REAL TIME DYNAMIC INDIAN SIGN LANGUAGE RECOGNITION

Problem statement

“The absence of real-time ISL recognition systems limits inclusive education for the hearing-impaired, due to a lack of structured datasets and underrepresentation in research.”

Literature Survey

“American sign language (ASL) recognition based on Hough transform and neural networks,”

"Real time Hand Gesture Recognition using different algorithms based on American Sign Language,"

, "Real-time American Sign Language Recognition using wrist-worn motion and surface EMG sensors,"

Learning from Literature

Dataset Description

Indian sign language research and training center (ISLRTC)

Dataset Challenges

- Extract frames
- Remove corrupted data
- Normalize frame sizes and lengths
- Blur noisy backgrounds
- Add padding for consistency
- Split the data into train and test sets

Data Pre-Processing

1. Frame Extraction and Cleaning

- All videos were broken down into frames of 30.
- corrupted, unreadable files and blank frames were removed.
- Pauses between signs in videos were removed.

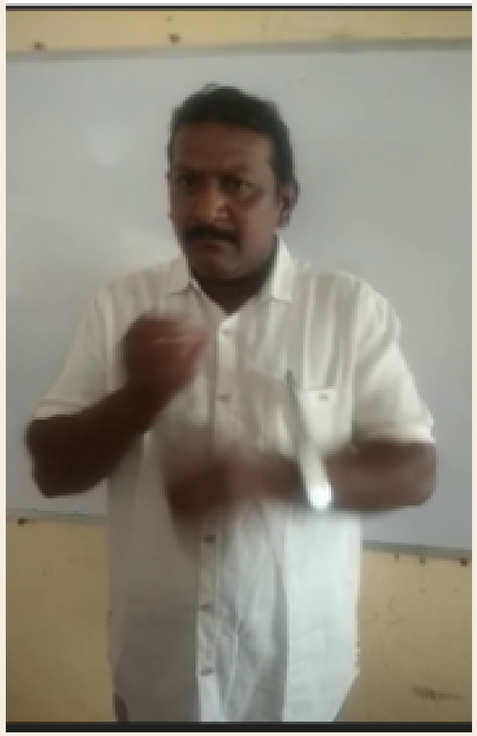
2. Resizing the frames and Normalizing

- All frames were first 64x64 pixels with ImageV2Net yielded low accuracy.
- 224*224 pixels with ImageV2Net yielded good accuracy.
- Better gradient flow.

3. Background and Noise Removal

- ISL is largely focused on hand and upper-body movements.
- Lower body and noise needed to be removed.
- Color thresholding, blur, median filtering.

3. Background and Noise Removal



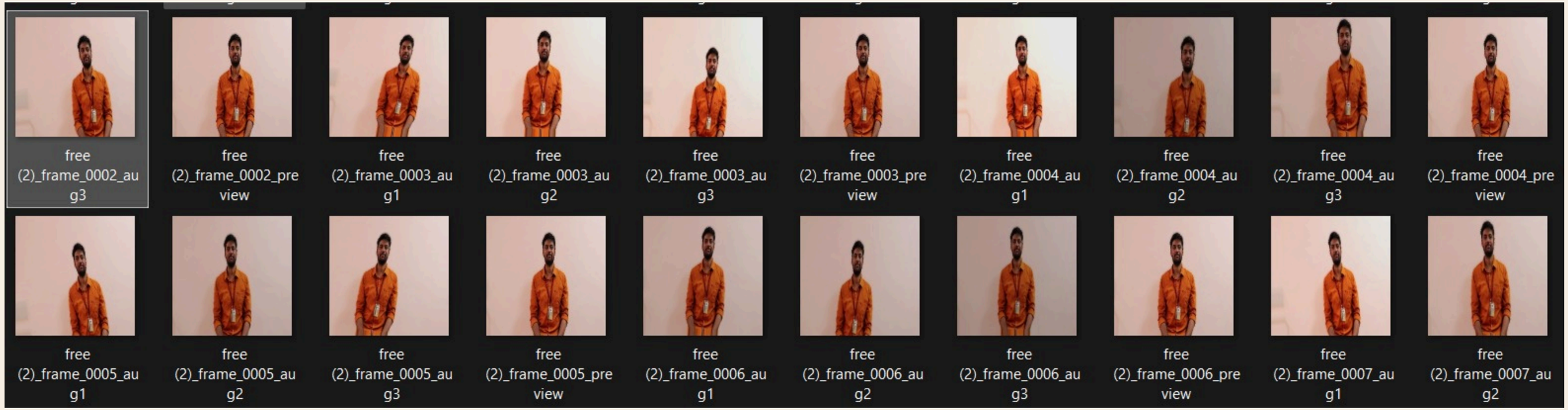
4. Data Augmentation and Generalization

- Very limited diversity. Causes overfitting.
- Random rotations, flips.
- Zoom and different scaling.
- Brightness and contrast adjustments.
- Backgrounds were blurred for all frames.

5. Frame Orientation

- Frames were rotated by 3-7 degrees.
- Better consistency.
- Better detection incase of slightly different orientations.

5. Frame Orientation



6. Label Encoding and Class Mapping

- Labeling our data for deep learning model.
  - x.npy
  - y.npy
  - classes.npy

Model Architecture and Working

Spatial info
Temporal dynamics

CNN

WHAT

WHY

Approach 1

Approach 2

LSTM(RNN)

WHAT

WHY

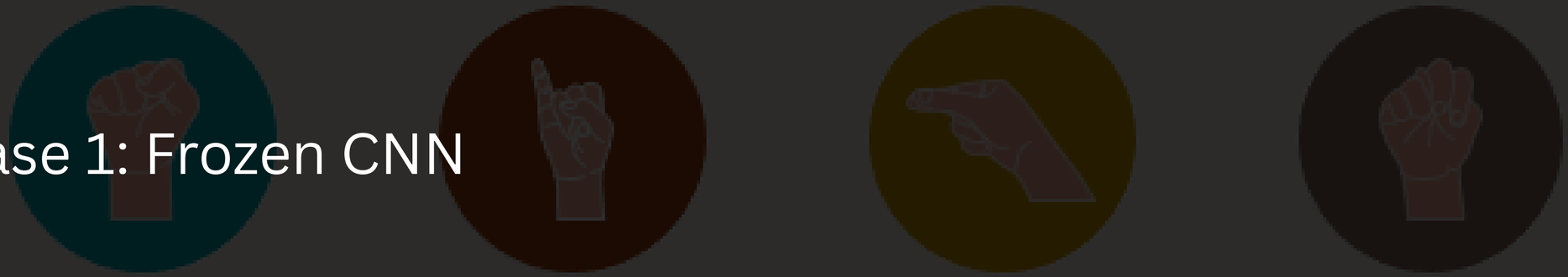
Model Architecture

- Feature Extraction
- Temporal Modeling
- Classification

Model Training

- Phase 1: Frozen CNN
- Adam
- Phase 2: Fine Tuning

Model Training

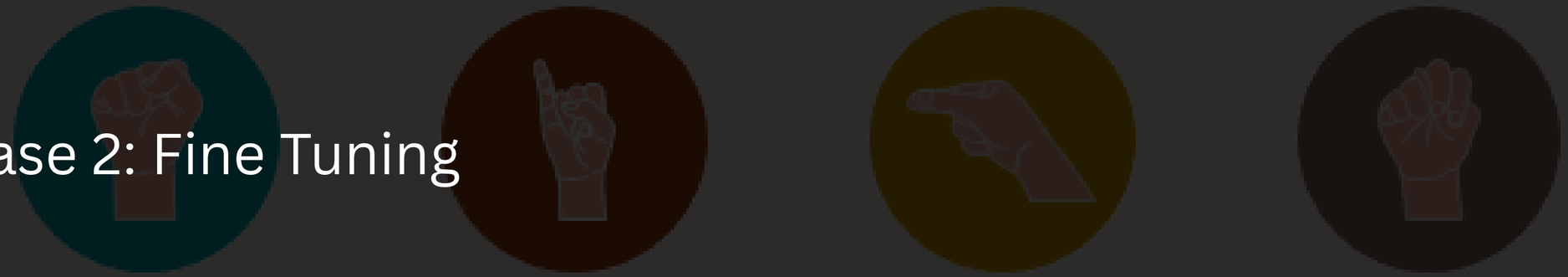


Phase 1: Frozen CNN

- MobileNetV2 Weights are frozen
- Temporal patterns learning as CNN trains on basic features in first phase, causing overfitting.
- Learning rate is higher.

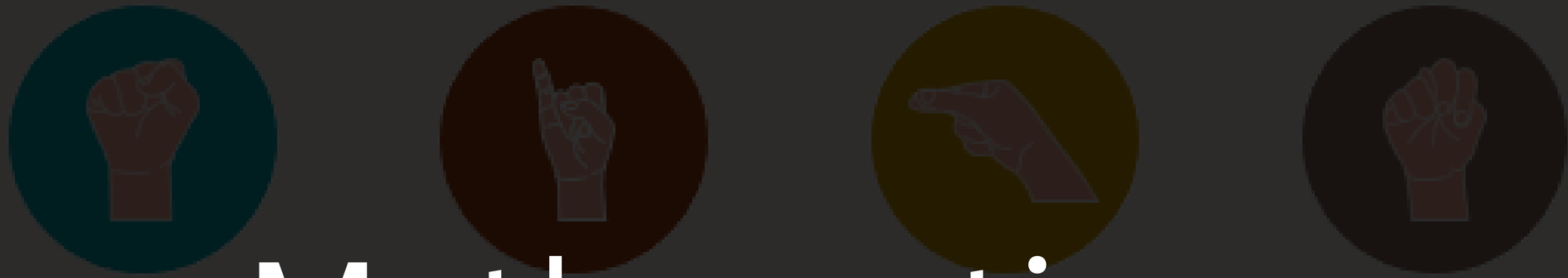
ADAM

Model Training



Phase 2: Fine Tuning

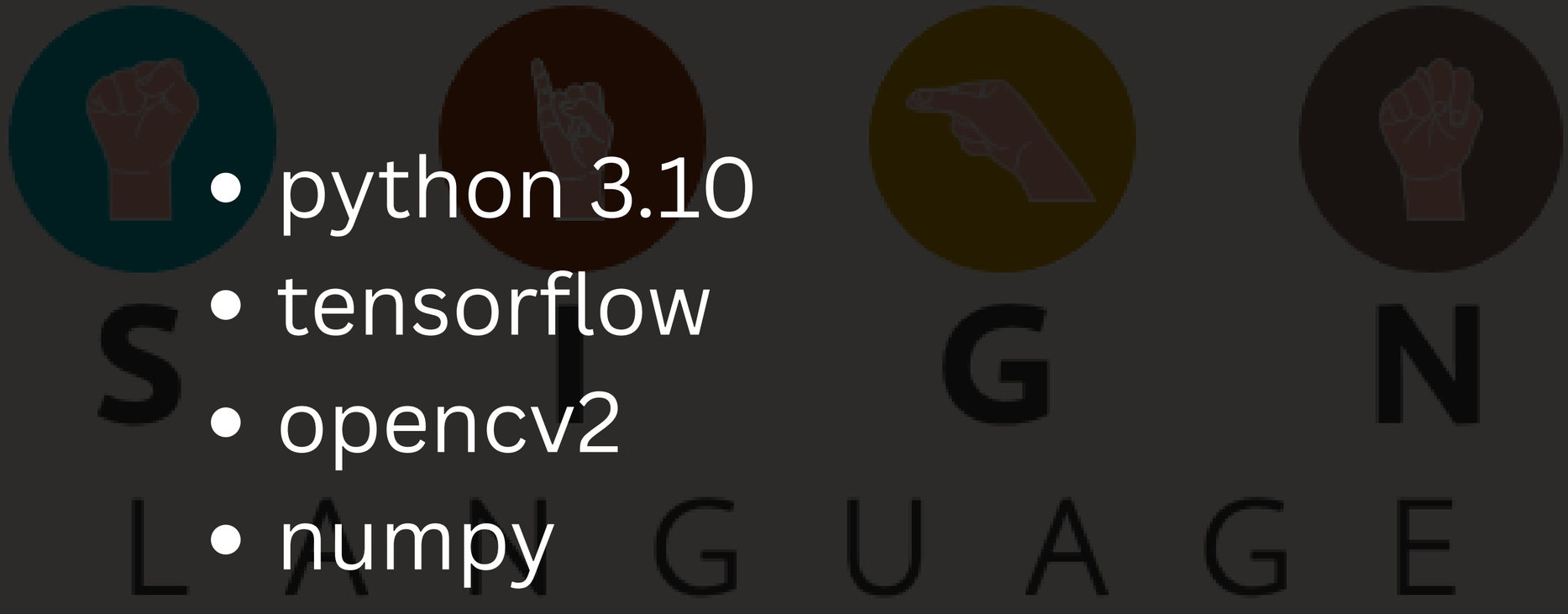
- CNN adapts feature extraction close to gesture-specific patterns.
- Hand and shape transitions are taken care of.
- Learning rate is lower as features are not basic.



Mathematics

SIGN LANGUAGE

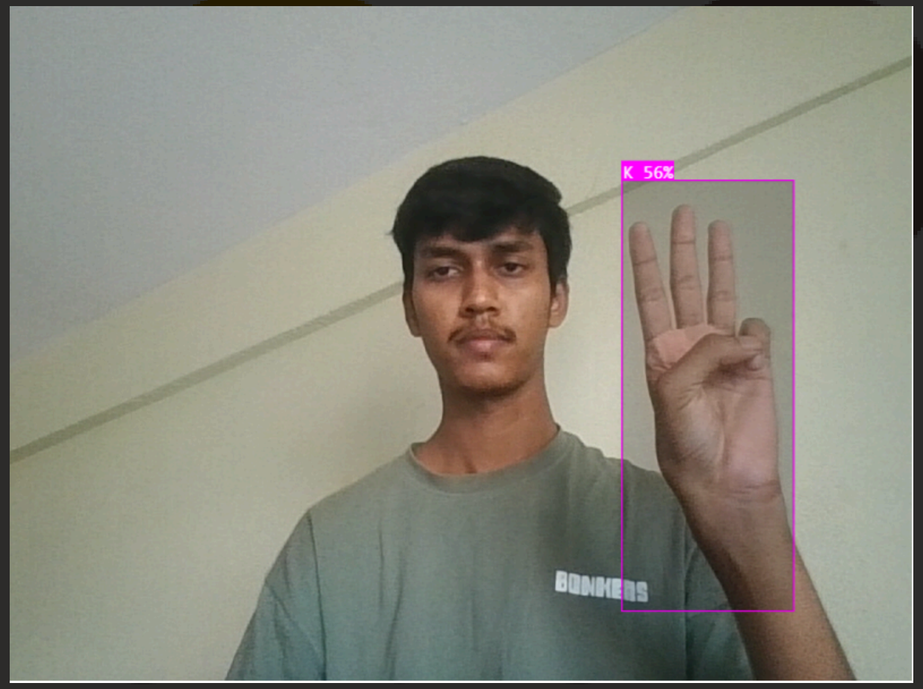
Development Environment



- python 3.10
- tensorflow
- opencv2
- numpy
- matplotlib

Results and Discussions

G



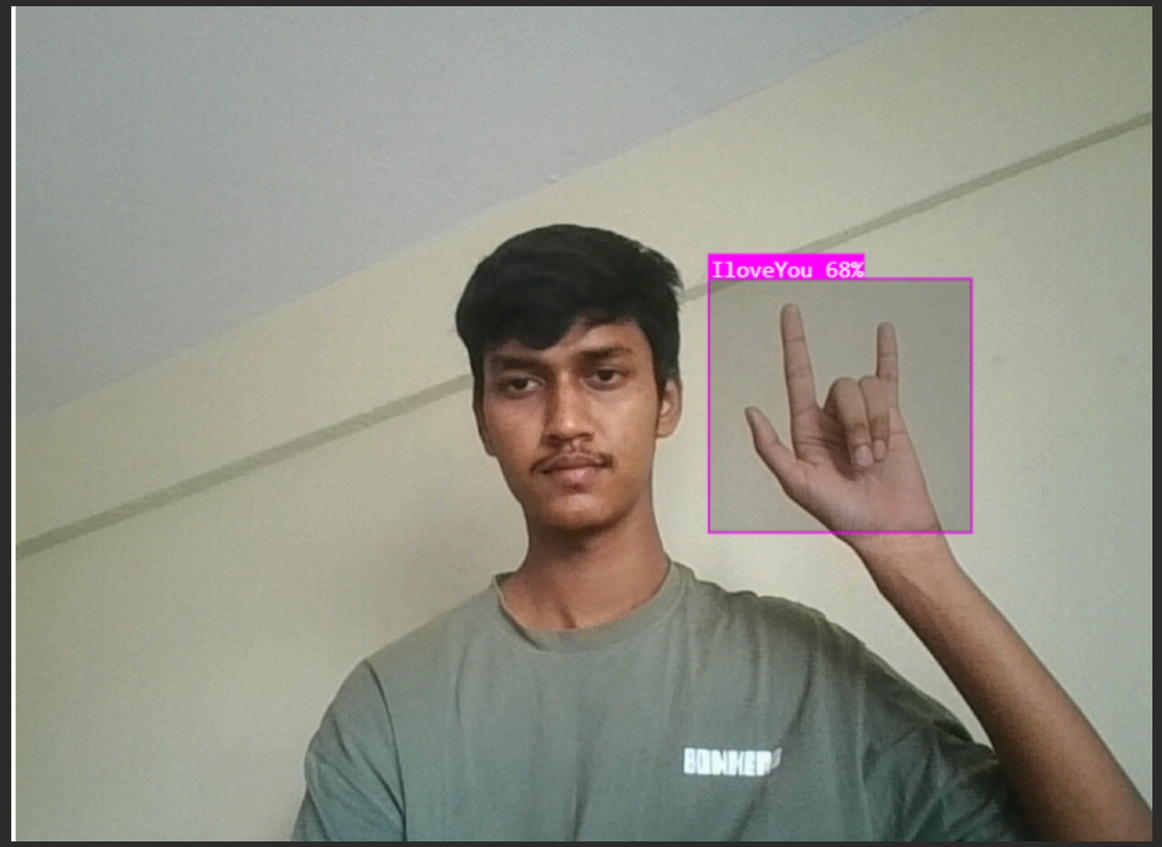
Results and Discussions

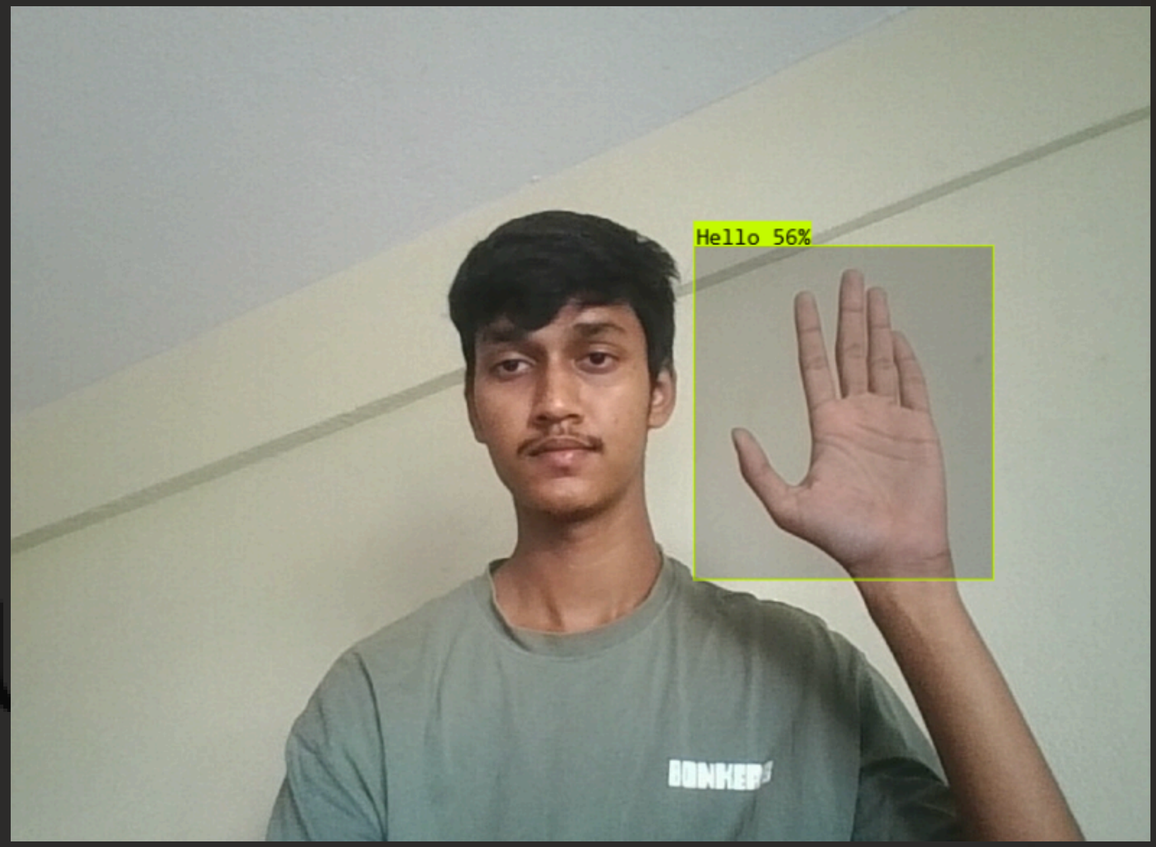
Class	Precision	Recall	F1-Score
A	0.99	0.96	0.99
B	0.98	0.96	0.98
C	0.97	0.98	0.97
X	1	0.97	0.99
Y	0.98	0.98	0.98
Macro Average	0.98	0.97	0.98
Accuracy			98.22%

Results and Discussions

	A	B	C	X	Y
A	98	2	0	0	0
B	2	98	0	0	0
C	0	0	96	4	0
X	0	0	4	96	0
Y	0	0	0	0	100

Results and Discussions

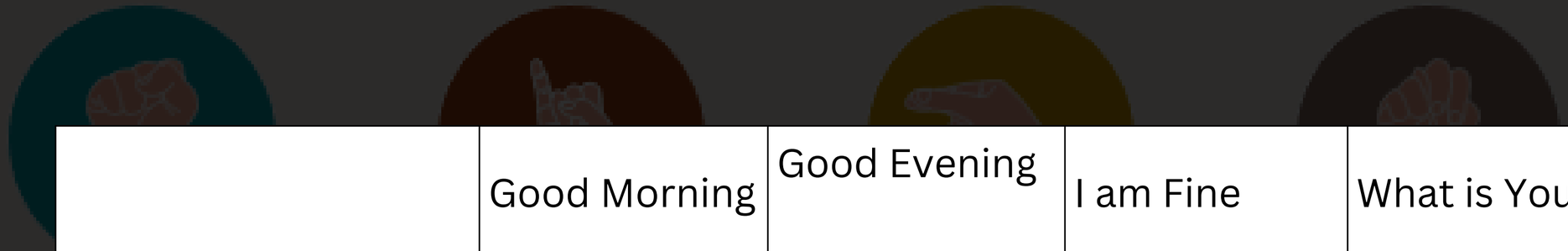




Results and Discussions

Class	Precision	Recall	F1-Score
Good Morning	0.74	0.72	0.73
Good Evening	0.71	0.69	0.7
I am Fine	0.76	0.72	0.77
What is Your Nam	0.73	0.78	0.7
Thank You	0.79	0.8	0.82
Macro Average	0.75	0.74	0.74
Accuracy			73%

Results and Discussions



	Good Morning	Good Evening	I am Fine	What is Your	Thank You
Good Morning	72	5	1	1	1
Good Evening	6	69	2	1	2
I am Fine	2	1	77	0	0
What is Your Name	1	1	0	74	2
Thank You	1	0	1	1	77

Novelty

LANGUAGE

Conclusions

LANGUAGE