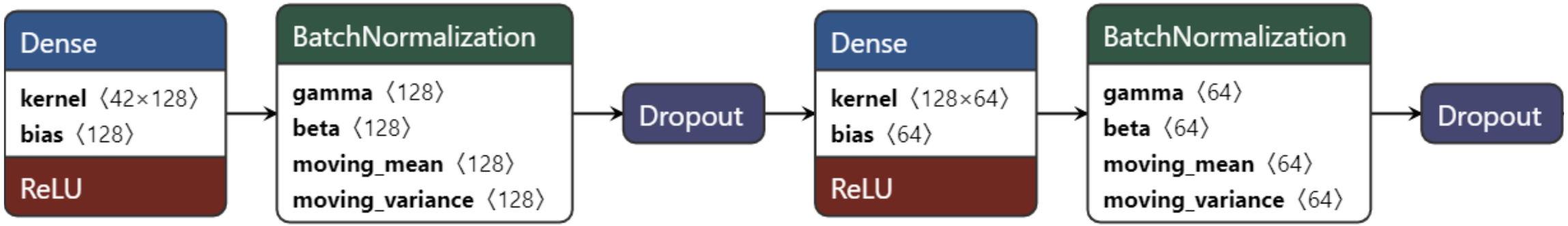
Dropout Layer

Randomly sets a part of the input units to 0 during the training time, which aids to prevent **overfitting**.

In this case, **20%** of the inputs are ignored.



HIDDEN LAYERS

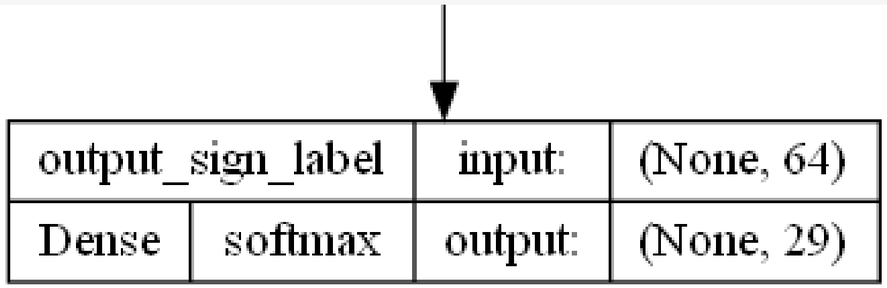


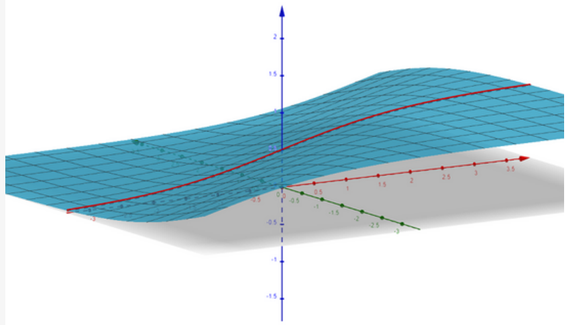
The succession of **Dense**, **Batch Normalization** and **Dropout** layers is repeated, but with **64** units.

This narrows down the number of neurons used, limiting the model's capacity and helping further prevent **overfitting**.

OUTPUT LAYER

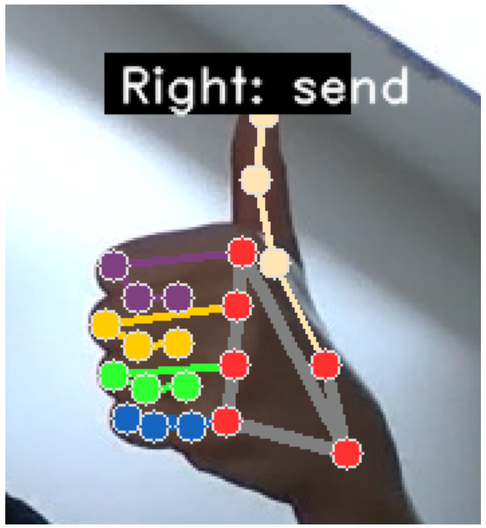
- the final layer of the neural network, having a number of neurons that is equal to the number of classes (i.e. the number of possible outcomes in which the model can classify input data)
- applies the **Softmax** activation function, which is a suitable choice for multi-class classification problems



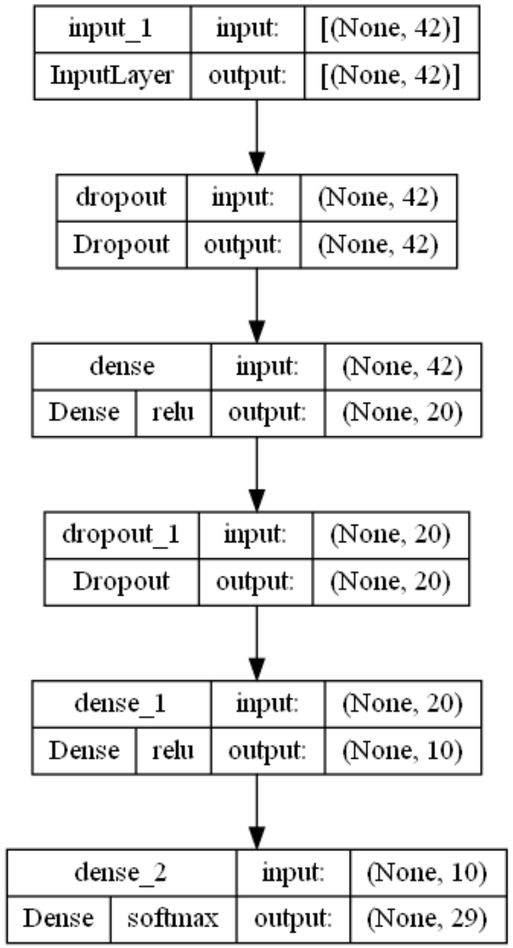


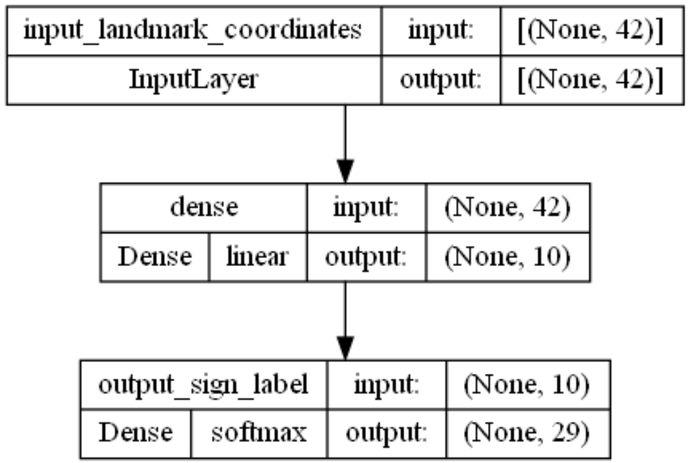
POSSIBLE OUTCOMES

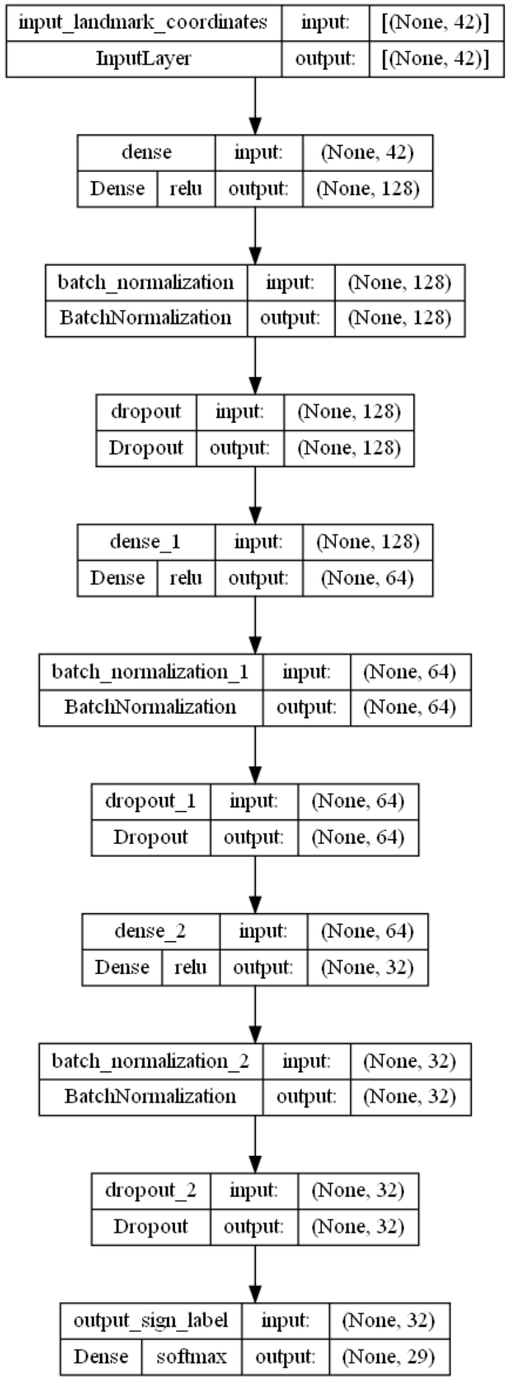
- the 26 letters from the American Sign Language (A-Z)
- 3 custom gestures:
  - **space:** place a space between the other gestures
  - **delete:** delete the most recently signed gesture
  - **send:** send a message in the chat system



INTERMEDIATE ARCHITECTURES







TECHNOLOGICAL FRAMEWORK

KERAS

High-level API for the TensorFlow library, that handles the entirety of the machine learning pipeline

MEDIAPIPE HANDS

Hand-tracking solution, that uses ML to recognize a hand from a frame and to infer 21 3D landmarks for it

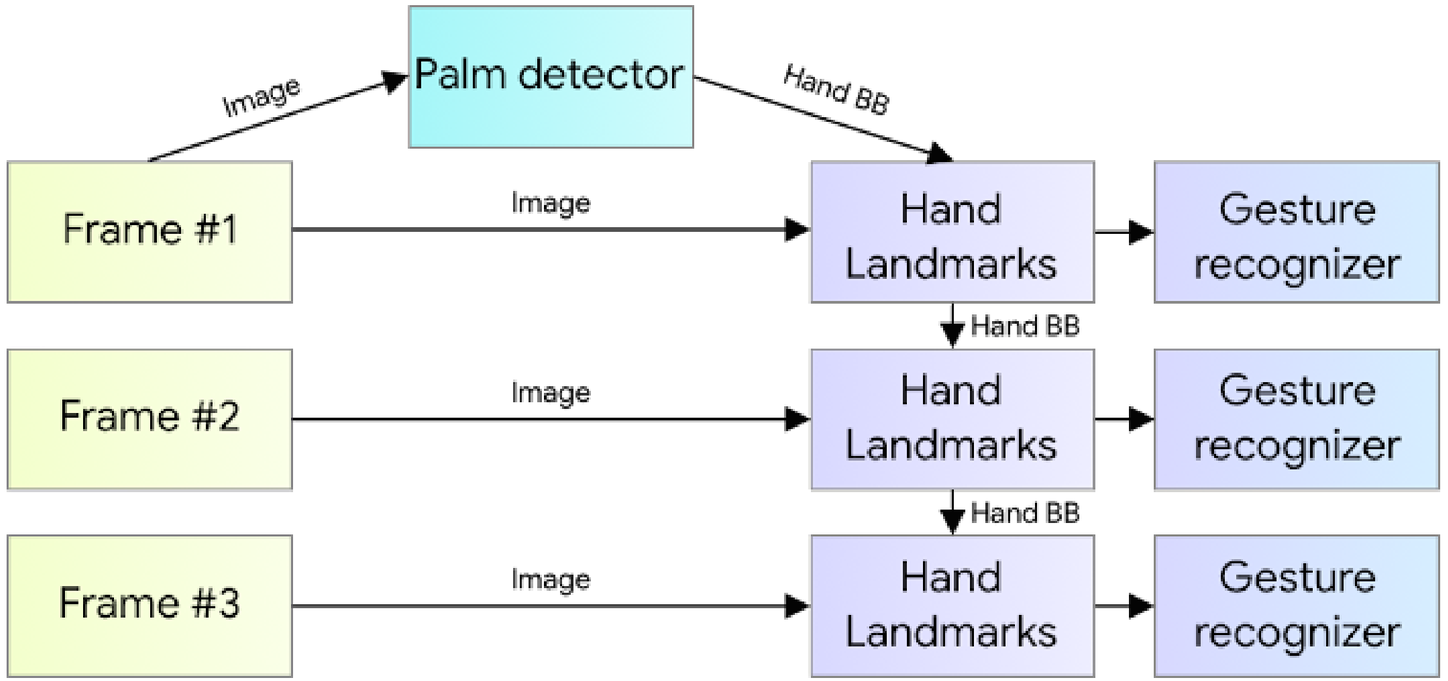
MEDIAPIPE HANDS

Palm Detection Model: BlazePalm

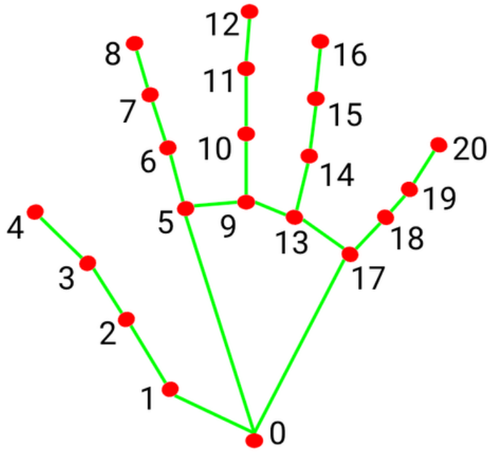
takes as input only a single frame and it can detect the hand location, surrounding the hand in a bounding box.

Hand Landmark Model

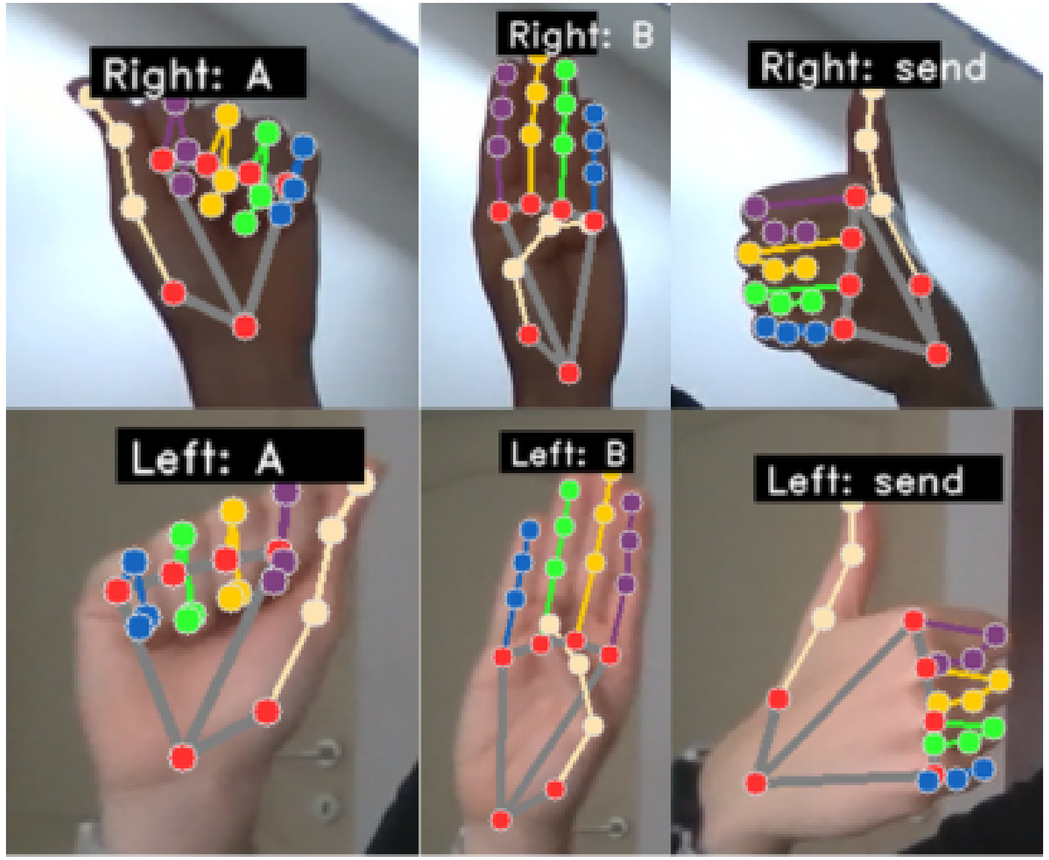
takes the cropped image of a hand and precisely locates the 21 3D landmarks of the detected hand



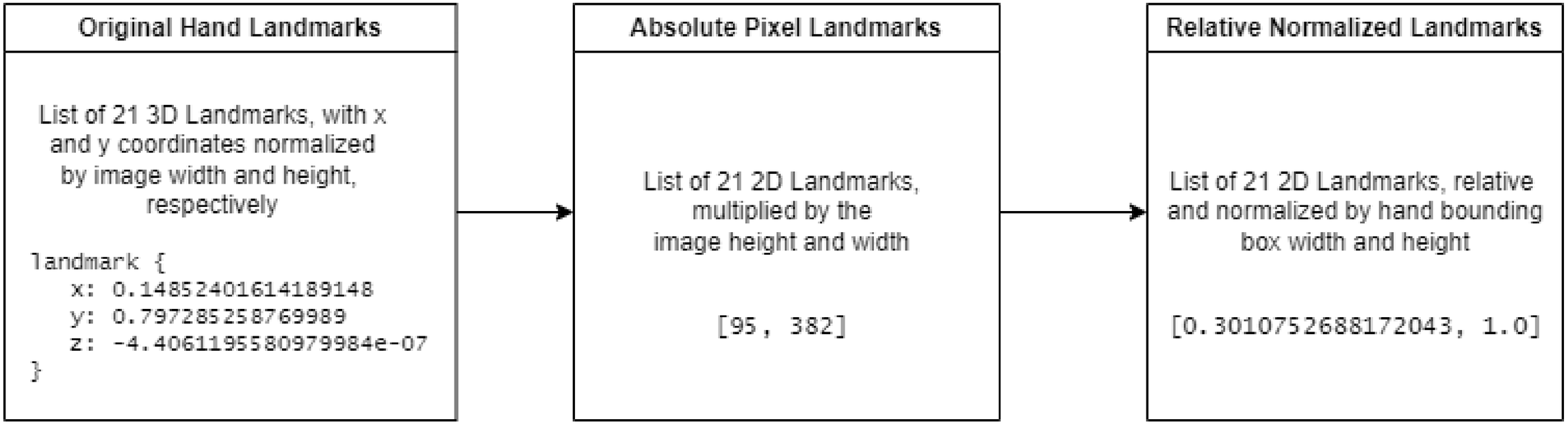
MEDIAPIPE HANDS



0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP



DATA PROCESSING FOR DATASET



RESULTS OBTAINED

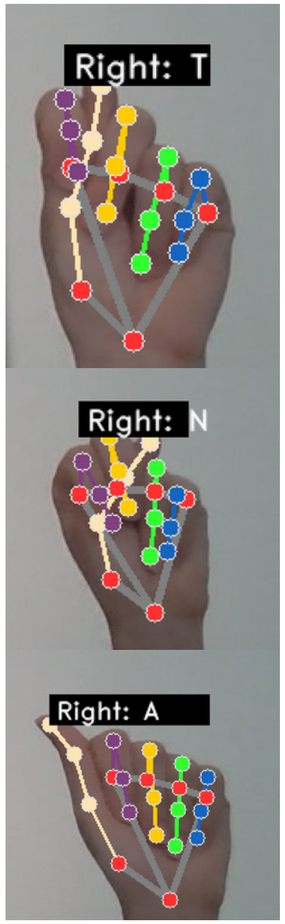
The Classification Report for the SLR model, including Precision, Recall, F1-score and Accuracy performance metrics.

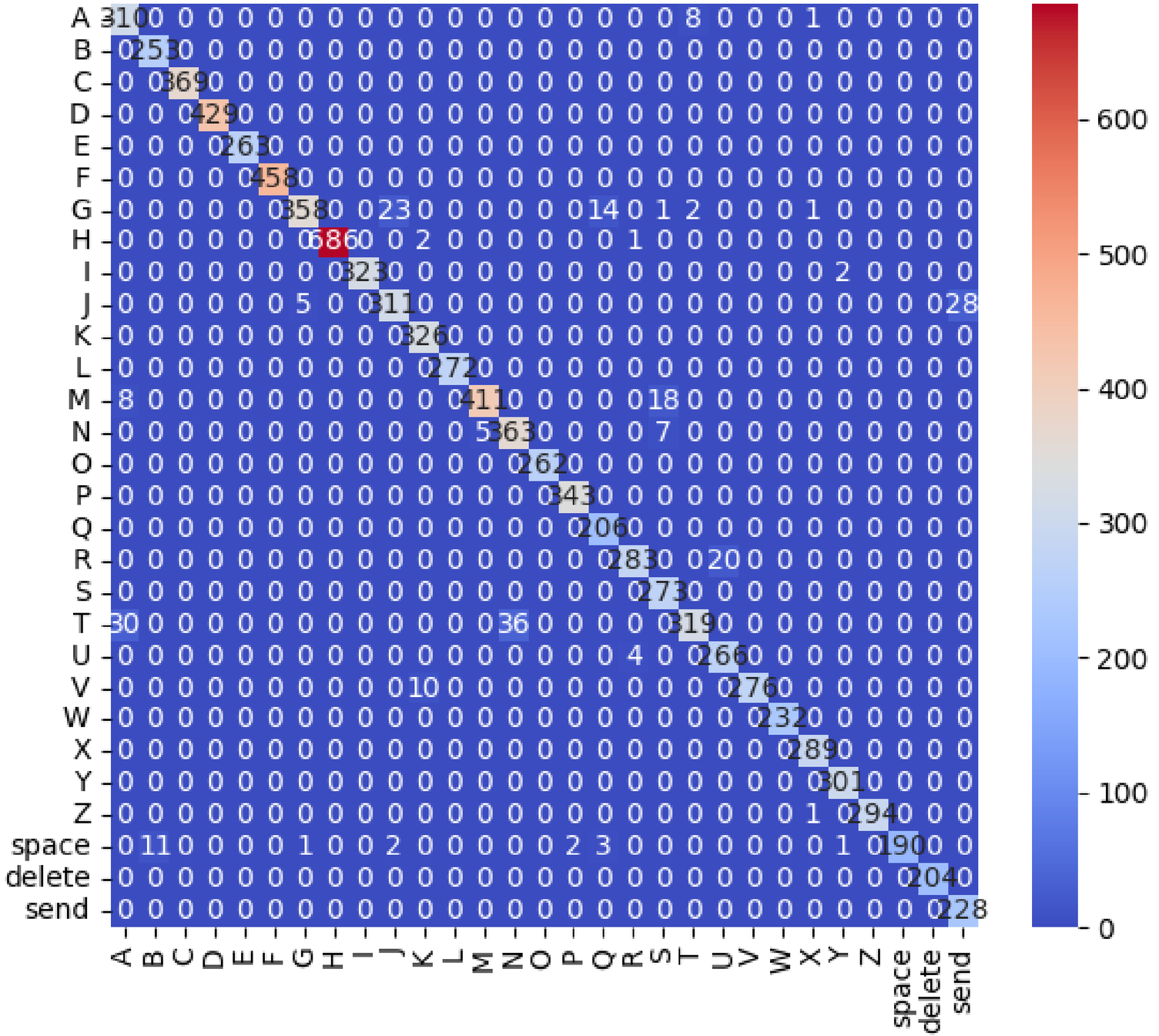
F1-score: [0.8559, 1.0000]

Table 6.2: Classification Report for Final SLR Model on Unseen Data

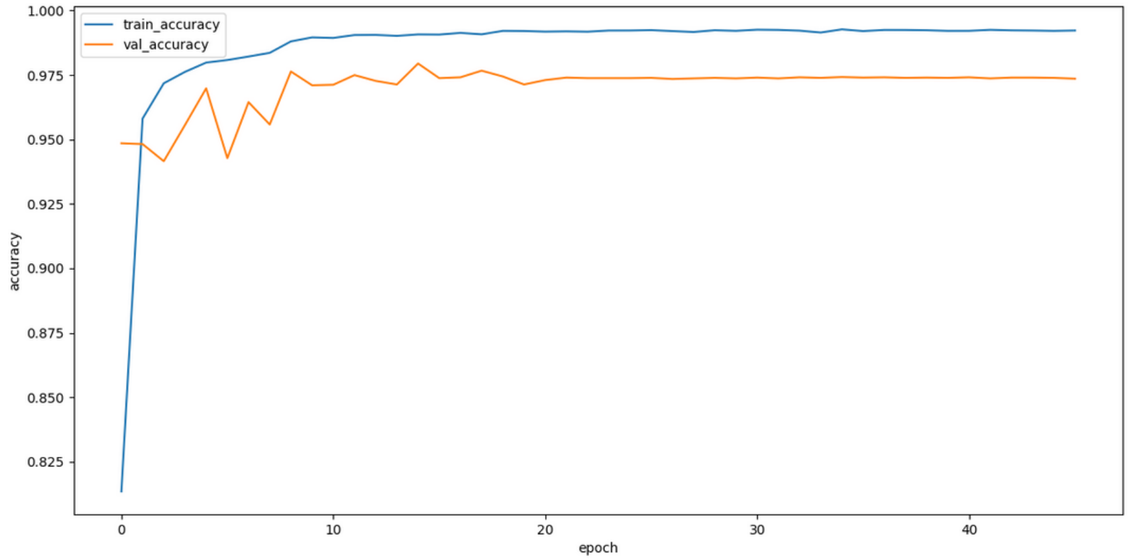
Class	Precision	Recall	F1-Score	Support
A	0.9408	0.9969	0.9680	319
B	0.9731	1.0000	0.9864	253
C	1.0000	0.9946	0.9973	369
D	1.0000	1.0000	1.0000	429
E	1.0000	1.0000	1.0000	263
F	1.0000	1.0000	1.0000	458
G	0.9922	0.9599	0.9758	399
H	1.0000	0.9971	0.9985	689
I	1.0000	0.9969	0.9985	325
J	0.9937	0.9157	0.9531	344
K	0.9702	1.0000	0.9849	326
L	1.0000	1.0000	1.0000	272
M	0.8401	0.9497	0.8915	437
N	0.9664	0.7680	0.8559	375
O	0.9924	1.0000	0.9962	262
P	0.9971	1.0000	0.9985	343
Q	0.9671	1.0000	0.9833	206
R	0.9676	0.9868	0.9771	303
S	0.9010	1.0000	0.9479	273
T	0.9944	0.9195	0.9555	385
U	0.9962	0.9704	0.9831	270
V	1.0000	0.9650	0.9822	286
W	0.9957	1.0000	0.9978	232
X	0.9633	1.0000	0.9813	289
Y	0.9901	1.0000	0.9950	301
Z	1.0000	1.0000	1.0000	295
space	1.0000	0.9381	0.9681	210
delete	1.0000	1.0000	1.0000	204
send	0.8941	1.0000	0.9441	228
accuracy		0.9758		9345
macro avg	0.9771	0.9779	0.9766	9345
weighted avg	0.9774	0.9758	0.9756	9345

CONFUSION MATRIX

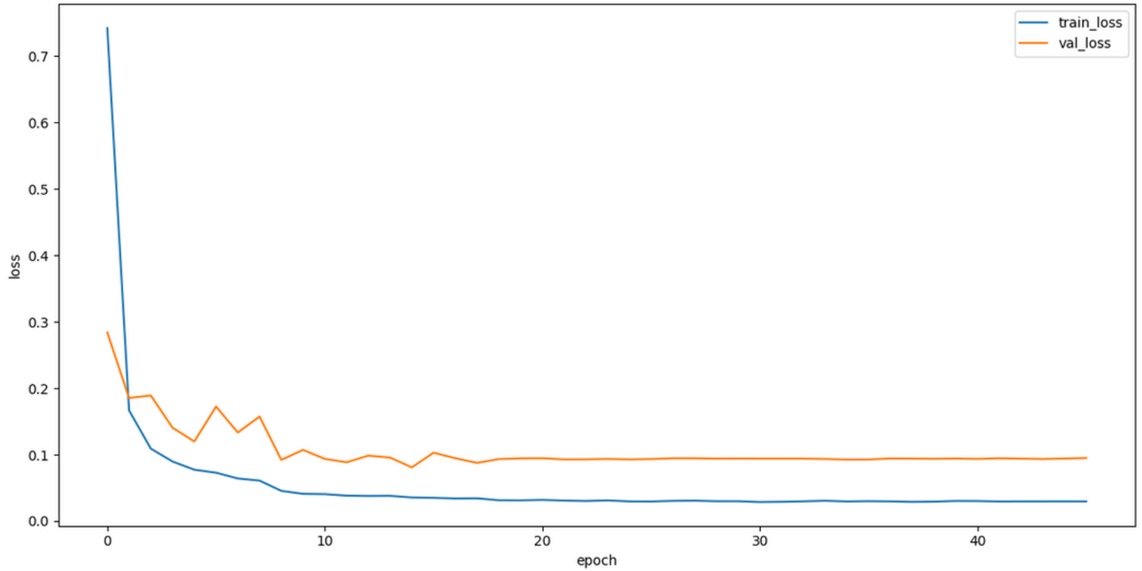




EVOLUTION OF ACCURACY AND LOSS



training and validation accuracy



training and validation loss

Accuracy: **97.58%**

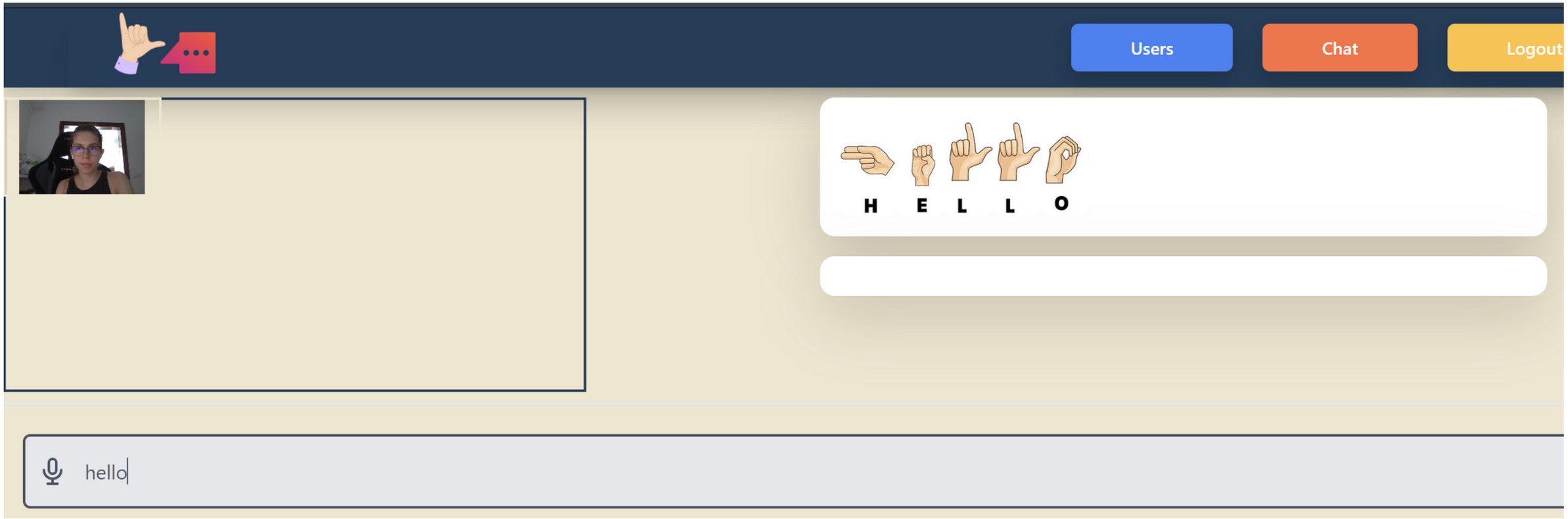
Loss: **0.09498**

INTEGRATION OF SLR MODEL INTO WEB APPLICATION

Steps:

- converting the **Keras** model into a TypeScript compatible format using the **tensorflowjs** library;
- loading the model into the frontend of the web application using an Angular service;
- make predictions on input data processed from video capture;
- display the predicted letter/gesture.

CONVERSION OF TEXT INTO SIGN LANGUAGE

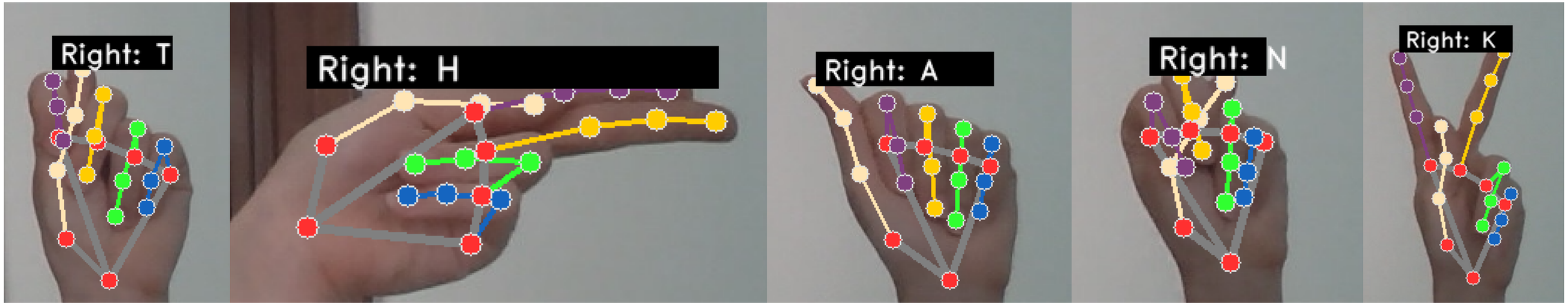


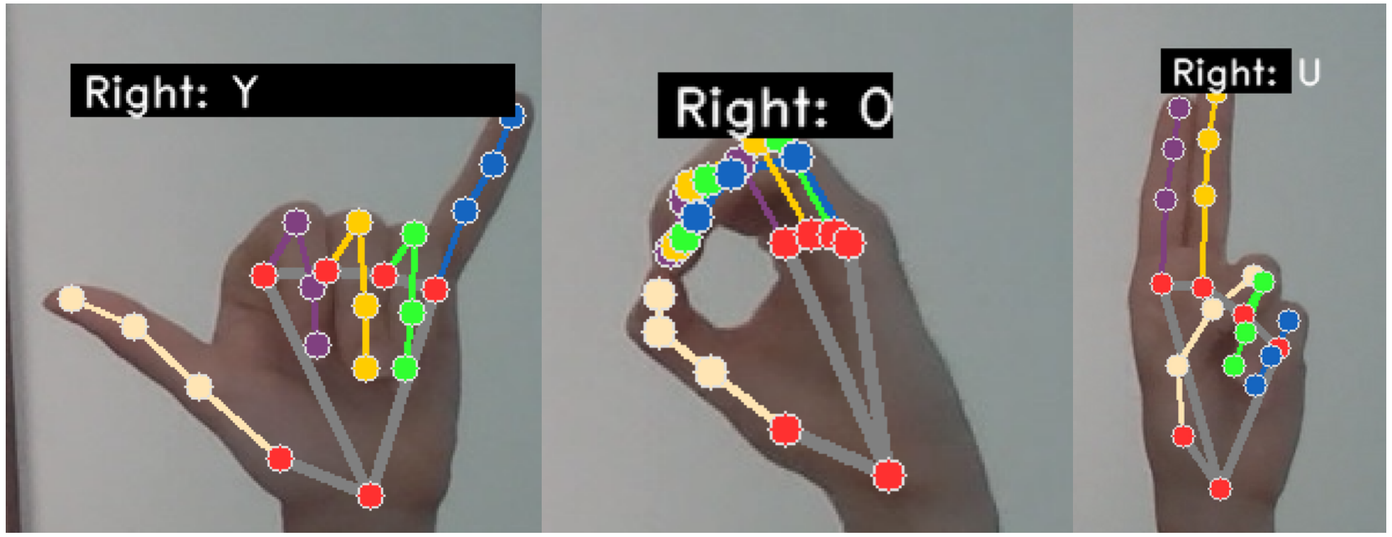
CONCLUSION

Further development:

- cover multiple sign languages, not just ASL;
- recognize a more complex vocabulary;
- classify dynamic signs, using a LSTM model;
- translate the written messages with entire words, not letter-by-letter.

This project is meant to be a step towards creating a more inclusive society, where technology is used as a means to decrease communication barriers.


Right: T
Right: H
Right: A
Right: N
Right: K


Right: Y
Right: O
Right: U