AMRITA VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY UNDER SECTION 3 OF UGC ACT, 1956

ENVIRONMENTAL QUALITY MONITORING

DONE BY -
MANASWINI - CB.AI.U4AIM24129
RAGHU HARSHA VARDHAN - CB.AI.U4AIM24135
VEDANG PRATAP SINGH - CB.AI.U4AIM24151
AYYAPPA HARSHA VARDHAN - CB.AI.U4AIM24155

INTRODUCTION

With the fast-changing world of today, tracking environmental parameters such as air conditions, temperature, and humidity is most important for wellness and sustainability.

Our goal is to develop working on building a smart Environmental Monitoring System (EMS) that unites smart sensors and AI-driven insights to discover patterns, predict changes in the environment using FNN, and offer valuable decision support.

OBJECTIVE

- Create an Environmental Monitoring System (EMS) to monitor important environmental parameters.
- Implement advanced sensors to monitor air quality, temperature, humidity, and other parameters.
- Apply AI and data analysis to identify patterns and forecast environmental changes.
- Offer real-time monitoring and alerts for proactive decision-making.
- Facilitate sustainability and public health through precise environmental insights.

PROBLEM STATEMENT

Traditional environmental monitoring is slow, manual, and lacks real-time data.
No early warning system for detecting harmful environmental changes.
Inefficient decision-making due to delayed or incomplete data.

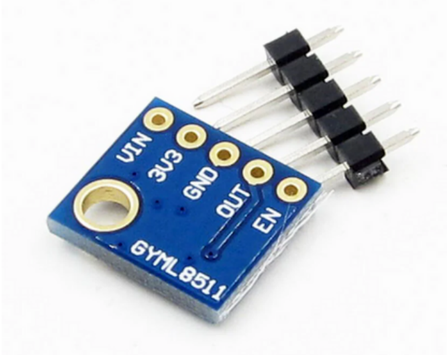
SOLUTION

Develop an AI-powered EMS with real-time sensors and an FNN-based prediction model which ensures early detection of environmental changes, enabling quick alerts and efficient decision-making.

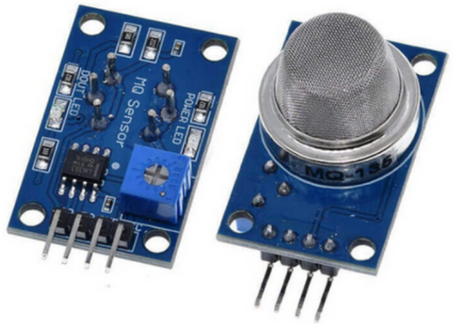
METHODOLOGY

1. Sensor Integration – Deploy environmental sensors to collect real-time data (e.g., temperature, humidity, air quality).
2. Data Collection & Preprocessing – Continuously gather sensor data, clean and normalize it for AI processing.
3. Prediction Module – Train the FNN to predict future environmental values based on historical sensor data.
4. Monitoring & Alerts – Display current and predicted data on a dashboard and send alerts for abnormal conditions.
5. Testing & Optimization – Evaluate model accuracy and optimize system performance using test datasets.
6. Deployment – Implement the system in real-world locations for continuous monitoring and prediction.

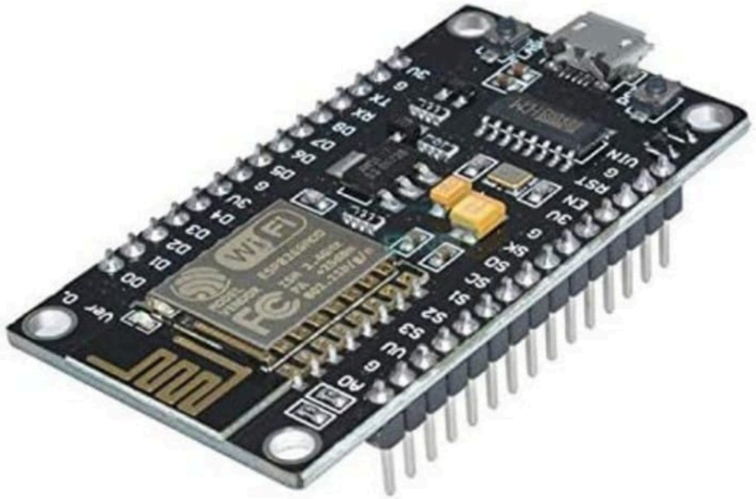
MATERIALS REQUIRED



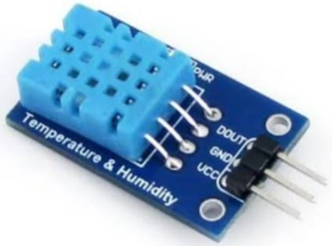
ML8511



MQ-135



ESP32



DTH11

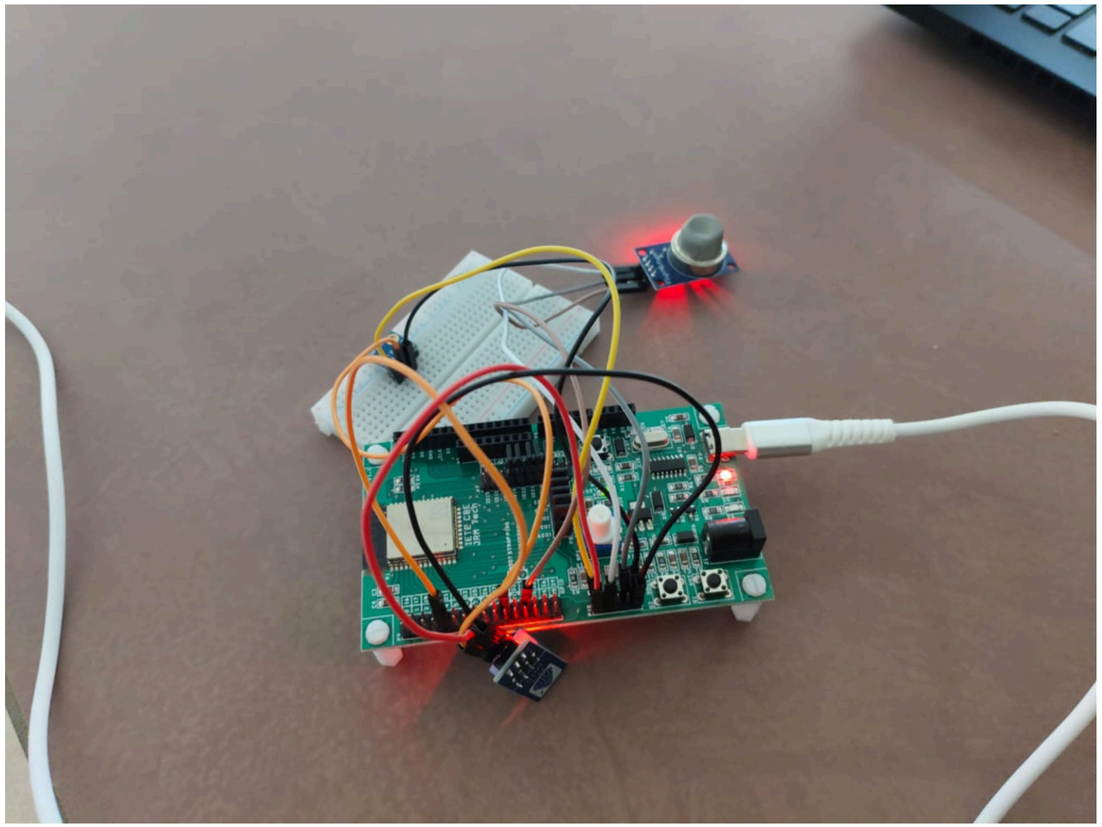
FOR FIRST REVIEW

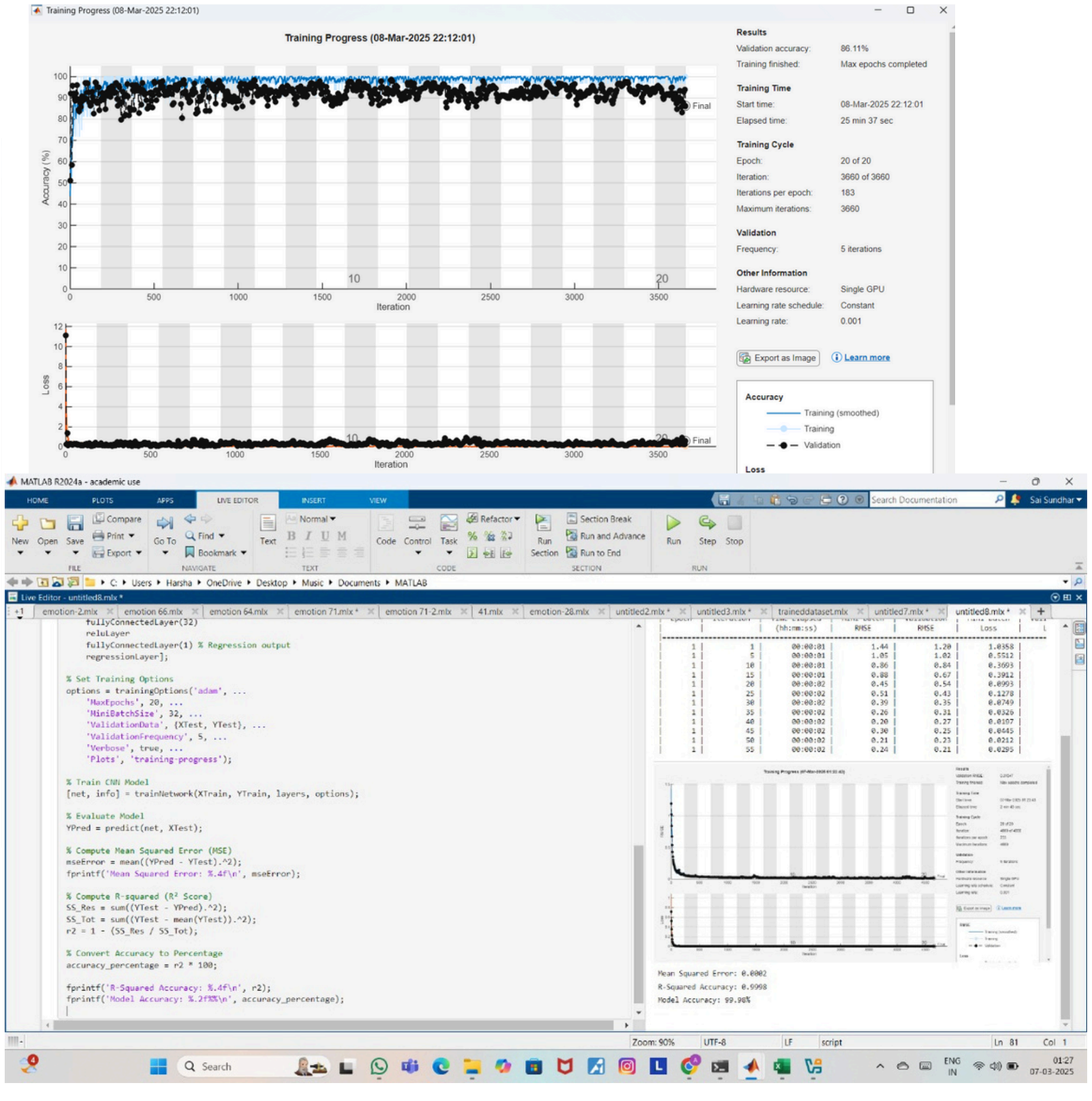
- using Arduino done a sample model
- Found the datasets related to our project

FOR SECOND REVIEW

- Used Esp32 instead of Arduino
- Used a FNN model for training and testing the dataset
- Converted the sensor values to proper units

RESULT




Training Progress (08-Mar-2025 22:12:01)
Training Progress (08-Mar-2025 22:12:01)
Accuracy (%)
Iteration
Loss
Final
Results
Validation accuracy: 86.11%
Training finished: Max epochs completed
Training Time
Start time: 08-Mar-2025 22:12:01
Elapsed time: 25 min 37 sec
Training Cycle
Epoch: 20 of 20
Iteration: 3660 of 3660
Iterations per epoch: 183
Maximum iterations: 3660
Validation
Frequency: 5 iterations
Other Information
Hardware resource: Single GPU
Learning rate schedule: Constant
Learning rate: 0.001
Export as Image
Learn more
Accuracy
Training (smoothed)
Training
Validation
Loss
MATLAB R2024a - academic use
Live Editor - untitled8.mlx *
% Set Training Options
% Train CNN Model
% Evaluate Model
% Compute Mean Squared Error (MSE)
% Compute R-squared (R² Score)
% Convert Accuracy to Percentage
Mean Squared Error: 0.0002
R-Squared Accuracy: 0.9998
Model Accuracy: 99.98%

Thank You