MoveMe : A Smart Public Transport Managment System with Blind Spot Warning and Solar Powered Bus Stop

1st Zobaer Ibn Razzaque
Computer Science and Engineering
United International University
Dhaka, Bangladesh
zrazzaque221135@bscse.uiu.ac.bd

2nd Md. Sadman Haque
Computer Science and Engineering
United International University
Dhaka, Bangladesh
mhaque221592@bscse.uiu.ac.bd

3rd Robiul Awoul Robin
Computer Science and Engineering
United International University
Dhaka, Bangladesh
rrobin221564@bscse.uiu.ac.bd

4th Fabiha Nawal Aurna
Computer Science and Engineering
United International University
Dhaka, Bangladesh
faurna191159@bscse.uiu.ac.bd

5th Md. Nayemul Islam Rakib
Computer Science and Engineering
United International University
Dhaka, Bangladesh
mrakib203064@bscse.uiu.ac.bd

6th *Fahim Hafiz*
Computer Science and Engineering
United International University
Dhaka, Bangladesh
fahimhafiz@cse.uiu.ac.bd

Abstract - In this paper we will demonstrate a novel approach on solving the current existing issues in the Bangladesh Public Transportation system. Public transportation systems often suffer from inefficiencies and lack of real-time information, resulting in poor user experience and safety concerns for passengers. We want to create a safe driving scenario with blind spot warning system and a more organized process of daily commute. Move Me addresses these challenges by integrating multiple technologies to enhance public transportation. The solution includes a web app for real-time tracking of buses, solar-powered bus stops that display passenger counts, and a card punch system for tracking passengers. Data is securely stored on a server and presented through a user-friendly dashboard for easier access. The implemented system optimizes user experience and enhances safety, providing accurate tracking and efficient data management for passengers and transit operators.

Index Terms - Object Detection, Solar Power, Bus Stop, Data Server, Data Visualization, Public Transportation, Tracking

Github : https://github.com/sadman-adib/Project-MoveMe

I. INTRODUCTION

MoveMe is designed to create a more organized and efficient public transportation system, where there will be real time tracking of the passengers count and the location of the bus, all synched within an end to end system, connecting with the smartphones of the passengers, a data server and the bus itself.

In Bangladesh, the bus drivers are more than reluctant to follow the traffic rules, let alone the requirement of parking the bus in a manner so that it doesn't cause traffic congestion. Moreover, their careless and reckless driving causes many road accidents. The passengers on the other hand have no idea when the next bus is coming, knowing how far the bus is, is an impossible task. Our project addresses these issues perfectly.

Prior to our approach, there have been works done on detecting objects near vehicle blind spots using camera [1], implementing YOLO to detect objects [2] [3], detecting objects which are in probability of collision in the blind spots of vehicles [4], harnessing solar energy to power IOT devices [5], data transfer using HTTP server system [6], IOT data transfer system [7], smart transportation system [8]

After carefully analyzing the above-mentioned papers, we implemented the SONAR system embedded in the vehicle's body, to further enhance the proximity of an object, should the camera fail to detect it. We implemented a smart bus stop, which will show the current number of passengers on the upcoming bus and a web app, which the users can login into, and track the location of the bus (The app is in beta version, further polishing is required). We collected the passenger count data from the RFID card reading system, and kept track of it using a mysql database, which can be accessed and reports generated from an admin frontend.

II. PROPOSED METHOD

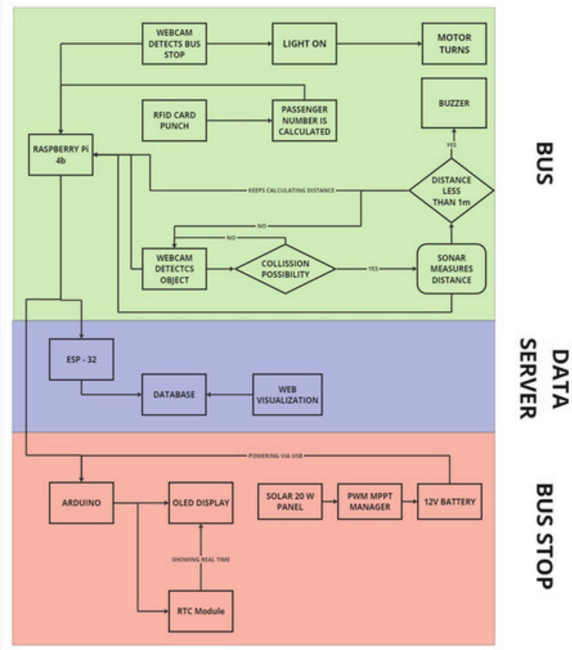


figure 1: Block Diagram of MoveMe

From this diagram we can see that the green highlighted part is implemented in the bus, controlled by the Raspberry Pi 4b, where it is connected to a webcam, a SONAR sensor and an RFID card reader for the sensor parts. For the actuator parts, there is a motor, an LED and a buzzer is installed. The camera detects upcoming objects and identifies them, the SONAR is constantly measuring the distance between various objects within proximity of the vehicle. If both the SONAR and the camera are aligned on an object being identified as there can be a collision, then the LED will turn on and the buzzer will sound to alert the driver. The camera also acts as a detector for the bus stops, where it can detect the bus stop, and allow the driver to potentially turn the motor and open the bus door. The RFID card system tracks the passenger count as the passenger passes the door and punches their RFID cards.This data is stored a copy in the raspberry pi itself, a copy to the ESP-32, and lastly to the arduino mega.

The ESP-32 is used primarily as a data server processor, it captures the data from the pi, pushes it into a mysql database, which is accessible from a webview frontend, with PHP and Javascript powered backend.

The Arduino itself is connected to the bus stop, being powered by solar energy. It also receives data from the pi, prompts the OLED display, it also utilizes an RTC module to keep track of the current time, and the last moment the data was refreshed. The arduino makes use of an ESP-8266 WiFi module.

III. IMPLEMENTED HARDWARE SYSTEM

A. List Of Hardware Components

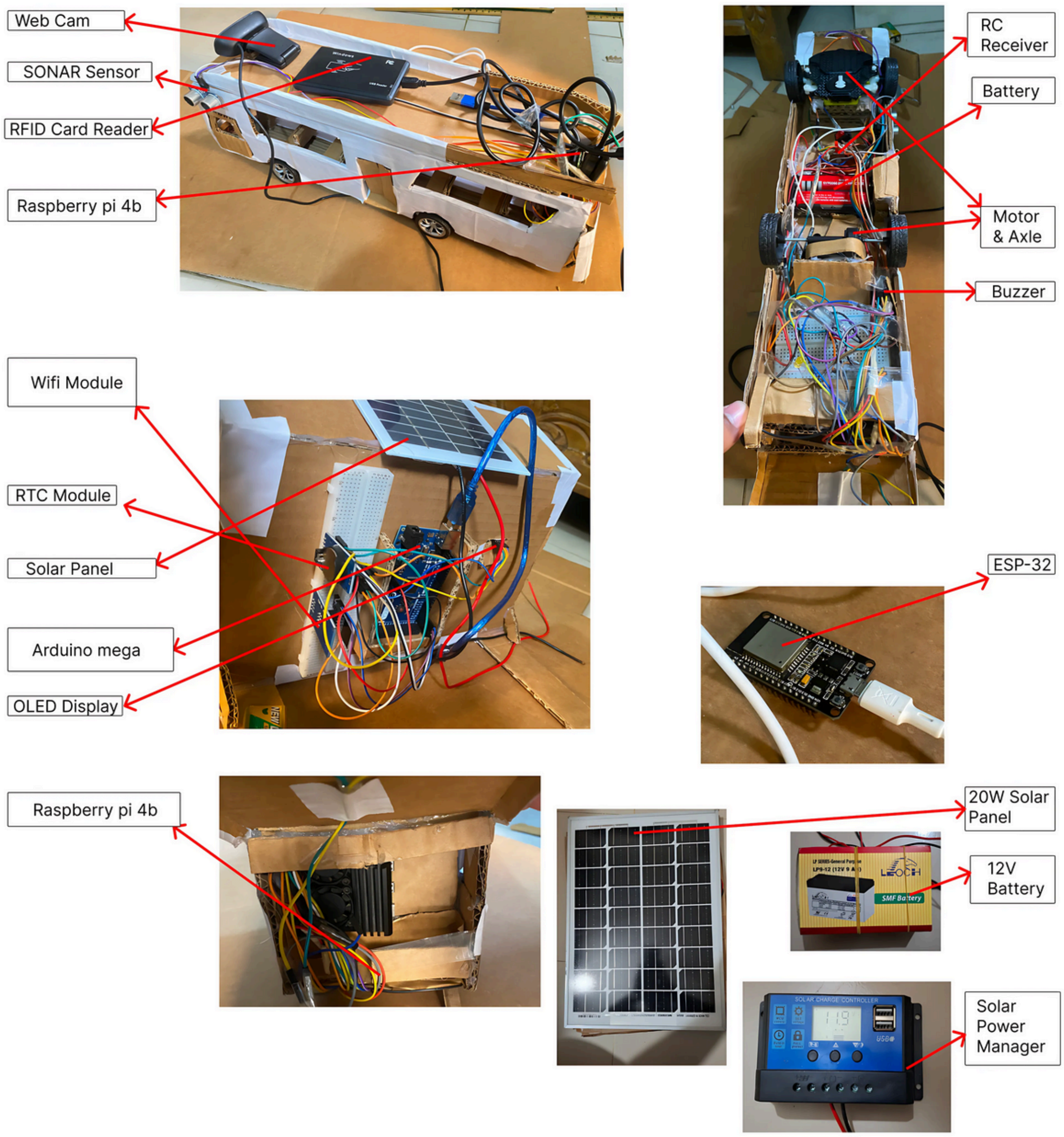
Table 1 : List of Hardware Components Used

Name	Specification	Cost-BDT	COST-USD
Arduino Mega	2560 CH340	1345	12.0
Buzzer	Active Buzzer 5V	20	1.0
ESP32	DevKitM-1	2190	19.0
LED Light	5 mm	10	1.0
OLED Display	0.96" Inch I2C	400	4.0
Push Button	Pull-up Resistor 1Kohm	5	1.0
RFID Card Reader	JT308 USB - 13.56 MegaHz	1300	11.0
Raspberry Pi 4	Version B, 4GB RAM	15000	125.0
RTC Module	DS3231	260	3.0
Solar Panel	20 Watt	1200	10.0
Ultrasonic Sensor	HC-SR04	100	1.0
Stepper Motor with Driver	12 Volt Motor, ULN2003 Driver Board	280	3.0
12 Volt battery	Lead-Acid Composite	650	6.0
Web-Cam	Logitech C270	2100	18.0
RC Car Accessories	2.4 GHz RC, Axle, Tires	1000	9.0
Arduino WiFi Module	ESP8266 & ESP01 Adapter	500	5.0
Solar Charge Controller	24V/12V 10A PWM, MPPT	550	5.0
Wire, Breadboard, Others	N/A	500	5.0
	TOTAL	27410	239.0

B. List Of Software Components

- Front End
  - HTML
    - To structure the frontend and give it a skeleton
  - CSS
    - Give some texture to the frontend, lift the look and feel
  - JavaScript
    - Add on page animation, generate graphs, add moving functionality
  - Next.JS
    - Web App Front End
- Back End
  - PHP
    - Robust code to handle all the admin panel features
  - JavaScript
    - Backend functions, chart generation and many more
  - Next.JS
    - Backend of the Web App
  - Google Maps API
    - Realtime map service provider
- Database
  - MySQL
    - Handling all the user data
  - Firebase
    - Realtime location update, quick service
- Deployment Tools
  - GitHub
    - Version controlling tool
  - Vercel
    - Free server to deploy the Web App

C. Hardware Implementation



D. Software Implementation

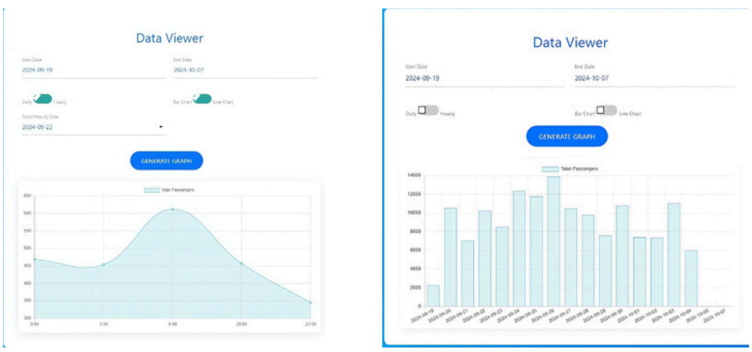


figure 2 : Admin Panel Data Visualisation

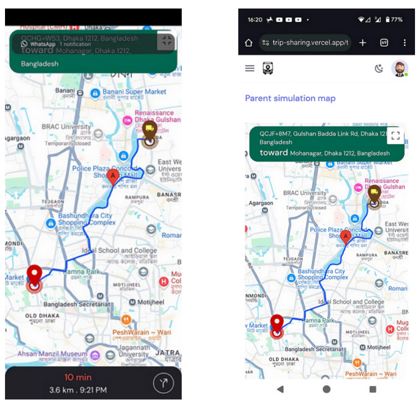


figure 3 : Web App for Bus Tracking

IV. RESULTS

- The Raspberry Pi 4b Successfully detects objects using the webcam, while utilizing the YOLOv4 tiny algorithm. Below it is described
  - How yolov4-tiny works :
    - YOLOv4-Tiny is a lightweight version of the YOLOv4 model, designed for real-time object detection with faster processing but slightly reduced accuracy compared to the full YOLOv4. It works by dividing an image into a grid, predicting bounding boxes, and classifying objects within those boxes in a single forward pass.
    - Here's a brief overview of how YOLOv4-Tiny works for object detection:
    - 1. Input Image: The input image is resized and fed into the neural network.
    - 2. Feature Extraction: YOLOv4-Tiny uses a simplified convolutional neural network (CNN) with fewer layers than YOLOv4, making it faster but less computationally intensive. It extracts important features (edges, textures, etc.) from the image.
    - 3. Grid Division: The image is divided into a grid, and for each grid cell, the model predicts bounding boxes (coordinates) and confidence scores for object presence.
    - 4. Object Classification: The model assigns a class label (e.g., person, vehicles, animals) to each bounding box based on the detected object.
    - 5. Non-Maximum Suppression: To reduce overlapping boxes for the same object, YOLOv4-Tiny applies non-maximum suppression, retaining only the box with the highest confidence.
  - The SONAR sensor can successfully detect the distance between object and the vehicle, pass it to the Pi consequently sounding the buzzer and turn on the LED
  - The object detection model successfully detects the bus stop and allows the driver to turn the door using stepper motor
  - RFID card is actively taking input from card punches, and extracting that information, passing it to the Pi, which in turn keeping a copy in a CSV and also passing them down to ESP-32 and Arduino Mega via WiFi in

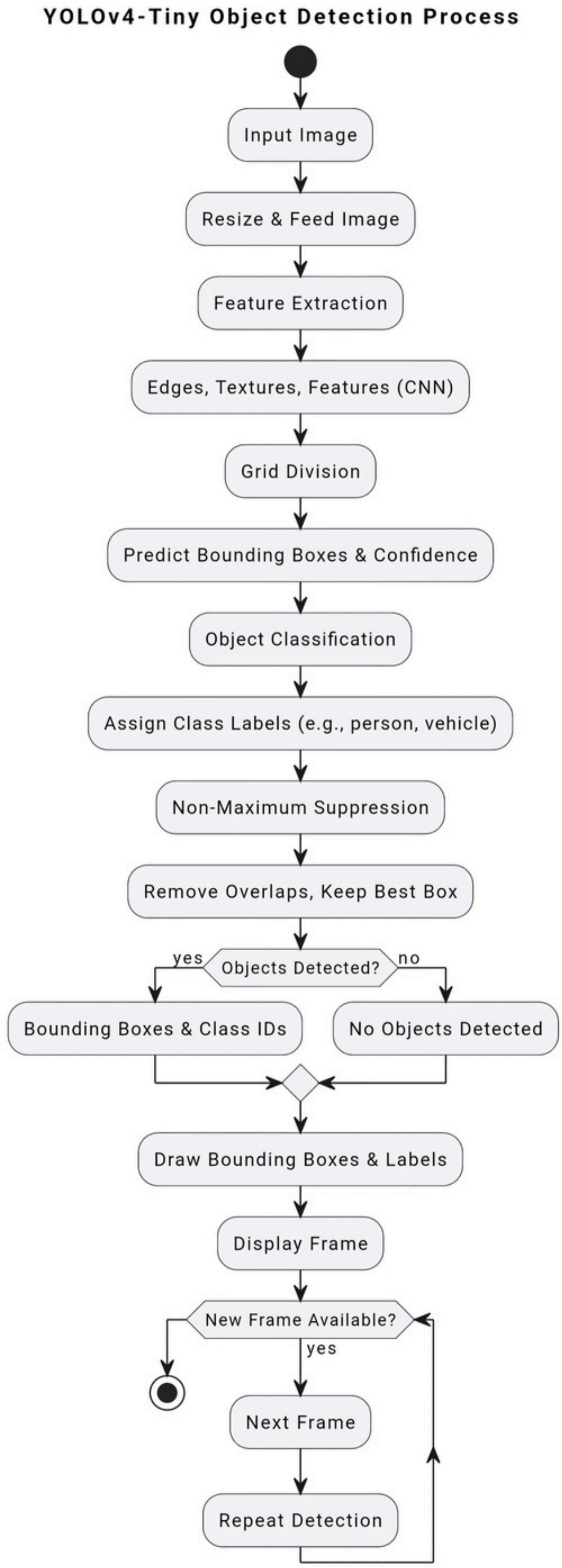


figure 4 : How YOLOv4 Tiny Works

- local LAN and persistent HTTP protocol
- The data is perfectly passed from the Pi to the ESP-32 and the Arduino Mega via WiFi and received successfully as well, all using the HTTP server methods
- The ESP-32 can properly push the data into the MySQL database
- Web admin panel can visualize the passenger data and create graphs
- Arduino Mega is working in harmony with the the RTC module, the ESP-8266 WiFi module and displaying the data correctly on the OLED
- Solar Panel is harnessing the energy from the sun using photovoltaic cells, the solar charge controller is doing it's part and charging the battery
- Arduino Mega is being powered by the battery from solar component
- Passengers can successfully login to the app and ensure a real time tracking is initiated
- The car RC module is working as expected and the car is driveable via the remote

V. Discussions

MoveMe is designed with a vision in mind, to bring dicipline in the public transport sector of Bangladesh, to modernize the day to day experience and to apply enhanced safety features to detect and avoid accidents. However, there are few limitations in this project.

Due to hardware limitations, the object detection model can't work process too many frames per second. Creating a delay in real life situations. When tested on a real car, we faced this issue. This can be overcome by implementing better hardware with more RAM. A local dataset will give us more advantage as well.

figure 5 : Implementation on Real Car

Currently the Tesla and VOLVO autonomous driving vehicles implement similar yet better and enhanced features like ours. Their object detection is faster, more accurate and their technology can eventually auto navigate the vehicle towards safety. However, they implement better hardware, better algorithm, use satelite feed data, which are expensive.

MoveMe, once implemented on a production scale, can eventually smooth out the daily experience of a public transport user in Bangladesh. This project is a small step towards a modern and smart city.

We have studied various documentation about the usage of the components, the ethical side of implementing this project is positive, as we are also enforcing clean energy.

The hardware and softwares are interconnected between each other and rely heavily on the input from each sensor to each actuator. The web app however is independently functional.

P1	P2	P3	P4	P5	P6	P7
✓	✓	✓	✓	✓		✓

TABLE II
COMPLEX ENGINEERING PROBLEM MAPPING. TIC P1 AND SOME OR MORE FROM P2-P7 AS APPLICABLE. P1 - DEPTH OF KNOWLEDGE, P2 - CONFLICTING REQUIREMENTS, P3 - DEPTH OF ANALYSIS, P4 - FAMILIARITY OF ISSUES, P5 - EXTENT OF APPLICABLE CODES, P6 - EXTENT OF STAKEHOLDERS, P7 - INTERDEPENDENCE

VI. CONCLUSION

MoveMe is a smart public transport managment system with blind spot warning and solar powered bus stop.

ACKNOWLEDGEMENT

We would like to extend our heartfelt gratitude to our faculty, Fahim Hafiz sir, also to the university, for allowing us to use the lab equipments, and everyone who were involved in any way.

REFERENCES

1. https://www.researchgate.net/publication/352600952_Blind_Spot_Warning_System_based_on_Vehicle_Analysis_in_Stream_Images_by_a_Real-Time_Self-Supervised_Deep_Learning_Model
2. https://www.irjmets.com/uploadedfiles/paper//issue_12_december_2022/32060/final/fin_irjmets1670926649.pdf
3. https://ijcsmc.com/docs/papers/July2021/V10I7202114.pdf
4. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9412342/
5. https://www.researchgate.net/publication/373613040_IoT-Enabled_Smart_Solar_Energy_Management_System_for_Enhancing_Smart_Grid_Power_Quality_and_Reliability
6. https://www.researchgate.net/publication/330281473_Internet_of_Things_Based_on_HTTP
7. https://www.researchgate.net/publication/324152744_Data_transfer_protocols_in_IoT-an_overview
8. https://www.researchgate.net/publication/354179565_Public_Transport_Systems_and_its_Impact_on_Sustainable_Smart_Cities_A_Systematic_Review