Road Safety Device for Accident Prevention

Srijon Mandal

It is a multi-faceted, cost-effective, portable and easily operable device that can be used in any type of vehicle and traffic poles, in order to generate relevant alerts to aid in safety for drivers and pedestrians. This device can also send critical events to emergency departments for appropriate actions and also provides rich data analytics for traffic assessment and planning.

It is aimed at United Nation’s Vision Zero initiative to reduce road fatalities across the globe.

Engineering

Introduction: Problem Statement

- Ensuring road safety is extremely critical as each year around 1.35 million lives are lost on roadways globally. Unfortunately ~38,000 people die every year in crashes on U.S. roadways. Additionally 4.4 million injured people requires medical attention. (Ref1)
- Though modern cars are equipped with collision avoidance systems, there is a strong need for cost effective and portable devices which can analyze and provide real-time alerts for a wide range of traffic incidents and road signs.
- Specially old automobiles, bikes and mobility equipments used by physically challenged people which can't integrate with advanced road safety devices urgently need an affordable low-cost solution.
- There is also a huge requirement for a portable traffic monitoring device which can be easily placed in City Poles, Community Gates and Traffic Lights.
- There is a increased need for automated real-time communication with emergency departments, traffic police and city departments in the event of hazardous situations on roadways.

Road safety management

© Vision Zero Network 2022

Introduction: Engineering Goal

Develop a portable cost-effective inter-connected smart road safety device

- Build the Single-Board Microcomputer to
  - Ingest video streams using camera, process, and run object detection algorithms to analyze the real-time streaming data
  - Display alerts via LCD Display
  - Announce events through a Speaker
  - Connect to a phone using Bluetooth
- Adopt the open source AI and ML libraries for automated identification. Train, test, and tune the deep learning models in a cloud-based system.
- Develop an Android phone app.
- Setup a database in cloud which can store the events taking place in a location detected by a device.
- Build a web application in cloud which can show the Dashboard.
- Develop the functionality to push messages to the phone.
- Implement the logic to send the alerts to database from the phone.
- Create charts and tables in Dashboard which can show the statistics in real-time.

Product Overview

Picture drawn by project team

Methods: Phase 1 - Designing the Portable Device

The device has a Raspberry Pi 4, a single board microcomputer, for object detection and identification based on computer vision and deep learning.

The device has following components as shown in the diagram.

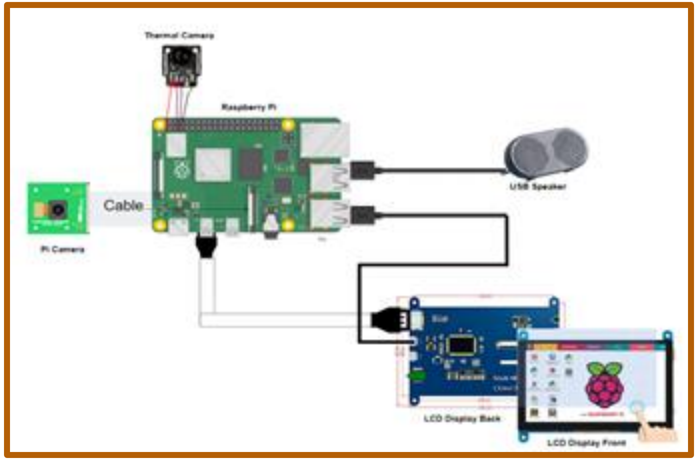
- MLX90640 IR Thermal Camera (connected over I2C)
- Raspberry Pi 4B IR Camera Module 75 (connected over CSI interface)
- USB Speaker (connected over the USB protocol)
- Raspberry Pi Touch Screen 5 inch Monitor LCD Display (connected over HDMI and USB)

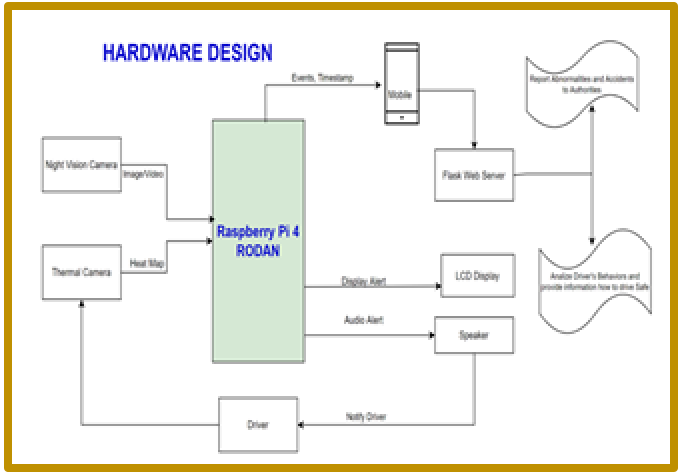
The code on the Raspberry Pi was written using python.
Following libraries were used for connecting with hardware device and bluetooth

- pygame for display device
- pyttsx3 (text-to-speech) for speaker integration
- pybluez and `flutter_bluetooth_serial` for bluetooth communication
- opencv for connecting with RPi camera
- adafruit_mlx90640 for connecting thermal camera.

The Display device and speaker were integrated successfully.

Schematic Diagram

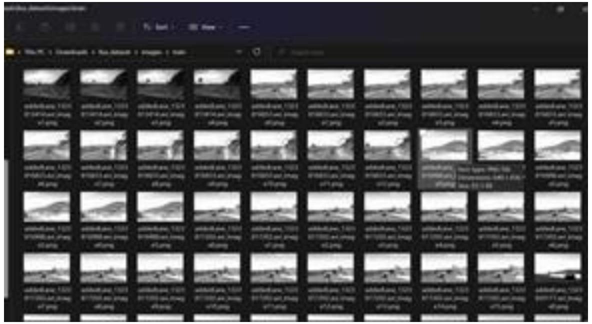




Diagrams prepared by project team

Methods: Phase 2 - Performing Object detection

- pre-recorded videos and images of road signs and objects fed to library.
- Initially OpenCV image processing library was used to check the bounding boxes.
- Once the code for component connectivity and object detection worked fine, then a pre-trained deep learning model YOLOv5 was used for Object detection.
- YOLO ('You Only Look Once') is an object detection algorithm that
  - performs inference across the entire test data set at once and runs very fast
  - divides images into a grid system and performs Hierarchical Feature learning and Multi-class prediction.
- Steps of object detection algorithm:
  - copy the weights of the trained model
  - feed video streams to the pretrained model
  - model infers the objects from the frames
  - store the detected objects and counts
  - create an alert message based on the type of object
  - send the messages to display and audio device.
- Finally, new PyTorch weights were generated by training the model in a powerful deep learning machine. Additionally, weights from other pretrained models were also combined to enhance the above algorithm and widen the coverage of objects.



Snapshot of road sign images(≈18k Dataset LISA)

Methods: Phase 3 - Develop Phone App and Event Reporting Dashboard

- A mobile app was developed using Flutter App.
- The messages are displayed in mobile app as soon as they are transferred using Bluetooth.
- The mobile app then sends the events to the database which is implemented using MongoDB hosted in the cloud for scalability.
- A Flask App was developed and hosted in Cloud to show the Dashboard.
- The Dashboard refreshes itself to show the statistics in the table.Dashboard reflects detail and comprehensive event details.
- Device will act as a Valuable Data Provider and Analytics Tool. The collected data will allow City officials to find important patterns in traffic incidents and plan traffic improvements accordingly.
- It will also help analyze the data to unearth valuable insights about different types of events and predict road congestions and critical issues.

Phone App and Dashboard by team

It Installed on Car Dashboard

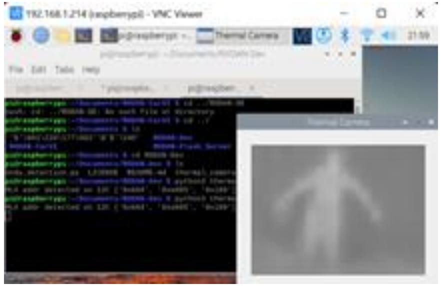
Methods: Testing Procedure followed across all the Development phases

- Object detection and Image analysis functionality was tested by feeding pre-recorded videos, images and live video containing various road signs and traffic situations during different time periods and lighting conditions.
- Initial testing was done against basic OpenCV and simple MobileSSD Net
- Final testing was done against YOLOv5 model which was trained with 18k of images in a powerful Kaggle Environment over multiple iterations.
- Model accuracy was captured during Machine Learning training process by instrumenting the pipeline in the test environment.
- The latency of object detection and processing was also calculated for different phases of the process.
- Different charts were created to analyze the performance of the model.
- Finally, it was tested if the detected object is same as the ground truth and if correct messages were displayed.
- Sanity testing was performed to verify the statistical counts of events with the actual count in Dashboard

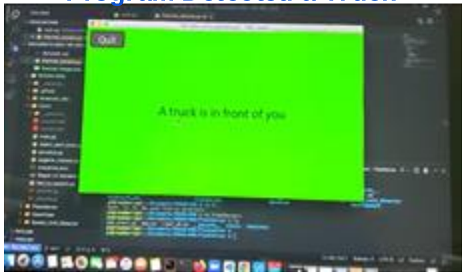
Results :

- Heatmap, image pixels, bounding boxes were created as a result of the testing process. This was required for the compatibility and performance of the camera.
- Verification of object with ground truths helped measure Accuracy Rates for pedestrians, traffic lights, stop signs, speed limits
- First phase of testing with a deep learning model MobileSSD and OpenCV reported latency 500 ms with accuracy 80%
- Second phase of testing with YOLOv5 resulted in latency (< 100 ms) and coverage for 40 objects with accuracy 90%
- Third phase of testing which involved custom-trained YOLOv5 covered 150 objects. with average accuracy 95%. The model itself was tested over 100s of iterations.
- The Alert messages were cross-verified with what was displayed , announced and what was stored in database.
- Statistical counts were calculated for all the events.

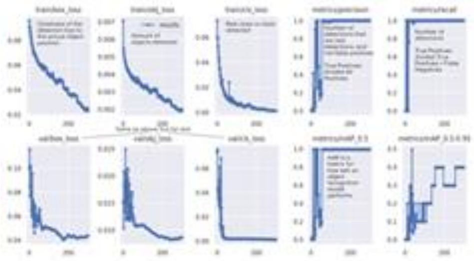
Thermal Camera Detecting a Person



Program Detected a Truck

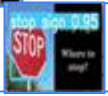
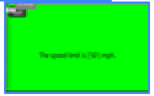
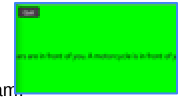


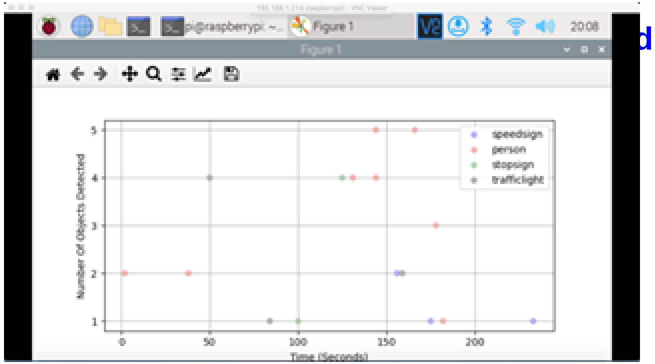
Deep Learning Model Detecting Objects Matrix



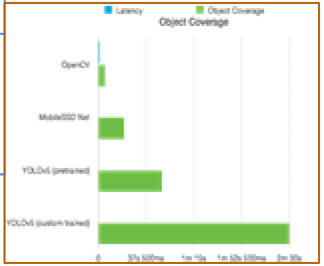
Testing screenshot by team

Results: Test Input, Output and Observations

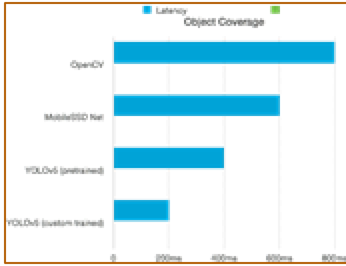
Num#	Type of Videos/Images	Detected Image by Our device	Alert Message	Model Training Accuracy	Testing Accuracy	Time Taken to Detect Objects
1	Pedestrian		Pedestrian ahead	90%	95%	0.3 Sec
2	Traffic Light		Detects Traffic Light	95%	98%	0.5 Sec
3	Stop Sign		Stop Sign Detected 	98%	98%	0.3 Sec
4	Speed Limit		Speed Limit Sign Detected 	98%	98%	0.3 Sec
5	Heavy Traffic		Heavy Traffic Detected after complex rules executed by program. 	98%	98%	0.3 Sec
6	Live testing while driving on City Road	https://youtu.be/WstS0Gf5iQA	All above Models are running	98%	98%	~0.2 sec



Coverage of Object Categories for per algorithm



Latency per Algorithm



Testing screenshot by Team

Discussion

- Analysis of image pixels, heat maps helped identify compatibility issues with a camera library early on. The issues was rectified quickly.
- Rigorous testing forced us improve the implementation approach by training YOLOv5 with a large number of images (20k) and variety of object classes (LISA) in a powerful GPU env. This enhancement from prior approach (using pretrained model) drastically reduced the latency and improved the accuracy with a higher coverage of objects as shown in Results. We were able to achieve the result as per the best industry standard.
- During the testing process, an unexpected issue occurred where that Display device, Speaker and Bluetooth failed to run in parallel in different processes. So an interesting solution was devised to parallelize the message communication using client-server socket communication.
- It was observed that same events were reported multiple times during initial quality analysis phase. This prompted the design of rule-based approach to derive a traffic situation like ‘Heavy Traffic building ahead’ from the count and occurrences of individual objects. Thus one critical event replaced multiple occurrences of individual events.
- After addressing the key problems and enhancing the product, the inter-connected safety device could generate instantaneous traffic alerts. It is extremely useful for all types vehicles (even modern vehicles only have collision detection, lane departure and car issues and definitely need to a solution to get timely alerts for traffic situations) . Specially old automobiles, electric wheelchairs and drivers who can’t afford costly devices will benefit immensely from this device.

References

https://www.asirt.org/safe-travel/road-safety-facts/

United Nations Launches New Initiative to Reduce Road Crashes https://visionzeronetwork.org/united-nations-launches-new-initiative-to-reduce-road-crashes/

[Rapid DNA origami nanostructure detection and classification using the YOLOv5 deep convolutional neural network] https://www.nature.com/articles/s41598-022-07759-3

[A Survey of Research on Crowd Abnormal Behavior Detection Algorithm Based on YOLO Network]

Credit: Public traffic videos , Images are captured by project team by driving around the city,highway.

Some initial testing videos were created by adding some video frames together