

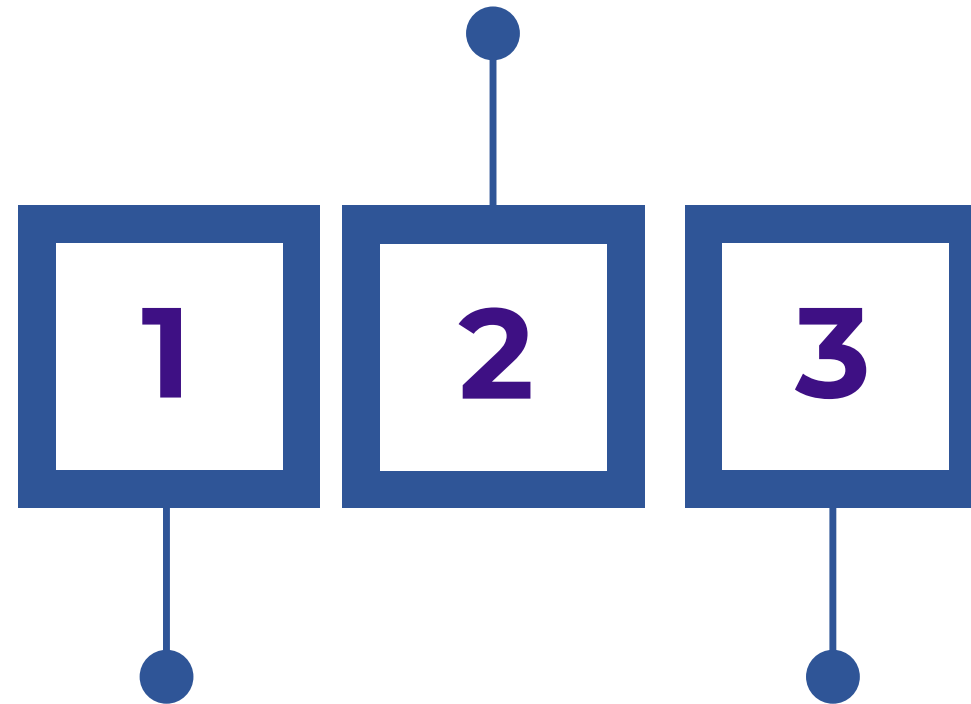
بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

# **ENHANCING HUMAN ACTIVITY RECOGNITION THROUGH IoT SENSOR DATA ANALYTICS**

**SUPERVISED BY: DR. MUHAMMAD DILSHAD SABIR**

# GROUP MEMBERS

M. Naeem Farooq  
FA21-BCE-052



Maheen Arshad  
FA21-BCE-035

Sabir Jan  
FA20-BCE-075

# OUTLINE

1

**INTRODUCTION**

2

**PROBLEM  
STATEMENT**

3

**LITERATURE  
REVIEW**

4

**BACKGROUND**

5

**PROPOSED  
METHODOLOGY**

6

**MODEL  
TRAINING**

7

**FUTURE WORK**

8

**EXPERIMENT  
DEMONSTRATION**



# 1. INTRODUCTION

# Human Activity and Facial Expression Recognition

- **Human Activity Recognition:**

Identifies actions like walking, sitting, running using sensor data.

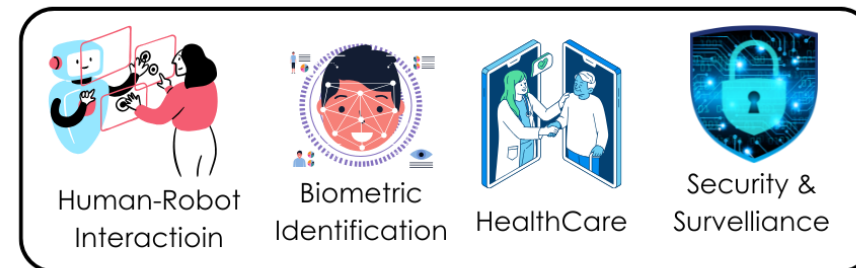


- **Facial Expression Recognition:** Detects emotions through body signals or expressions.



- **Applications of Facial Expression Recognition:**

- ✓ Healthcare
- ✓ Security
- ✓ Smart Homes



# Evolution of Emotion & Activity Recognition Techniques

## Traditional Methods

### 1. Physiological Sensors (EEG, ECG, GSR)

- **Strengths:** Accurate and reliable.
- **Limitations:** Invasive, lab-based, uncomfortable.

### 2. Behavioral Analysis (Face, Voice, Body)

- **Strengths:** Non-contact emotion/activity detection.
- **Limitations:** Affected by noise, lighting, and biases.

### 3. Self-Reports & Brain Scans (Surveys, fMRI)

- **Strengths:** Deep psychological insights.
- **Limitations:** Costly, subjective, not real-time.

## Emerging Approaches

### 1. Wearable Devices

- Less uncomfortable, easier to use in daily life.

### 2. Camera & Wi-Fi-Based Systems

- Can monitor emotions and actions from a distance, without any contact.

### ➤ Ongoing Challenges:

- Privacy and data security concerns in use of camera
- Wearable devices may be uncomfortable for some users



## 2. PROBLEM STATEMENT

# Problem Statement

Existing emotion and activity recognition methods have notable limitations. Traditional techniques like physiological sensors (EEG, ECG), behavioral analysis, and self-reports are often invasive, uncomfortable, expensive, and restricted to lab settings. Emerging solutions such as wearable devices and camera-based systems offer better usability but still raise concerns around privacy, cost, and user discomfort.

The ESP32 is a small, affordable device with untapped potential to recognize emotions and activities using Wi-Fi Channel State Information (CSI) in a completely contactless way. However, this approach remains underutilized.

There is a need for a simple, real-time, and privacy-friendly system that uses Wi-Fi CSI from the ESP32 to accurately detect human emotions and activities—addressing the limitations of both traditional and emerging methods.



## 3. BACKGROUND

## CSI Vs RSS

### RSS

- Received Signal Strength
- Measures the power of received signals.
- As RSS is not stable compared with CSI, it cannot properly capture dynamic changes in the signal while the activity is performed. [6]

### CSI

- Channel State Information
- Captures how Wi-Fi signals interact with the environment
- Describes **amplitude** and **phase variations** across multiple subcarrier frequencies as signals propagate from a transmitter to a receiver.

[6] Zhang, W.; Zhou, S.; Yang, L.; Ou, L.; Xiao, Z. Wi-FiMap+: High-Level Indoor Semantic Inference with Wi-Fi Human Activity and Environment. *IEEE Trans. Veh. Technol.* **2019**, 68, 7890–7903. [[Google Scholar](#)]

# Mathematical Model of CSI & Calculating Amplitude

## ➤ Frequency-Domain Model:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}$$

- $\mathbf{x}$ : Transmitted signal (complex vector).
- $\mathbf{y}$ : Received signal (complex vector).
- $\mathbf{H}$ : Channel information matrix (CSI).
- $\mathbf{n}$ : Noise vector.

## ➤ Amplitude Calculation:

$$A^{(i)} = \sqrt{(\mathbf{H}_{im}^{(i)})^2 + (\mathbf{H}_r^{(i)})^2}$$

## ➤ Phase Calculation:

$$\Phi^i = \text{atan2}(\mathbf{H}_{im}^{(i)}, \mathbf{H}_r^{(i)})$$

# Comparison of CSI Collecting Hardware

Device	Cost	Size (cm)	Weight
USRP (Ettus)	\$8,400	26.7 x 21.8	1.6 kg
Atheros AR9580 NIC	\$11	2.98 x 2.82	>1 kg*
Intel 5300	\$11	3 x 2.68	>1 kg*
ESP32	\$6	5.5 x 2.8	10g



ESP32

**Table 1.** Comparison of CSI collecting Hardware.

Cost-Effective

Compact Size

Lightweight

\*Note: Weight includes the computer.



# 4. LITERATURE REVIEW

## 4. LITERATURE REVIEW

Reference Paper	Author	Published	Limitation
Efficient Wi-Fi-Based Human Activity Recognition Using Adaptive Antenna Elimination [1]	Jannat, Mir and Islam, Md and Yang, Sung-Hyun and Liu, Hui	October 2023	Limited Emotion Analysis
Non-Contact Heart Rate Monitoring Method Based on Wi-Fi CSI Signal [2]	Sun J, Bian X	March 2024	High space occupation setup being used
Human Activity Recognition using Wireless Signals and Low-Cost Embedded Devices [3]	Tong, Thuan and Bui-Thanh, Binh and Nguyen T. H., Phuoc	July 2024	Low accuracy when test in different environment.
Directional Antenna Systems for Long-Range Through-Wall Human Activity Recognition [4]	Strohmayr, Julian and Kampel, Martin	October 2024	Low sampling rate used
RF sensing enabled tracking of human facial expressions using machine learning algorithms [5]	Hameed, Hira and Elsayed, Mostafa and Kaur, Jaspreet and Usman, Muhammad	October 2024	Highly controlled Environment utilized

[1] Jannat, M. a.-H. (2023). *Efficient Wi-Fi-Based Human Activity Recognition Using Adaptive Antenna Elimination*. IEEE Access, 105440-105454.

[2] Sun, J. a. (2024). *Non-Contact Heart Rate Monitoring Method Based on Wi-Fi CSI Signal*. Sensors. Retrieved from <https://www.mdpi.com/1424-8220/24/7/2111>

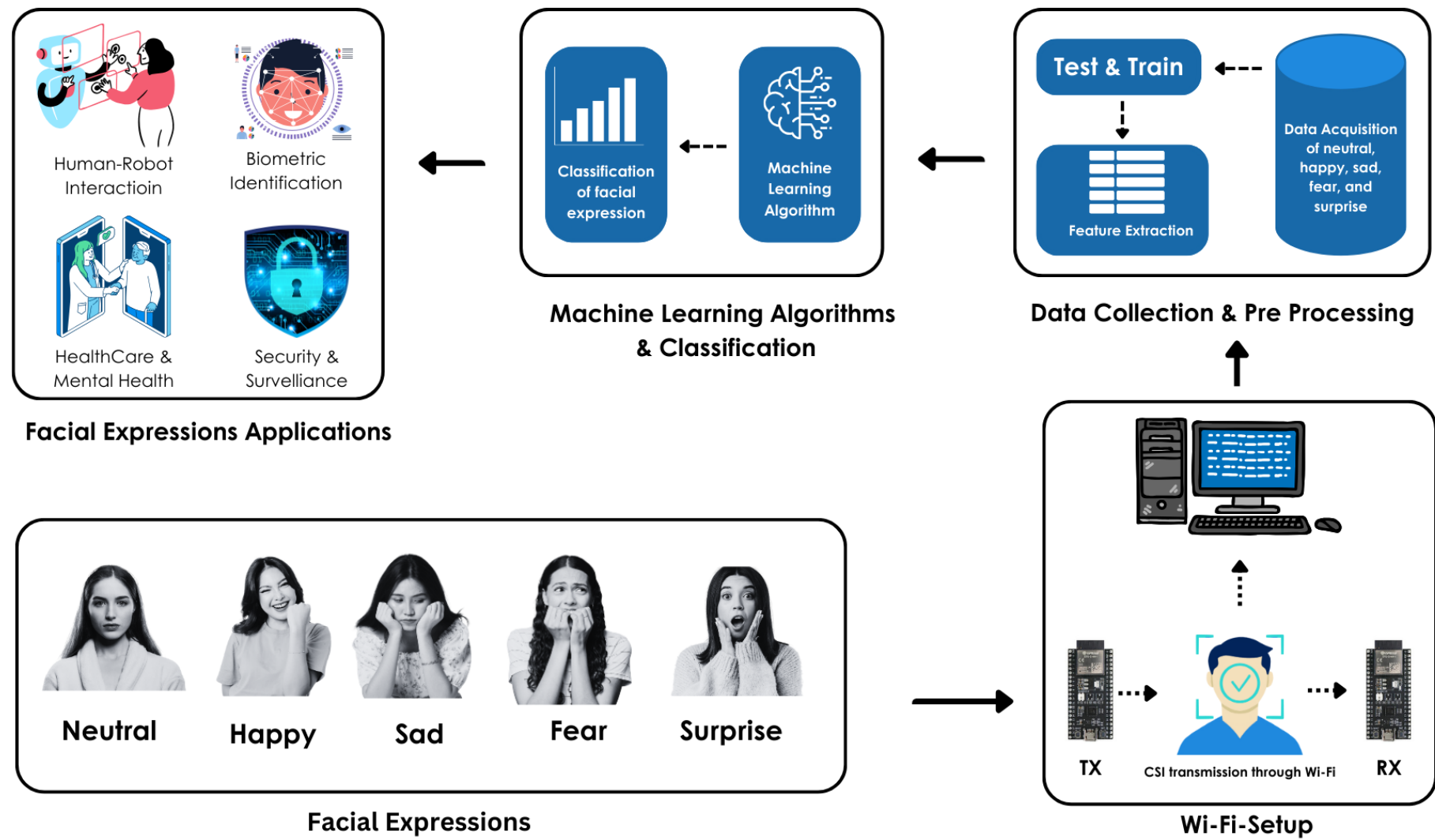
[3] Tong, T. a.-T. (2024). *Human Activity Recognition using Wireless Signals and Low-Cost Embedded Devices*. 7-12.

[4] Strohmayr, J. a. (2024). *Directional Antenna Systems for Long-Range Through-Wall Human Activity Recognition*. 3594-3599.

[5] Hameed, H. a. (2024). *RF sensing enabled tracking of human facial expressions using machine learning algorithms*. Scientific Reports. doi:10.1038/s41598-024-75909-w

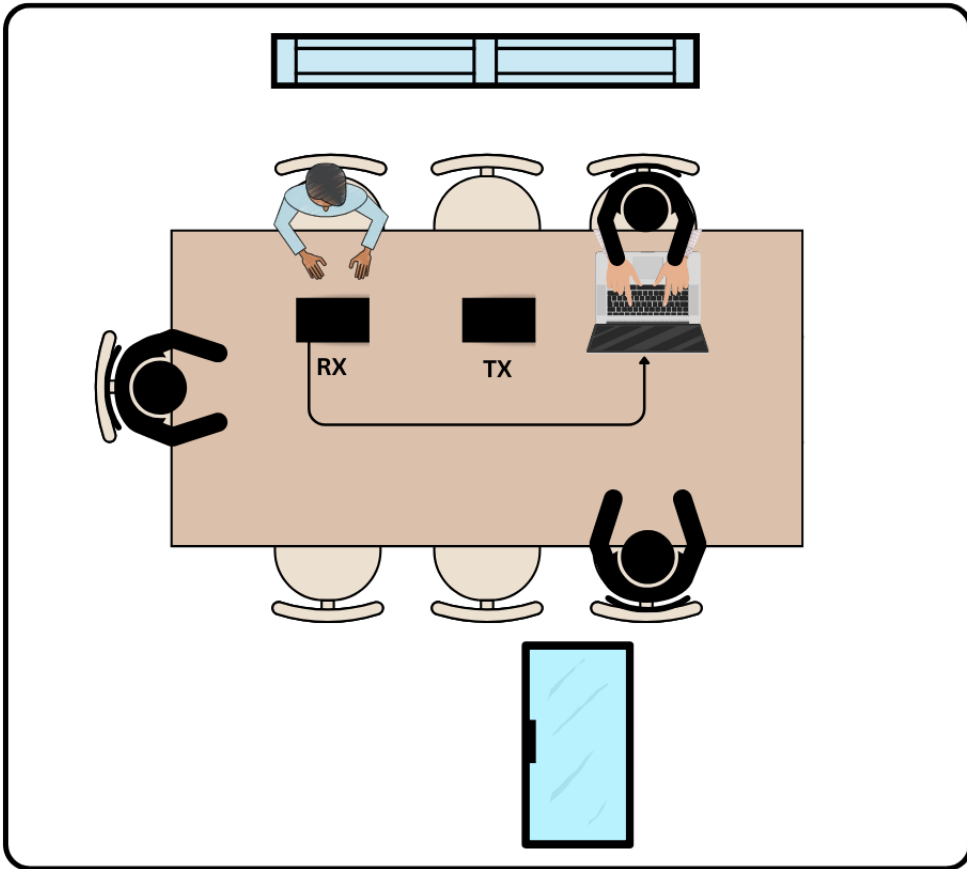


## 5. PROPOSED METHODOLOGY



**Fig. 1.** The overall flow diagram of proposed facial expressions system.

## Experimental Setup



### Fig. 2. Experimental Environment



**Fig. 3.** Experimental Room

Parameter	Value
Wi-Fi Channel	6
Channel Bandwidth	20 MHz
CSI Sampling Rate	100 pps
Number Of Subcarriers	64
Center Frequency	2.4 GHz
Number Of Classes	5
Sampling Time Window	5sec
Sampling Frequency	100Hz
Transmitter Power	20 dBm

**Table 2.** Selected hardware and software parameter settings

# Methodologies Used for Dataset Collection with ESP 32 Built-in Antenna

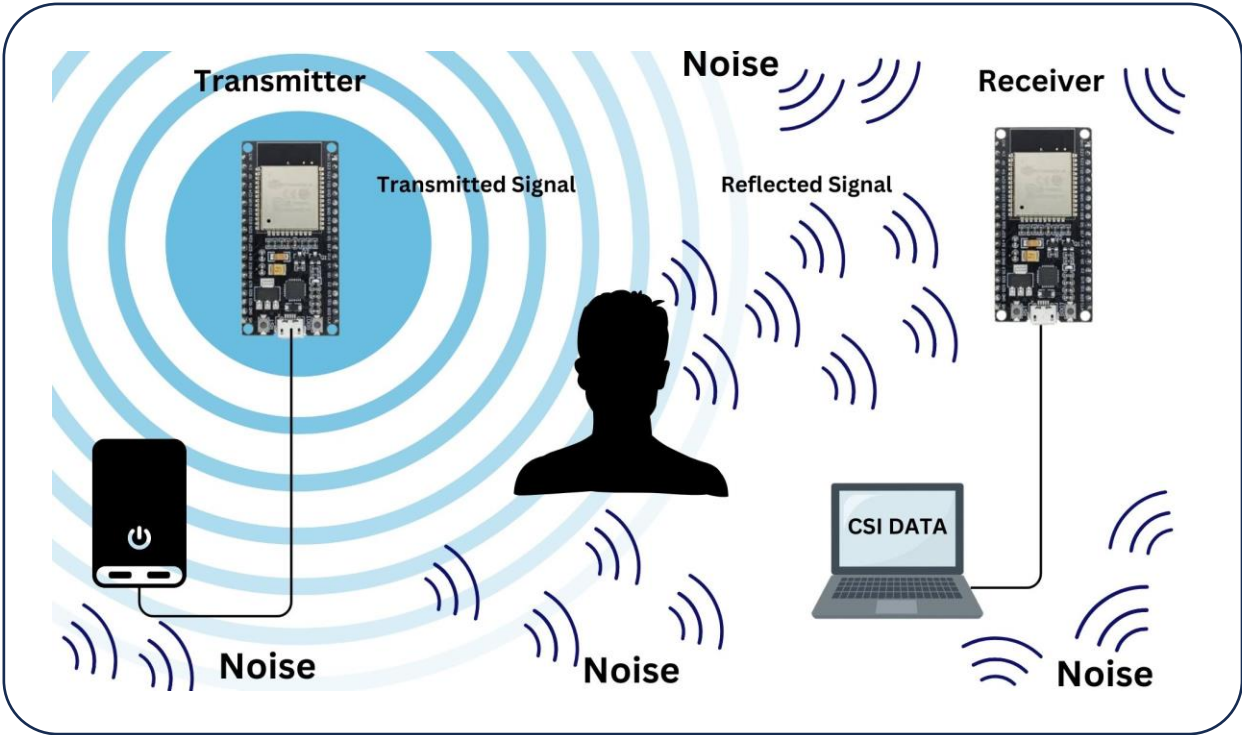


Fig 4. Antenna Without Reflector

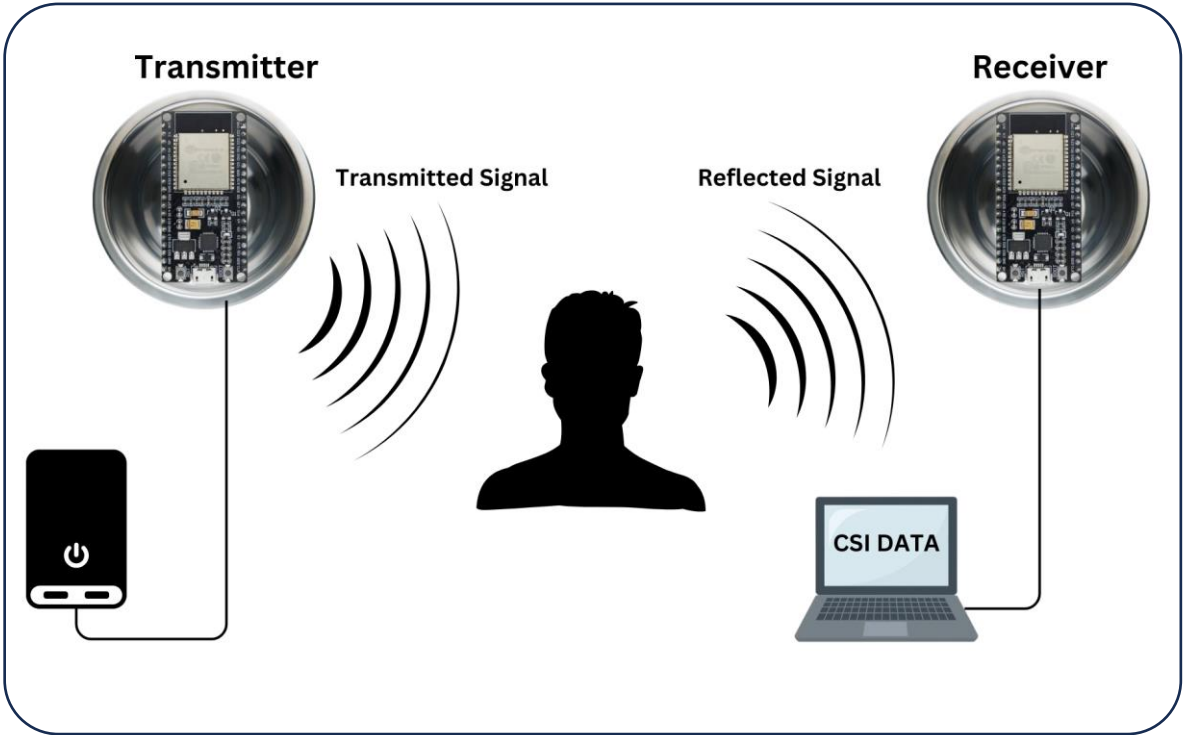


Fig 5. Antenna With Reflector

### Why we used signal Reflectors ?

#### Initial Setup (Built-in Antennas Only)

- Used ESP32 modules with default internal antennas.
- No external reflectors were attached.
- Human activity caused **minimal visible variation** in spectrograms.
- Weak multipath effect → **poor signal contrast**.

#### Issues Identified

- Low directional gain.
- Activity-induced changes were **not clearly captured**.
- Deep learning model performance was **limited**.

## Using Reflectors with ESP32 Modules

### What Reflectors Offer

- Focus Wi-Fi signals in a specific direction
- Enhance signal strength and multipath reflections
- Improve sensitivity to human movement

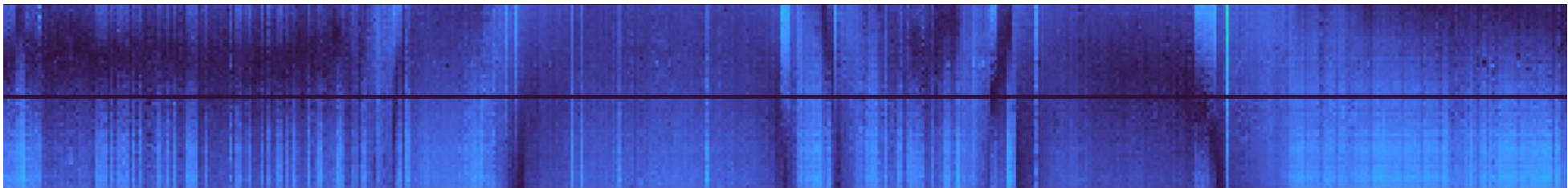
### Impact On CSI

- Greater variation in amplitude and phase
- More pronounced response to activity-induced signal changes
- Cleaner, more informative CSI data

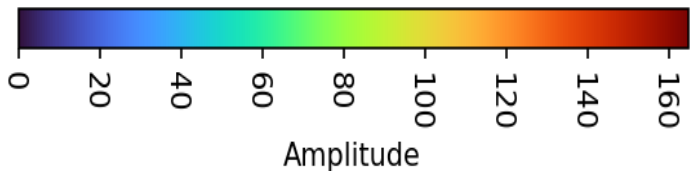
# Effect of Reflectors on Spectrograms

## Result on Spectrograms

- Clearer, sharper activity patterns
- Better separation between different activities
- Improved input quality for training deep learning models

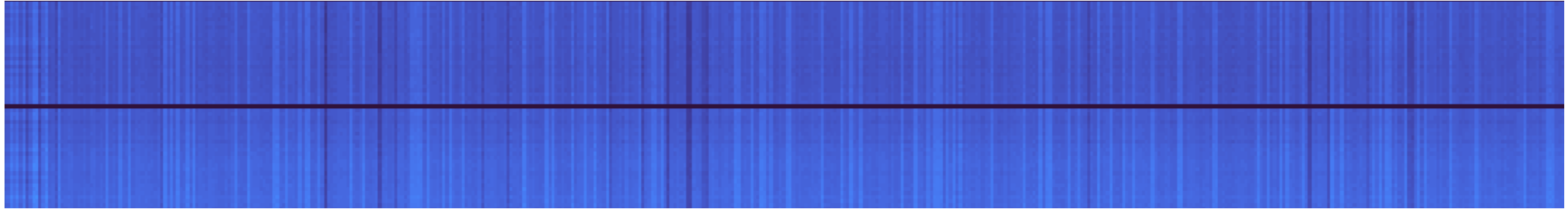


**Fig 6.** Sample spectrogram of Activity Surprised

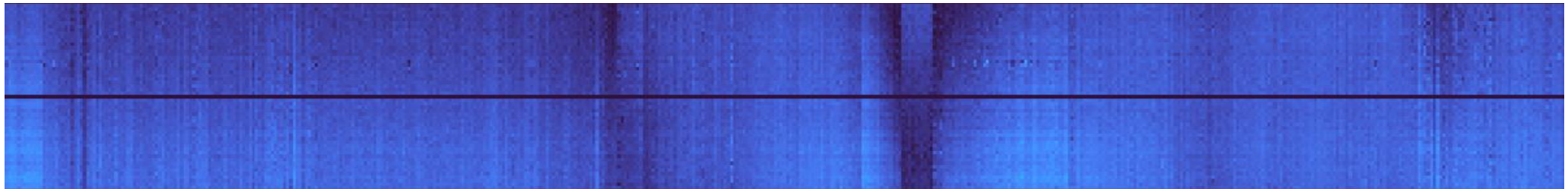


- Light blue part of spectrogram represent high amplitude
- Dark blue lines represent low amplitude

## Comparison Analysis (Happy)



**Fig 7.** Sample spectrogram (Antennas without Reflectors)



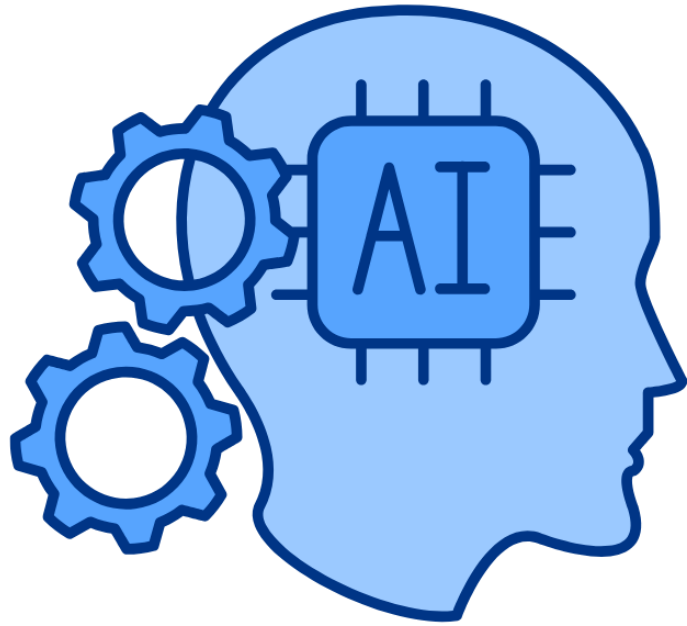
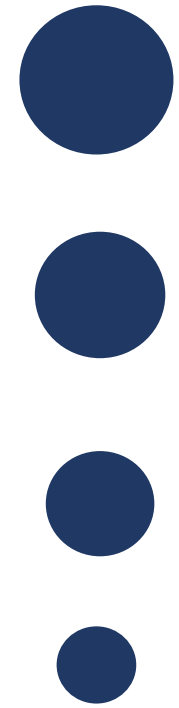
**Fig 8.** Sample spectrogram (Antennas with Reflectors)

# Collected Dataset Summary for two approaches

Classes	Samples Per Person	Total Samples (Without Reflectors)	Total Sample (With Reflectors)
Neutral	2	60	188
Happy	2	60	188
Sad	2	60	188
Fear	2	60	188
Surprise	2	60	188
Total	10	600	940

Table 3. Collected Dataset Summary

## 5. MODEL TRAINING



## 6. MODEL TRAINING

# Emotion Classification from Wi-Fi CSI Spectrograms



**Generalization**  
Model's ability to predict unseen data



**Model**  
Learns mapping function



**Label**  
Emotion category



**Input**  
Spectrogram image

# Using CNNs for Emotion Recognition with Random CSV Data Splitting

1. **Image Segmentation:** CNNs break images into parts for analysis.
2. **Pattern Recognition:** CNNs identify patterns like edges and shapes.
3. **Emotion Classification:** CNNs classify emotions based on patterns.
4. **Spectrogram Analysis:** CNNs analyze spectrograms for emotion patterns.



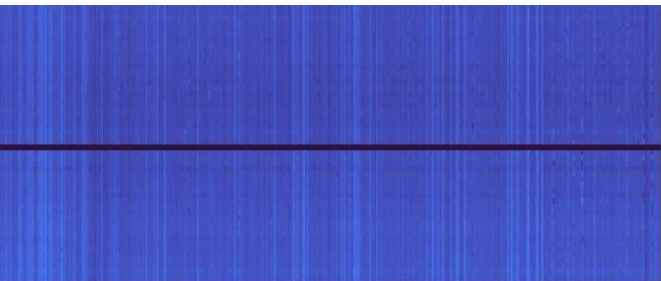
**Training: 80%**

**Validation: 10%**

**Testing: 10%**

6. MODEL TRAINING

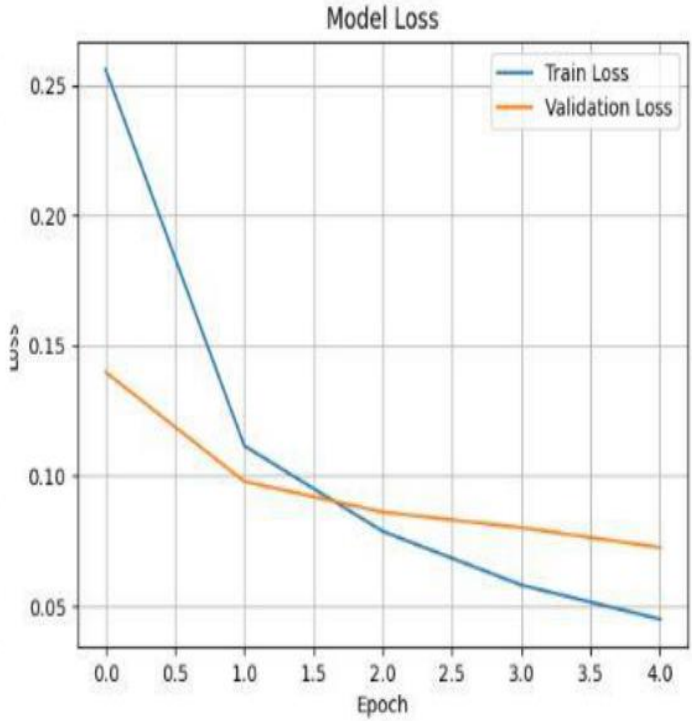
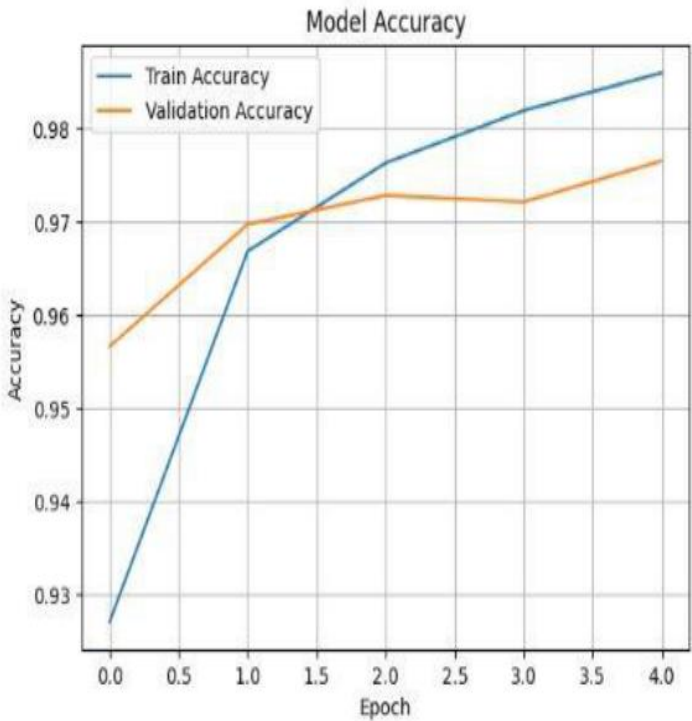
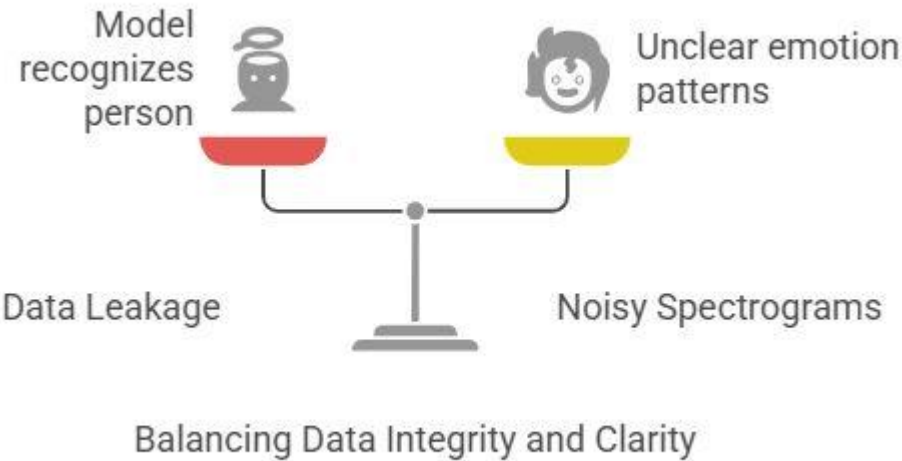
Experiment 01  
CNN Model used: (ResNet-18)  
Spectrogram Images in RGB



Why Did We Choose ResNet18?

It's a simple and fast deep learning model for image classification. **Problem( Test 49%)**  
It uses "skip connections" which help it learn better, even if the model is deep.  
A great starting point to test how well a model can learn emotional patterns.

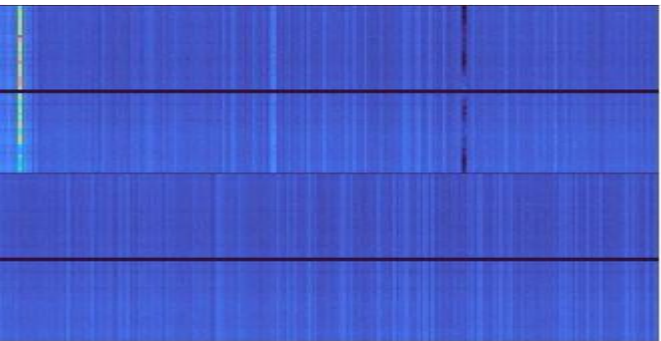
Performance



6. MODEL TRAINING

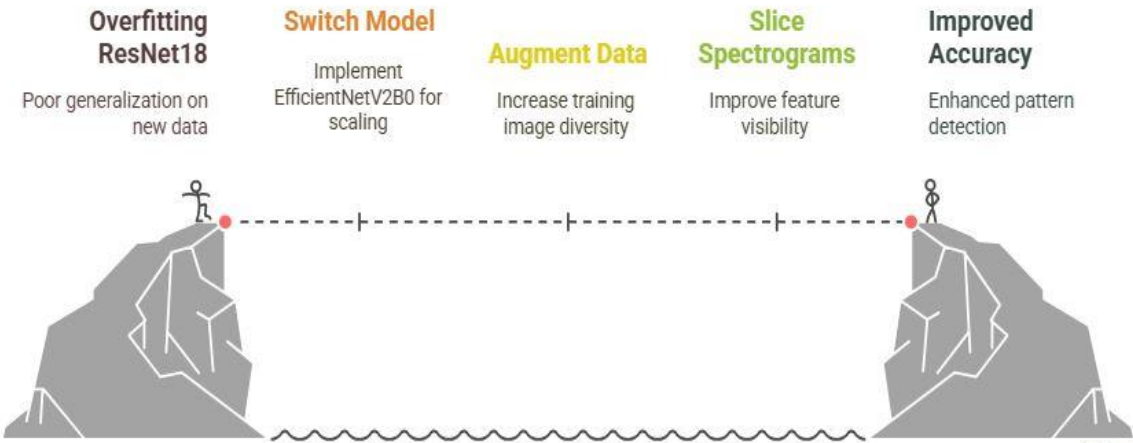
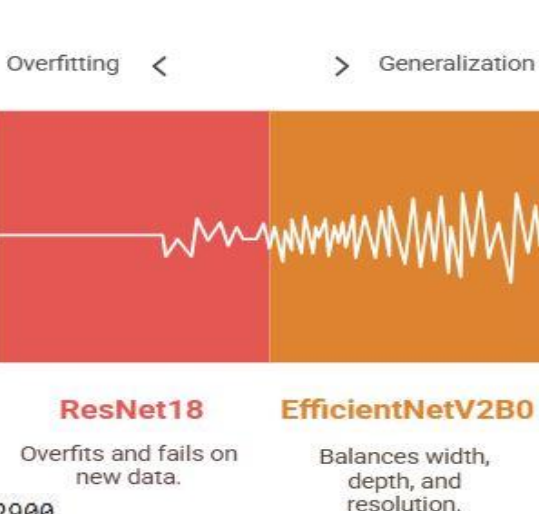
Experiment 02

CNN Model used: EfficientNetV2B0  
RGB Spectrograms (Slicing & Augmentation Applied)



Model accuracy depends on generalization and data handling.

Enhancing Model Performance



Test Accuracy: 0.3400

Test Accuracy: 0.2900

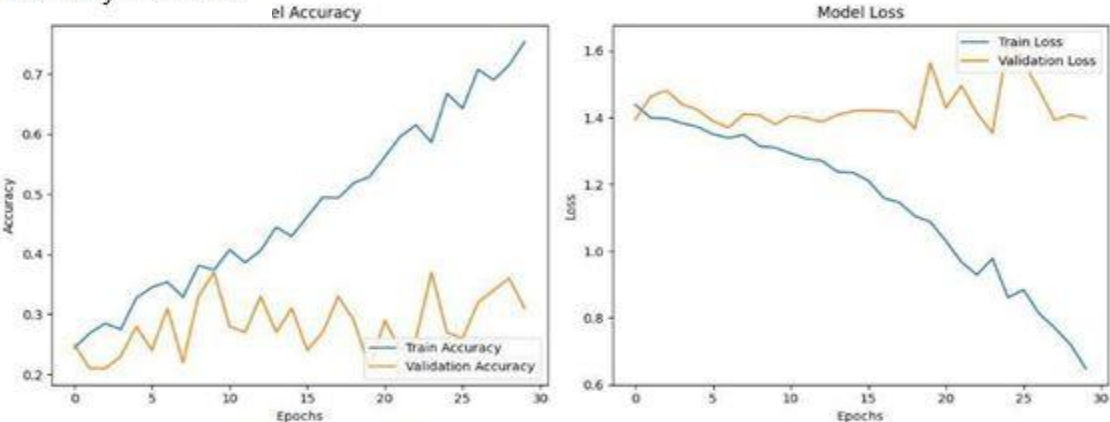


Figure 13: Slicing without augmentation result

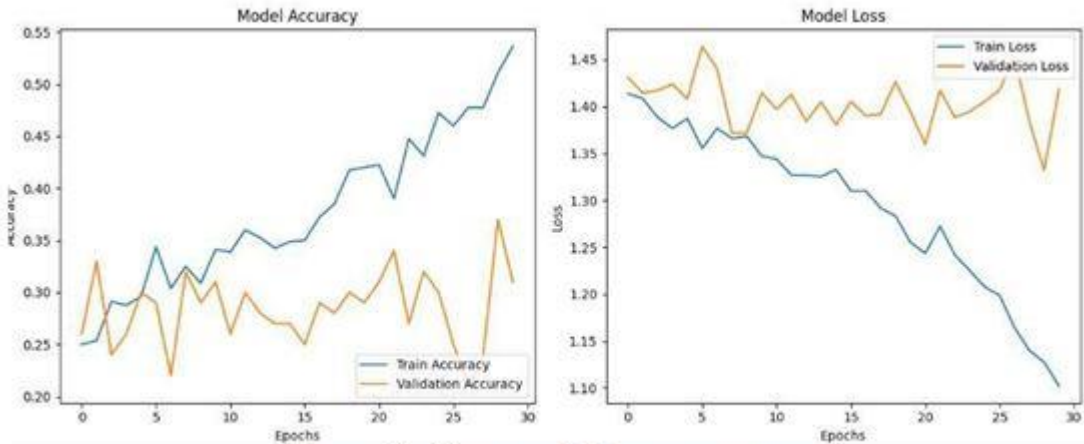


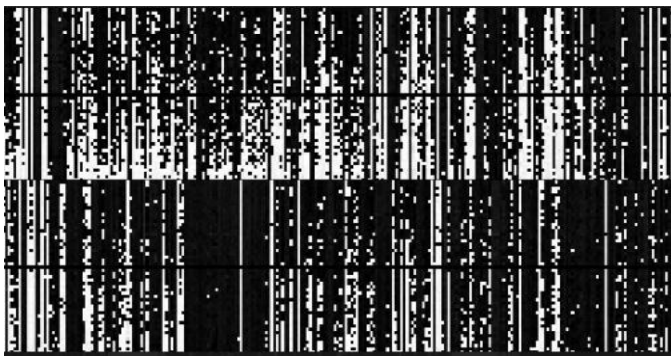
Figure 14: Slicing with Augmentation result

# Experiment 03

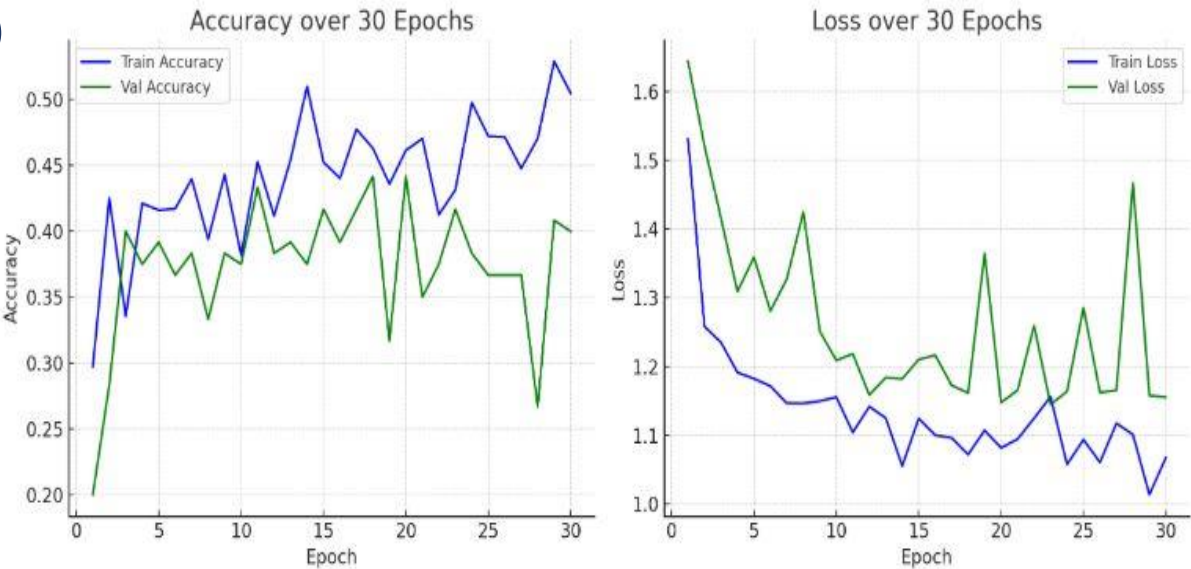
CNN: EfficientNetV2B0  
Absolute Difference with Slicing

Despite better results with EfficientNetV2B0 + Slicing, a large gap remained between training and test accuracy. This was likely due to noisy spectrograms containing irrelevant or overlapping signals, making it hard for the model to learn true emotional features.

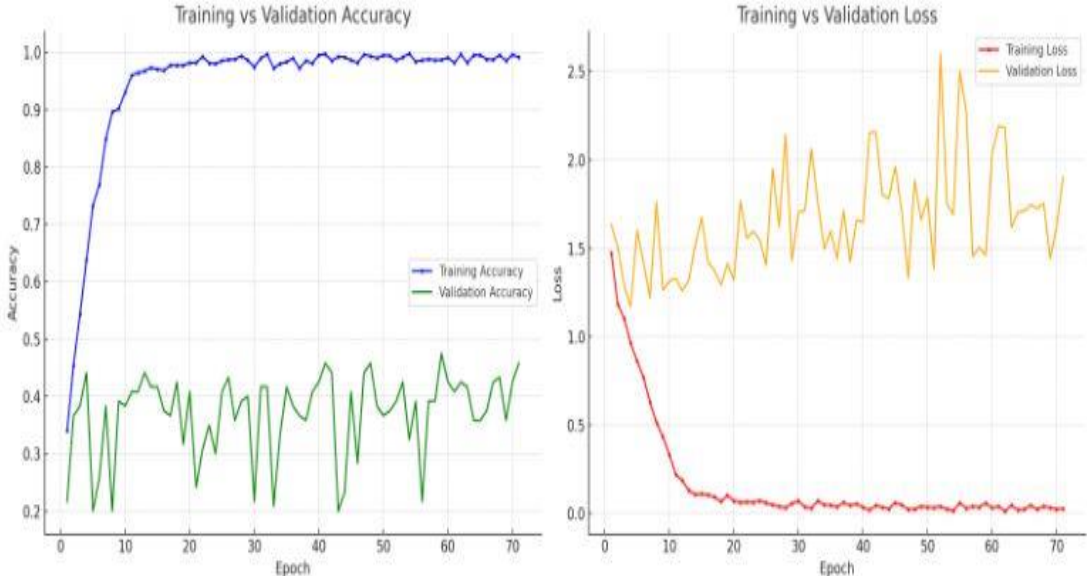
- Absolute Difference Impact:**
  - Better separation of emotional signals
  - Improved validation and test accuracy
  - Model learned meaningful patterns more easily



**Improved Data Splitting**  
Previously: Random split led to possible data leakage  
Now: All 5 emotions from a person are kept in the same split (train, val, or test), ensuring fair evaluation on unseen individuals



With Augmentation:  
Test accuracy- 0.4400



Without Augmentation:  
Test accuracy- 0.424

6. MODEL TRAINING

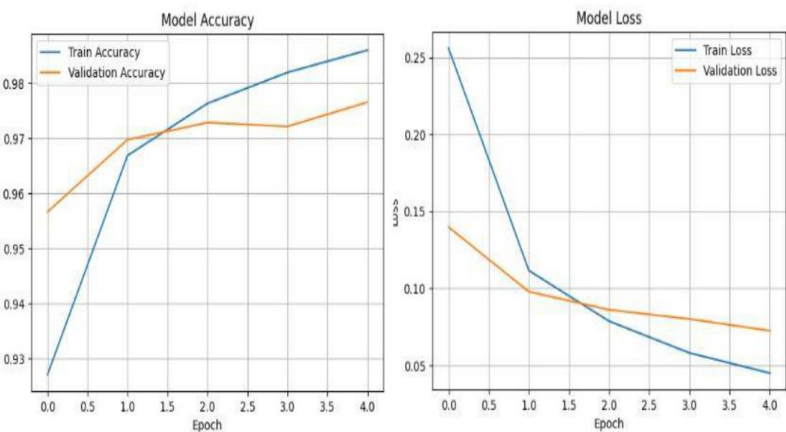
Experiment 01

Model: ResNet-18  
Data Prep: Raw RGB  
Spectrograms

Test Accuracy: 49%

Primary Insight:

- Baseline established.
- Identified noisy data & unclear patterns.
- Highlighted potential data leakage.



Comparative Analysis

Experiment 02

Model: EfficientNetV280  
Data Prep: RGB  
Spectrograms with Slicing  
& Augmentation

Test Accuracy: ~35%

Primary Insight:

- Overfitting matter resolved .
- Slicing improved feature focus.
- Augmentation enhanced generalization.

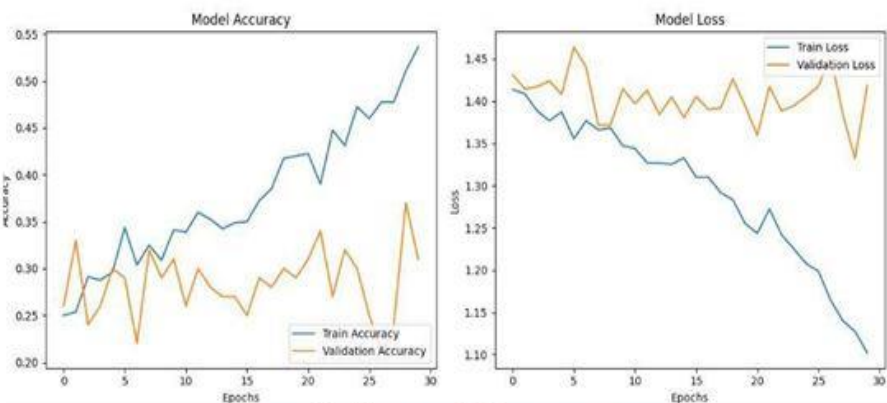


Figure 14: Slicing with Augmentation result

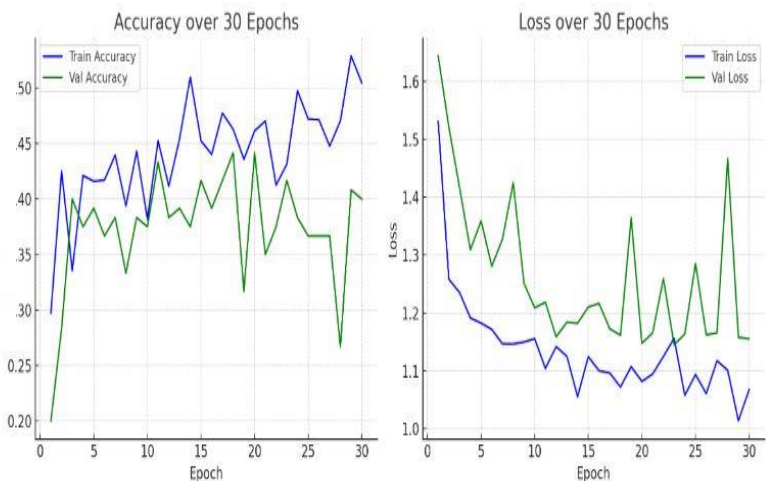
Experiment 03

Model: EfficientNetV2B0  
Data Prep: Absolute  
Difference with Slicing &  
Improved Data Splitting  
(Person-wise)

Test Accuracy: ~47%

Primary Insight:

- Focus on True Generalization.
- Eliminated data leakage via person-wise split.
- More reliable, albeit lower, test accuracy.



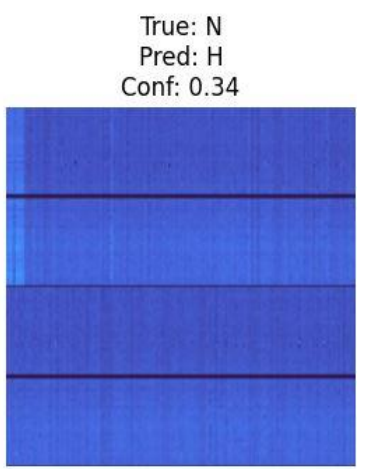
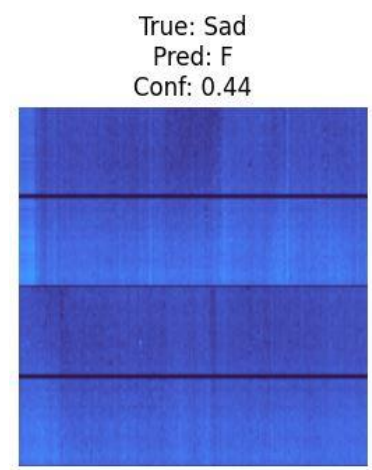
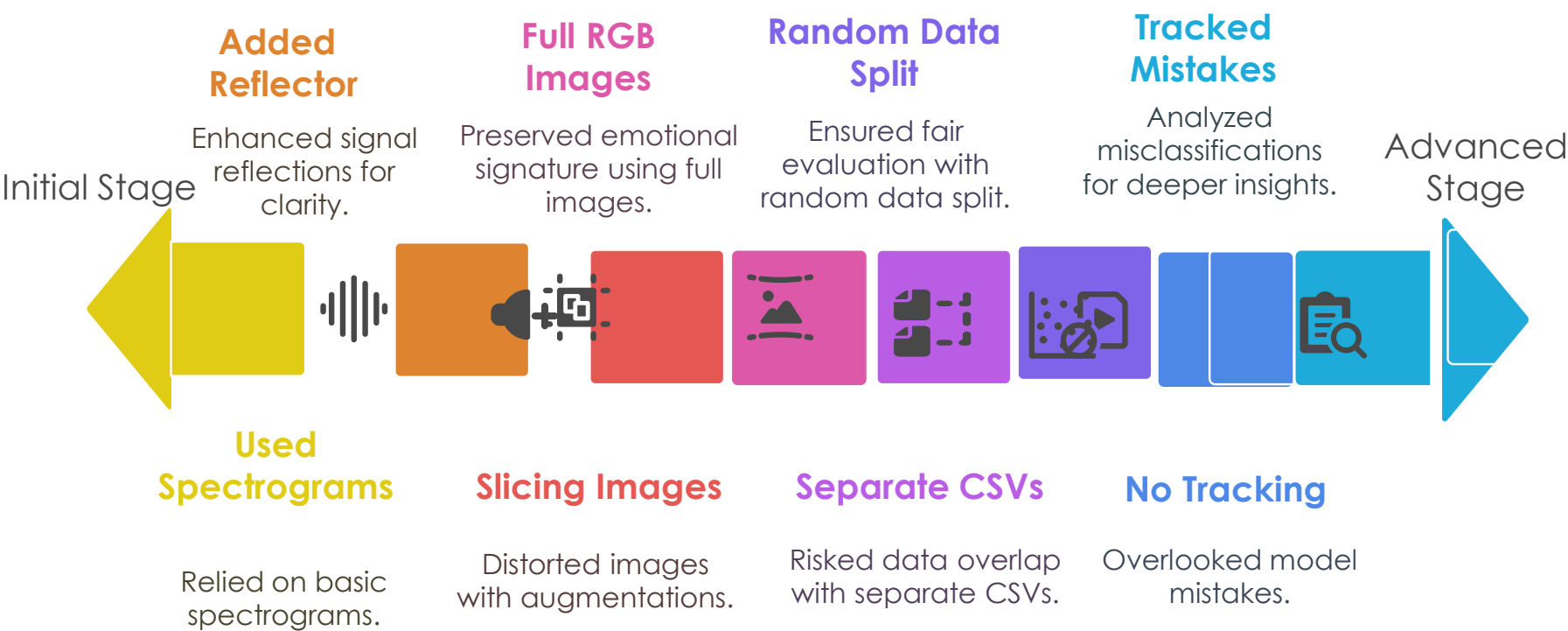
Overall Evolution: From basic model to advanced architectures, with iterative refinements in data preprocessing and crucial methodological rigor for true generalization.

Next Step: Enhance input data quality with our **\*\*Spectrograms with Reflectors Approach\*\***.

6. MODEL TRAINING

# Improved Model Performance with Reflector Spectrograms

Model improvement by refining data processing and analysis methods.



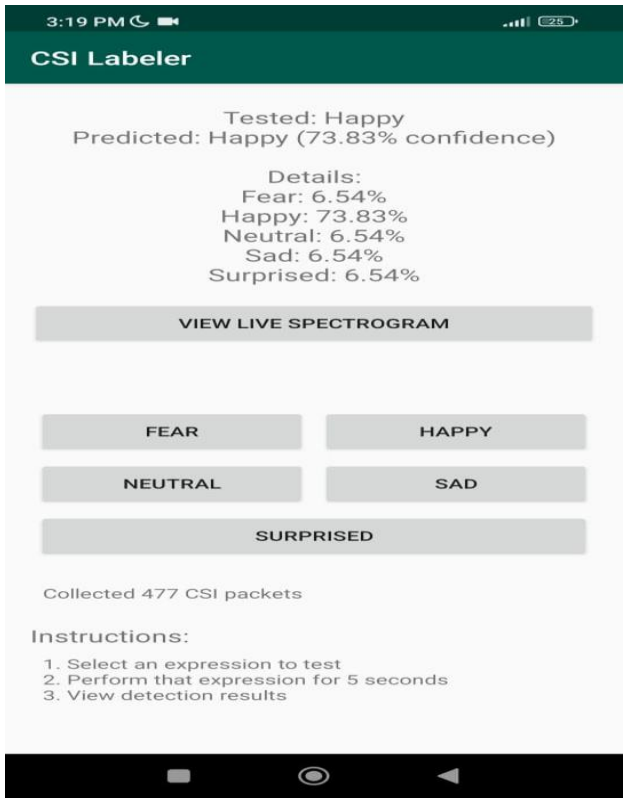
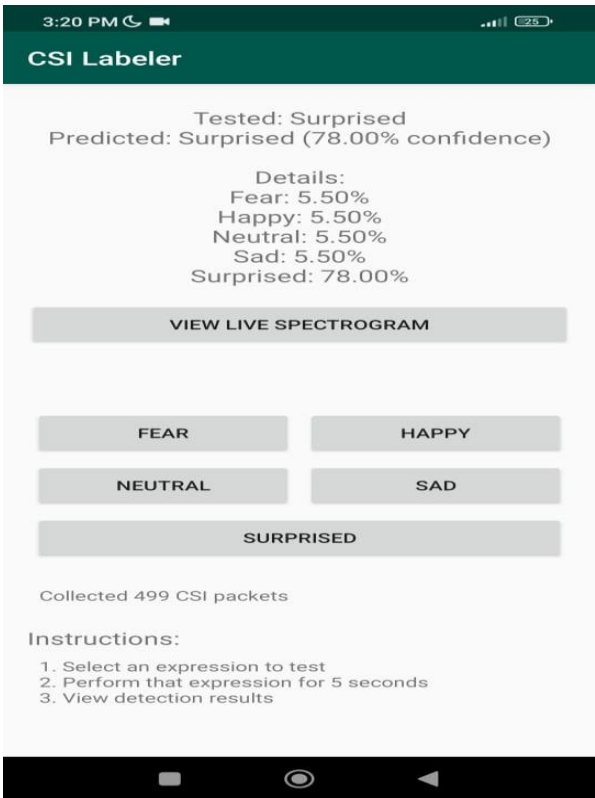
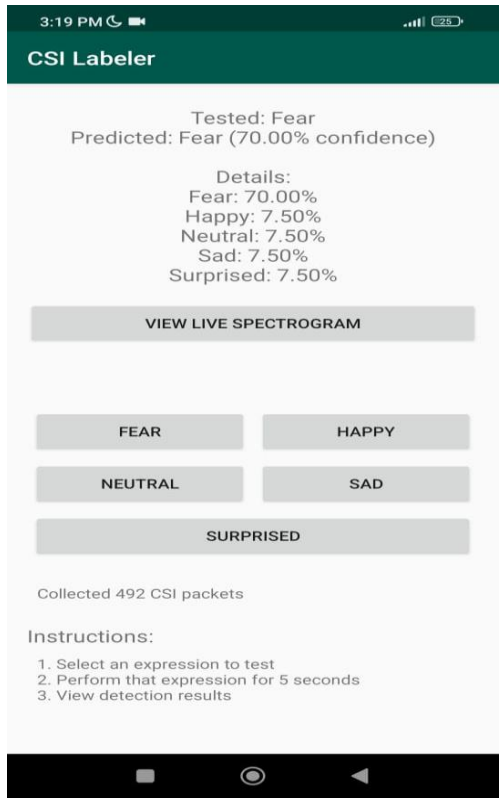
6. MODEL TRAINING



ESP32-PC Integration & Mobile App

Mobile App – Real-Time Emotion Sensing

- Connects to ESP32 via USB to stream CSI data
- Computes amplitude:  $\sqrt{(\text{real}^2 + \text{imag}^2)}$  for each subcarrier
- Generates and displays live spectrogram
- Runs TorchScript model for real-time emotion prediction
- Full pipeline: **ESP32** → **USB** → **Amplitude** → **Spectrogram** → **Emotion**





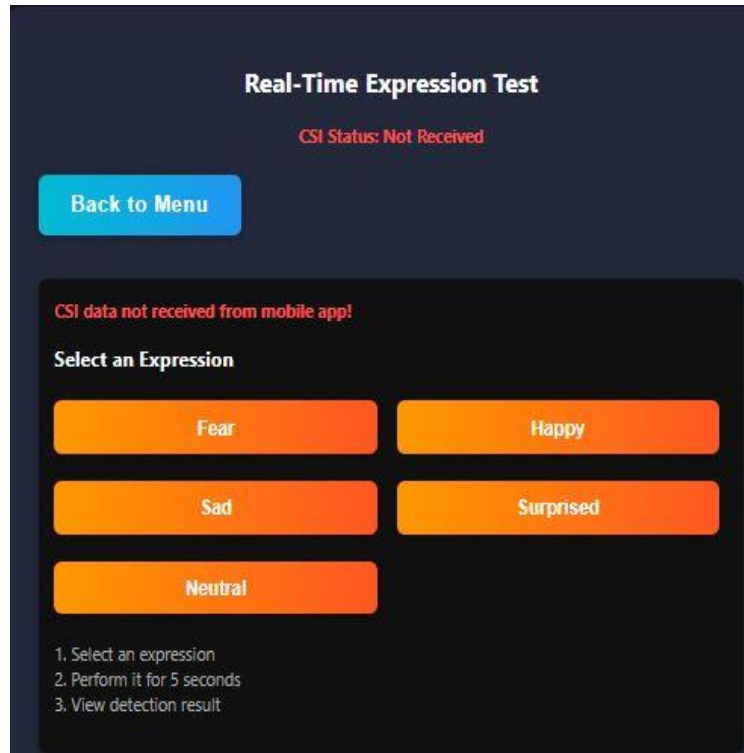
## 7. FUTURE WORK

## 7. FUTURE WORK

# Future Work

### Goal:

Make the system accessible from anywhere — so users can view live spectrograms and emotion predictions in a web browser without using the mobile app.

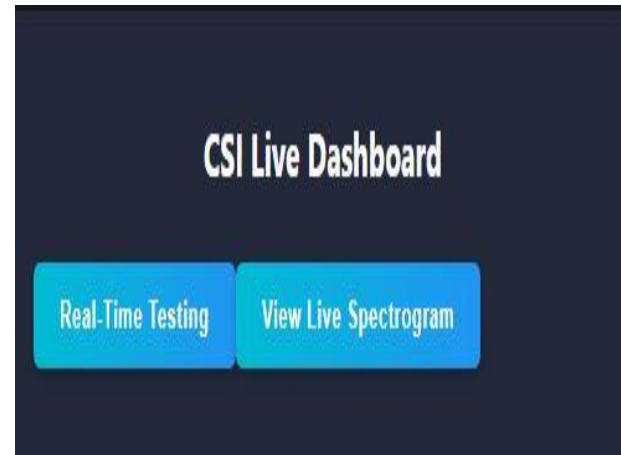


### Current Progress:

- Backend and dashboard are connected and working to some extent.
- Emotion prediction works in the browser using ONNX.js.
- Live data forwarding from mobile app to dashboard still has delays or drops, due to time/resource constraints.

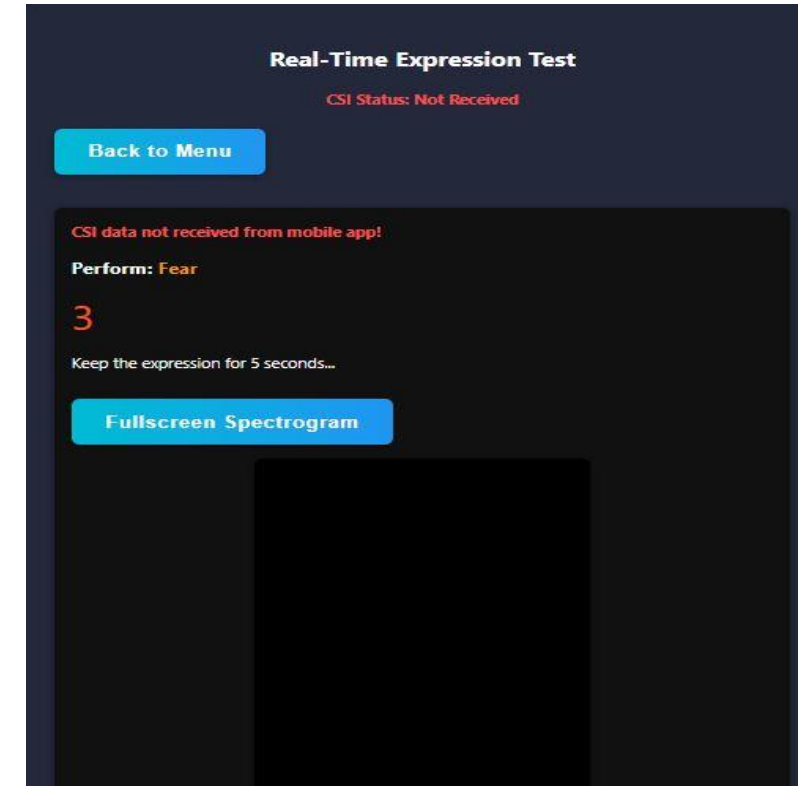
### System Overview:

- The **mobile app** collects CSI data from the ESP32 and sends it to a **backend server** using WiFi (HTTP POST).
- It also provides an **API** (a way for other apps to get data) for the web dashboard to fetch the latest CSI signals.
- The **backend server** stores or buffers the incoming CSI amplitude data.



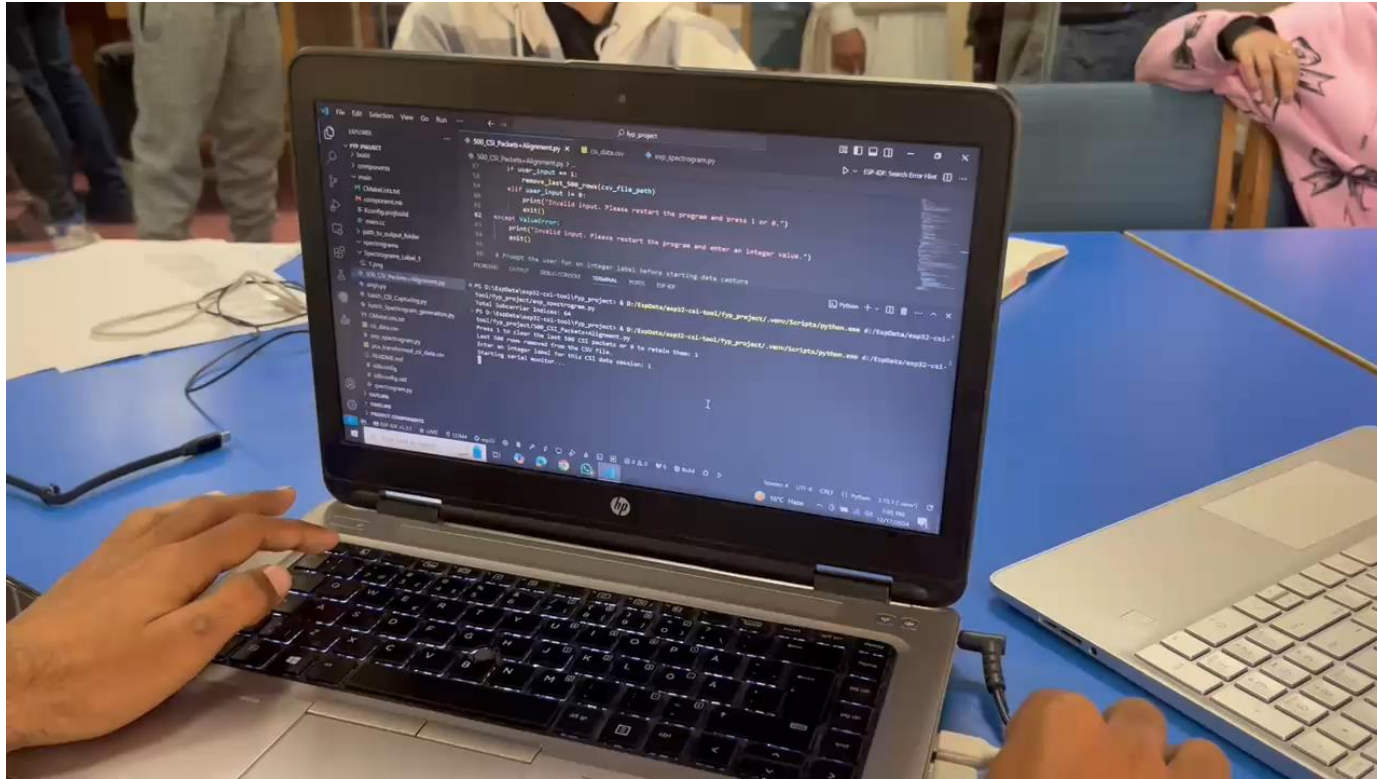
### React Dashboard (Web Page):

- Built with **React.js**, the dashboard regularly asks the backend for new CSI data.
- It processes and visualizes the data as a **live spectrogram** in the browser.
- The dashboard loads the trained **ONNX model** to predict the user's emotion in real-time — right in the browser, with no server-side inference needed.



### Next Steps:

- Fix the data streaming between mobile and web for smooth, real-time updates.
- Add features like emotion display, session recording, and remote monitoring.
- Ensure that the **web dashboard matches the mobile app's capabilities** for full remote access.



## 8. Experiment Demonstration

**THANK YOU  
FOR YOUR  
ATTENTION !!**

