Sahaayi-Malayalam Sign Alphabet Recognition Application

Final Project Presentation - GROUP 1

21- Artha P Satheesh
28- Fadia Haris
47- Shahnavas K M
55- Sukritha K K

Guided By: Prof. Bincy Antony
Department of Computer Science and Engineering
Government College of Engineering Kannur,Kerala-670653

February 15, 2023

Outline

Introduction

- In a predominantly aural society, sign language users are often deprived of effective communication.
- Applications for recognition of previously formed sign languages are already available.
- National Institute of Speech and Hearing(NISH) had recently introduced Signs for Malayalam letters.
- Aiding this, we came up with an application that recognizes these Malayalam sign letters.

Motivation

Figure: Indian Sign Language Alphabet in Malayalam - Released

Problem Statement

To build a Mobile Application that can assist the deaf and mute to translate words, while communicating with others using uniform Indian Sign Language Alphabet in Malayalam.

Objective

- To bridge the gap between hearing impaired and normal people for effective communication.
- Dataset Creation
  To create a complete data set for Malayalam Sign Alphabet's.
- Build a mobile application that is been portable for them to be used in a standalone device.
- Conversion of signs to text.

Literature Review

Author	Title	Methodology	Advantage & Disadvantage	Challenges
Brandon Garcia and Sigberto Alarcon Viesca	Real-time American Sign Language Recognition with Convolutional Neural Networks [1].	Based on CNN. Utilized a pre-trained GoogLeNet architecture trained on the ILSVRC2012 dataset. Transfer learning approach is used.	Able to produce a robust model for letters a-e, and a modest one for letters a-k (excluding j). Fail to recognize all the letters.	Lack of variation in the dataset used.
Archana S. Ghotkar1 and Dr. Gajanan K. Kharate	Study of vision based hand gesture recognition using indian sign language[2].	Hand tracking and segmentation- HSV and YCbCr color model. Feature extraction- Orientation Histogram, Neural network, Complex background- 92% accuracy.	Gave an overall glimpse of SL interpretation need. Fail to give an appropriate solution to overcome the problem of scarce dataset.	Availability of standard dataset.

Literature Review

Author	Title	Methodology	Advantage & Disadvantage	Challenges
Herman Gunawan, Narada Thiracitta, Ariadi Nugroho	Sign Language Recognition Using Modified Convolutional Neural Network Model [3].	Makes use of i3d inception model.	Got 100% accuracy on training with 10 words and 10 signers with 100 classes but the validation accuracy is pretty low. This model is too overfit.	Need to overcome the problem of over fitting, by any of the following- , like freeze the layers, remove some inception module, remove the transfer learning, and change the fully connected layer into another deep learning model.
Kumud Tripathi, Neha Baranwal, G. C. Nandi.	Continuous Indian Sign Language Gesture Recognition and Sentence Formation[4].	Gradient based key frame extraction method - Recognizing a sign language gestures from continuous gestures. Feature extraction (OH). Dimension reduction(PCA).	Different distance based classifiers are used for testing. Satisfactory results with Euclidean distance and correlation.	Later there was a need for creating dataset with different background and different illumination conditions.

System Architecture

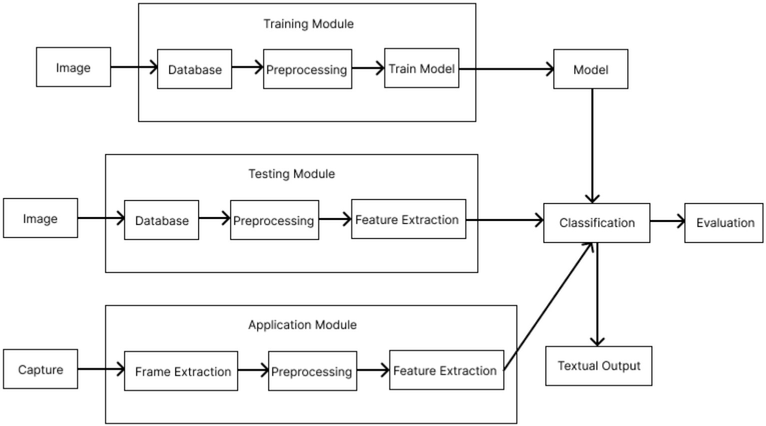


Figure: General Architecture

Signs

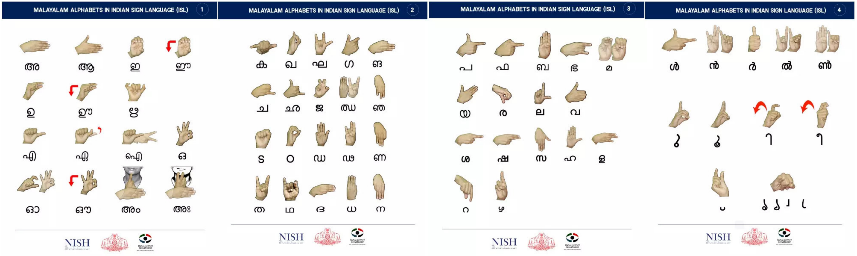


Figure: Malayalam Alphabets in Indian Sign Language

Dataset Creation

- There was no previous dataset for Malayalam Sign Alphabets.
- Signs include single hand or both the hands and it could be static or dynamic signs.
- 24600 images were created.
- The dataset was created using opencv.
- To increase the quality of images, images were collected from phone by connecting it to laptop through IP Address.

```
cap = cv2.VideoCapture("https://192.168.43.1:8080/video")
```

Dataset Creation

```
if start:
    roi = frame[100:800, 100:800]
    save_path = os.path.join(IMG_CLASS_PATH, '{}.jpg'.format(count + 1))
    cv2.imwrite(save_path, roi)
    count += 1
```

Figure: Dataset Creation

- A Region Of Interest was defined to capture image.
- cv2.imwrite() method in opencv is used to save an image to the device.

Dataset Created

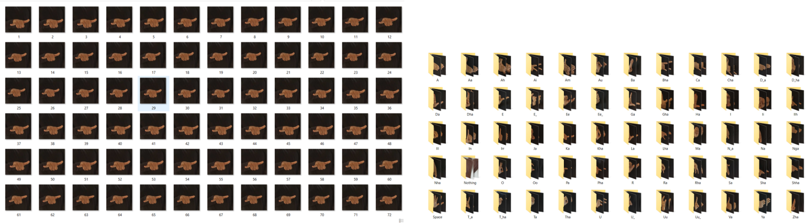


Figure: Dataset Created

Database - https://drive.google.com/drive/folders/11fxq9GOKa_QQzcrdBbRgVjBQ3hjiFkN2

Challenges in Dataset Creation

- Labelling using Malayalam alphabet.
- Difficulty in mapping the data.
- Use of dynamic sign made it difficult for representing the sign for image classification.
- Ambiguity within the signs proposed by NISH.
- Misclassification due to similarity in signs.
- Repetition of signs.

Challenges in Dataset Creation

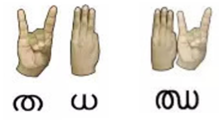


Figure: Example of ending position resembling another letter

Figure: Single sign for different symbols

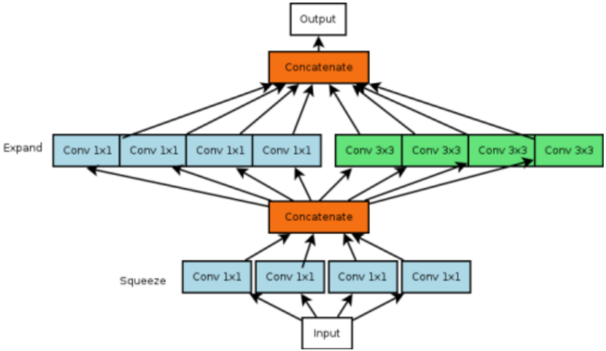
Challenges in Dataset Creation

ത - ഘ
ഋ - ക
സ - ധ
ങ - ദ
ൻ - ൽ
ജ - മ
അ - ദ
പ - ൾ
ഫ - ശ
◌ു - ◌ൂ

Figure: Conflicting Signs

Model Training

- A pretrained CNN model named Squeezenet was used for training.
- The primary objective of the squeezenet architecture is to reduce the number of parameters, and in turn the size of the network



Pre-processing

- Images collected in dataset are pre-processed inorder to fit into the squeezenet architecture for training.

```
image_path = os.path.join(path,file_name)
# prepare the image
img = cv2.imread(image_path)
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
img = cv2.resize(img, (227, 227))
```

Figure: Pre-processing

Model Parameters

```
def get_model():
    model = Sequential([
        SqueezeNet(input_shape=(227, 227, 3),
        include_top=False),
        Dropout(0.5),
        Convolution2D(NUM_CLASSES, (1, 1),
        padding='valid'),
        Activation('relu'),
        GlobalAveragePooling2D(),
        Activation('softmax')
    ])
    return model
```

Figure: Model Parameters

Model Parameters

- **Dropout** - Added to reduce overfitting
- **Convolution2D** - To control the size of the layer by appending to the already defined network
- **Activation(ReLU)** - Rectified Linear Unit turns negative values into 0 and outputs positive values
- **GlobalAveragePooling2D** - performs classification and calculates average out- put of each feature map in previous layer. (i.e data reduction layer)
- **Input shape** - Is an image size of 227 x 227 pixel
- **Include top** - lets to select if you want a final dense layer or not.

Training

```
model = get_model()
model.compile(
    optimizer=Adam(lr=0.0001),
    loss='categorical_crossentropy',
    metrics=['accuracy']
)
```

Figure: Training

- Adam optimizer is used with a learning rate of 0.001.
- Categorical crossentropy used for the loss function.
- A metric is used to judge the performance of the model with accuracy set as class.

Model Evaluation

- The model is tested using the test dataset. Based on it the predictions, the learning rate and epoches are adjusted to increase accuracy.
- Alphabets Aa, Ai, E, Ka, La, Rha, Sha, Tha were correctly predicted all the time.
- Alphabets Bha, Oo, T ha, Va, Zha were correctly predicted most of the time. Rest of the alphabets were rarely predicted.
- 10 images for each label, summing up to 600 images were taken for testing.

Application Development

- The Mobile application named SAHAAYI was developed using Flutter.
- Initial user interface and UX was designed using Figma.
- The application consist of two screens.
- The two main plugins used are : Camera plugin(camera: 0.9.8+1) and tflite(tflite: 1.1.2).
- The model created was integrated with the app using a plugin tflite.

Application Screens

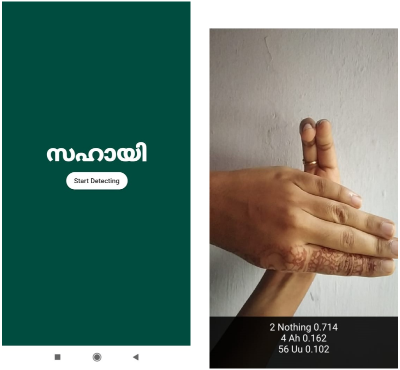
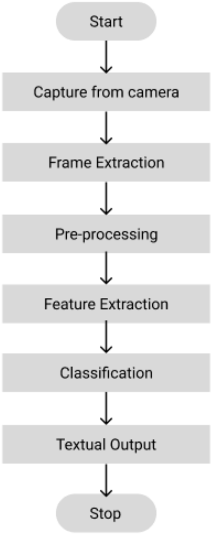
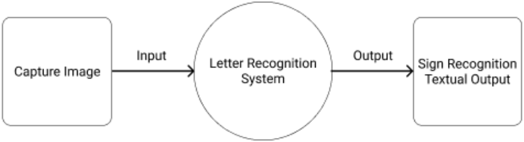


Figure: Sahaayi Application

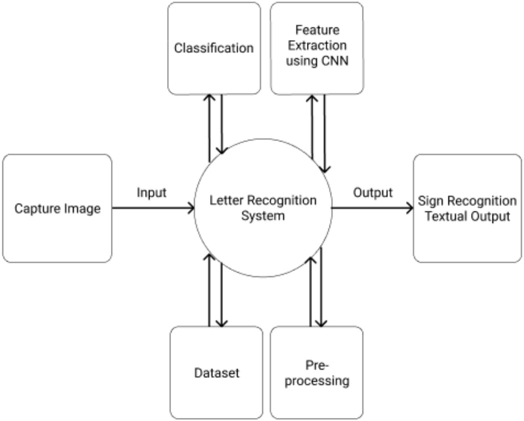
Flowchart Diagram



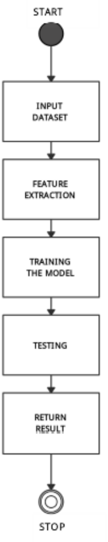
Dataflow Diagram Level 0



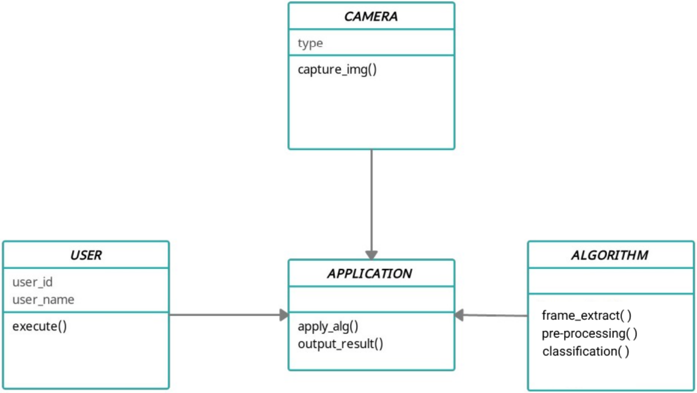
Dataflow Diagram Level 1



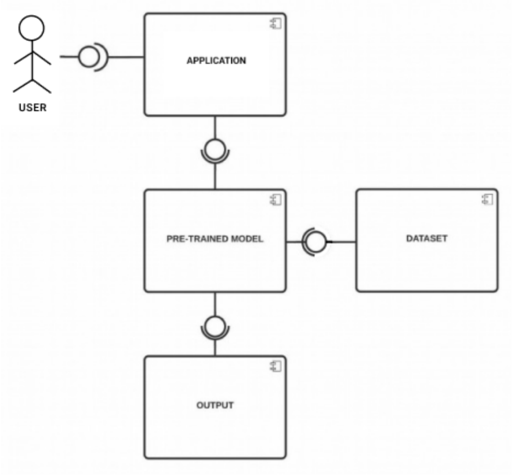
Activity Diagram



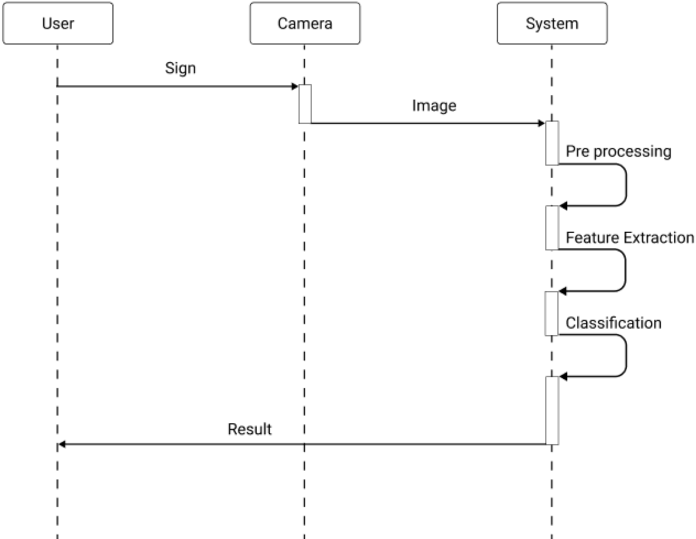
Class Diagram



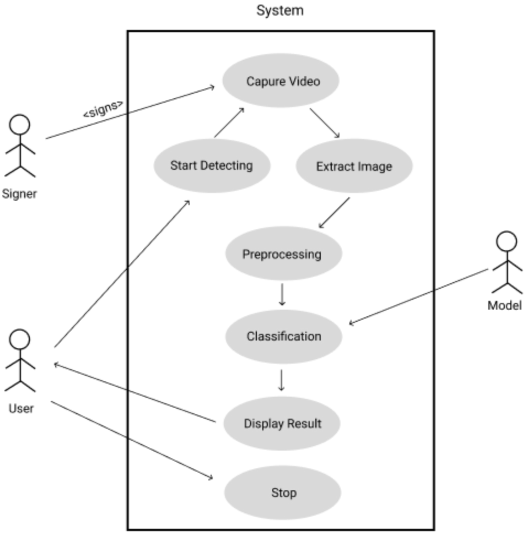
Component Diagram



Sequence Diagram



Usecase Diagram



Programming Tools

- Python 3.8- Programming language used
- OpenCV- provides an optimized Computer Vision libraries, tools, hardware, and supports model execution.
- Tensorflow- Software library for training and inference of deep neural networks
- Sklearn (Scikit-Learn)- Software ML library that provides classic machine learning models and evaluation metrics.
- Google Colab- Used for getting GPU support while training.
- Keras - Deep learning API written in Python.
- Flutter - Used for application development.

Results

- Dataset
  A dataset of twenty four thousand images were created, 400 images for each alphabet.
- Mobile Application
  Created Sahaayi Mobile Application using Flutter working both in IOS and Android devices.
- Smooth integration of the model to the application.
- Some of the letters correctly predicted under favourable background conditions.

Result

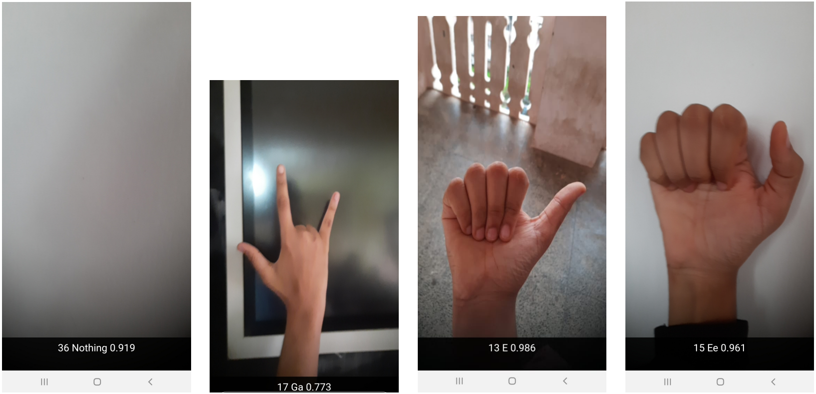
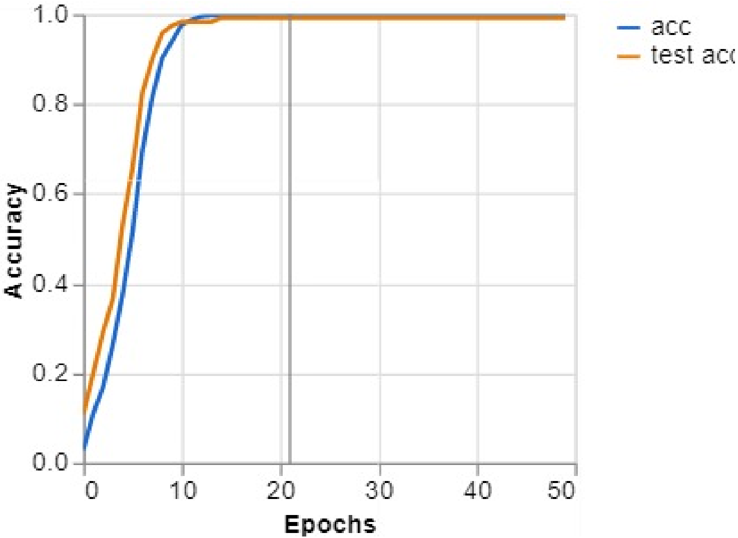
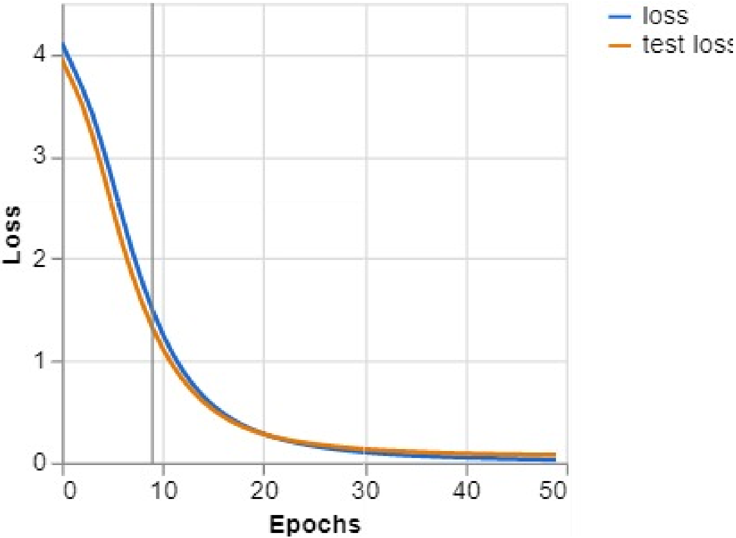


Figure: Example of correct recognition

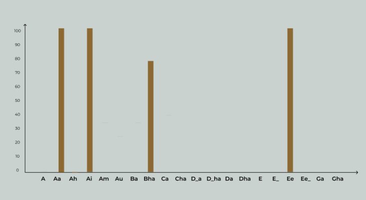
Accuracy VS Epoch Plot



Loss VS Epoch Plot



Test Result Plot



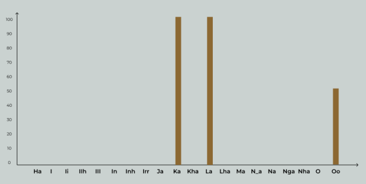
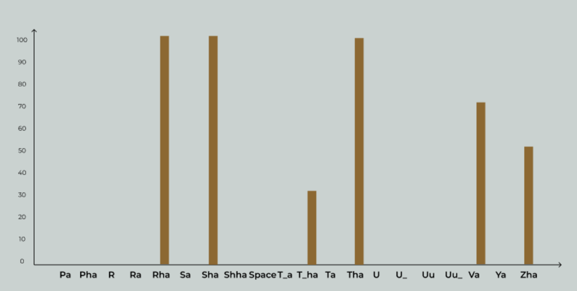


Figure: Test Result Plot

Test Result Plot



Confusion Matrix

		Predicted Class		
		Positive	**Negative**	
Actual Class	**Positive**	True Positive (TP)	False Negative (FN) **Type II Error**	**Sensitivity** $\frac{TP}{(TP+FN)}$
	Negative	False Positive (FP) **Type I Error**	True Negative (TN)	**Specificity** $\frac{TN}{(TN+FP)}$
		Precision $\frac{TP}{(TP+FP)}$	**Negative Predictive Value** $\frac{TN}{(TN+FN)}$	**Accuracy** $\frac{TP+TN}{(TP+TN+FP+FN)}$

Figure: Confusion matrix

Confusion Matrix

Predicted

Actual		P	N	
	P	TP=7	FN=3	Sensitivity=0.7
	N	FP=0	TN=590	Specificity=1
		Precision = 1	Negative Predictive Value = 0.9	Accuracy=0.99

Confusion matrix of വ

Confusion Matrix

Actual \ Predicted	P	N	
P	TP=5	FN=5	Sensitivity=0.5
N	FP=32	TN=558	Specificity=0.94
	Precision = 0.13	Negative Predictive Value = 0.99	Accuracy=0.93

Confusion matrix of ഴ

Result Analysis

- Low accuracy when it comes to application.
- Fluctuation between letters like:

വ - ഘ

ള്ള - ര

സ - ദ

ണ - ദ

ത - ൽ

ജ - വ

Conclusion

- The application hardly recognizes the signs given as input.
- Most of the signs are similar thus the model merely classifies.
- The model is successfully built and integrated to the application.
- Challenges faced during the dataset creation were labelling the images using Malayalam, including dynamic symbols and Some letters showed resemblance in sign.
- The accuracy of the system is 20%, it may be due to less dataset used for model training.

Future Scope

- Can be integrated with video conferencing softwares like Skype, Google Meet etc.
- Improving the proposed application provides better accessibility and ease of communication.
- Can be extended by integrating Malayalam gesture recognition.
- In the future, the output of the system can be enhanced into delivering speech output.
- Developing the system into handling dynamic signs will elevate its application to more areas.

References

- Brandon Garcia and Sigberto Alarcon Viesca, Real-time American Sign Language Recognition with Convolutional Neural Networks, *1st International Cognitive Cities Conference (IC3). IEEE*,pp. 202–203,September(2018).
- Ghotkar, Archana and Gajanan, K and Kharate, Gajanan, Study of vision based hand gesture recognition using indian sign language, *International journal on smart sensing and intillegent system*,vol.7,pp.96-115,March(2019).
- Suharjito, Suharjito and Gunawan, Herman and Thiracitta, Narada and Nugroho, Ariadi, Sign Language Recognition Using Modified Convolutional Neural Network Model,*13th international conference on computing communication technologies (iccct)*,pp.1-5,September(2018).

References

- Kumud Tripathi, Neha Baranwal and G. C. Nandi, Continuous Indian Sign Language Gesture Recognition and Sentence Formation, *Procedia Computer Science*,vol.54,pp.523-531,December(2019).
- `https://opencv.org/about/` accessed on april 2022
- `https://https://flutter.dev/` accessed on may 2022
- `https://pub.dev/packages/tflite` accessed on may 2022
- `http://www.nish.ac.in/others/news/882-indian-sign-language\-alphabet-in-malayalam-released` accessed on march 2022.