CAPITOL Technology University

RoadRunner™ Design

by

Bilal Ashiq

Kireh Wright

SE-457 Senior Design 1

Fall 2018

Advisor: Professor Mohammad Malkawi

Table of Contents

Abstract

The RoadRunner Project will test and evaluate the applications of machine learning and computer vision as they intersect big data aggregation and processing. The Project will offer a crowdsourced reporting system for infrastructure issues for use by drivers and with planned development for easy transfer to vehicle manufacturers. So far in this project we derived business requirements and began configuration for our Raspberry Pi test platforms. Next semester, we plan on developing a Minimal Viable Product (MVP) by testing different computer vision and object recognition algorithms.

1. Introduction

1.1. Motivation:

1.1.1. As more individuals and organizations are contending the possible applications of artificial intelligence, firms are looking for employees with a vision for how AI can disrupt our current way of living. For instance, the use of artificial intelligence can revolutionize the management and repair of infrastructure. Incorporating AI into local planning with tools like RoadRunner can streamline issue reporting and preemptively detect degraded infrastructure with minimal long-term increases in budgeting requirements. In addition, this project can be expanded to provide anonymized data on commuter choices/habits and be used to plan new infrastructure as traffic patterns evolve or monetized as an open source research data aggregator. In addition to offering a glimpse into how AI might fit into the most mundane of tasks, this project will give us a platform on which to build a useful AI implementation for a general task.

1.2. Background:

1.2.1. The project was chosen since we wanted to build a complete product that includes hardware and software for detecting potholes. Potholes cause serious damage to vehicles and pose a threat to motorist resulting in accidents.. According to Quigg (n.d), potholes cause billions of dollars of damage to cars every year. According to Li(2018), the current way to report potholes is by going to your state government website and reporting it through their services, and as we all know this service is at best so-so. With RoadRunner, the plan is to come up with a solution that can identify potholes and in the future be reported directly to the necessary agencies. RoadRunner will use Artificial Intelligence (AI) more specifically camera vision within machine learning for detecting of the potholes. The purpose of the project is to find the most optimal and accurate algorithm that will be capable of detecting a pothole on the road.

1.3. Objective:

1.3.1. The objectives for the RoadRunner project include:

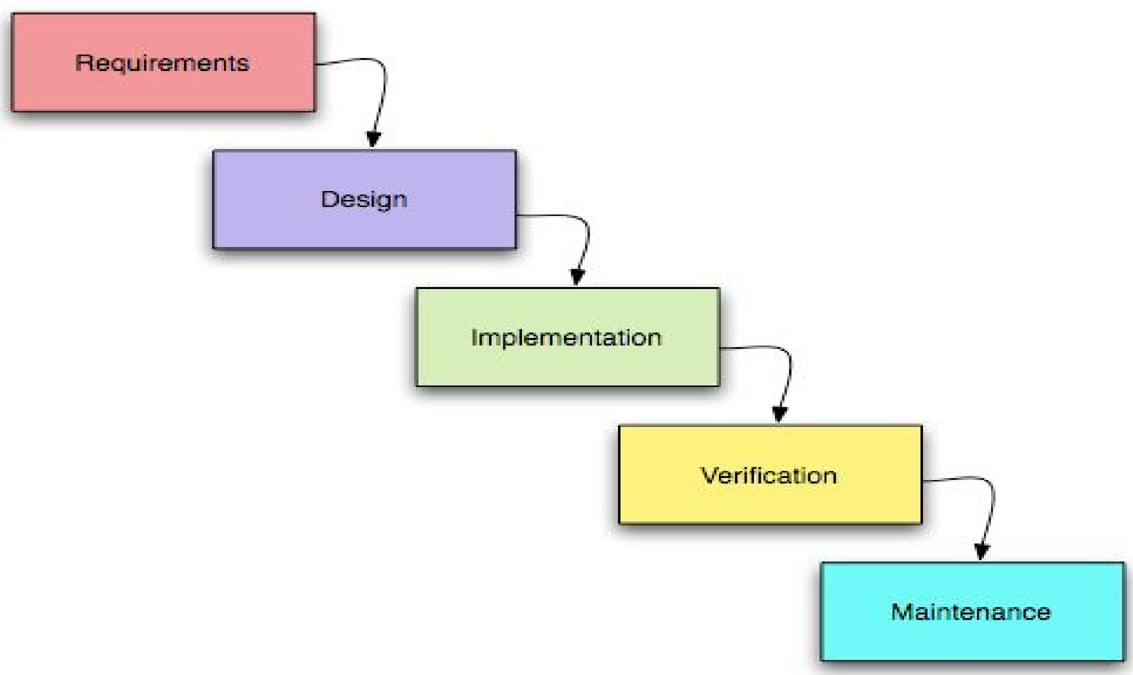
1.3.1.1. To evaluate and tune machine learning resource management algorithms to optimize detection rates and minimize false-positives.

1.3.1.2. To use hardware that is affordable and easily accessible, maximize participation rates, and deliver results that are easy to replicate in other markets

1.3.1.3. To highlight the economic advantage of leveraging machine learning and big data aggregation in local infrastructure planning

1.4. Methodology: Waterfall methodology

In our project, we've decided to implement a traditional waterfall planning system due to the accelerated timeline of the project and the overhead that would be introduced by trying to maintain an Agile environment between two developers. Although our development cycle will include periodic meetings and discussion of progress, we will not be releasing successive iterations of our project as a result of these meetings.



2. Project Initialization & Planning

2.1. Project Organization:

2.1.1. Group Name: Silkweave

2.1.2. Group Members:

2.1.2.1. Bilal Ashiq (Developer & Business Analyst)

2.1.2.2. Kireh Wright (Developer & Business Analyst)

2.2. Project Scope:

2.2.1. Stakeholders

2.2.1.1. Typical non-commercial Drivers

2.2.1.2. Capitol Technology University Staff

2.2.1.3. Maryland State Highway Administration

2.2.2. The system software shall be freely available to the general public for installation as an open source standard with reference hardware.

2.2.3. The system shall be able to identify potholes successfully at speeds of up to 35 mph using the Pi Camera.

2.2.4. The system shall be easy to install and integrate into a vehicle.

2.2.5. The development period will run from December 2018 to April 2019. Once the development period finishes, development will cease regardless of actual progress.

2.3. Assumptions:

2.3.1. We make the following assumptions for the proposed system:

2.3.1.1. We will define a pothole as a structural failure in the road surface that occurs as result of fatigue and cracking in response to the repeated freezing and thawing of water in cracks in the road surface. Approximation dimensions are greater than: 9” x 9” x 3”

2.3.1.2. We expect that the users of the system drive at speeds up to normal city limits for Baltimore, MD (maximum of 35 MPH).

2.3.1.3. We expect that the system will not be used on freeways, interstate highways, nor in any other environment where speeds may exceed 50 MPH.

2.3.1.4. We will be designing our product under optimized weather conditions at first: plentiful ambient lighting, with no rain, fog, or other visibility issues.

2.3.1.5. The system will be used in an environment that maintains an appropriate power supply during expected use.

2.3.1.6. The system will be physically shielded from the elements.

2.3.1.7. We assume the system will only be used to capture imagery of road conditions. It should not be regularly exposed to any other environment (e.g. indoors or unpaved environments).

2.3.1.8. We assume that in some cases, the same individual may function as both driver and admin users.

2.3.1.9. Reliability is contingent on hardware: if the camera doesn't send a signal back, or the board proves to be faulty, the system would ultimately fail and need replacement.

2.4. Risk & Constraints:

2.4.1. We anticipate the following constraints which may cause risks to the project:

2.4.1.1. A major constraint is availability for our team members. Both members maintain jobs as well full time academic careers. It is necessary that we allocate enough time to do the project and start early in order to allow for necessary slack.

2.4.1.2. Our team has not had exposure to computer vision applications. We are building and running an application for a Raspberry Pi Model 3 B+ which is also a new technology we will be using. As a result, we will be forced to learn take on the challenge of learning the new technologies.

2.5. Project Scheduling

2.5.1. Gantt Chart is attached

3. Analysis

3.1. Techniques Used To Collect Requirements:

Stakeholders	**Elicitation Method**	**Reason**
Typical driver	Questionnaire	There are millions of drivers in the US, so we have a large, random pool of participants
Capitol Technology University staff	Interview	In this case we must interview the department chair for project complexity requirements
Regulatory Government Agencies including the Maryland State Highway Administration (SHA)	Review policies and regulations.	Government agencies create documentation to be made widely available, providing a common baseline to an industry/field

Figure 3.1 *-Requirements are collected using the techniques described in the table*

3.2. User Functional Requirements:

3.2.1. The following requirements pertain to the driver's perspective:

3.2.1.1. The system shall provide for drivers to record their driving session.

3.2.1.2. The system shall prohibit the driver from uploading unverified pictures through the application.

3.2.2. The following requirements pertain to the admin's perspective:

3.2.2.1. The system shall provide for admin to review images taken from the Raspberry Pi.

3.2.2.2. The System shall provide for admin to tune the algorithms that detect the target object.

3.2.2.3. The System shall provide for admin to give feedback (photo verification) to the machine learning algorithm.

3.3. System Requirements:

3.3.1. The system shall provide a camera to take images of the nearby surroundings.

3.3.2. The system shall process images provided by a camera system, identifying any potholes present.

3.3.3. The system shall log the location of the images flagged or otherwise identified by the algorithm.

3.3.4. The system shall provide local storage to hold images and accompanying metadata.

3.3.5. The system shall be able to connect to the internet via Wifi or Ethernet connections.

3.3.6. The system shall provide a mode of record keeping for verified potholes or other findings.

3.3.7. The system shall store images at 720p.

3.4. Nonfunctional Requirements:

3.4.1. Availability & Reliability

3.4.1.1. The system shall only have to be set-up initially by the driver. Future boots will happen on power with reusable configuration.

3.4.1.2. The system prototype shall have an MTBF of at least 100 hours.

3.4.1.3. The system shall be available whenever powered on by the user. When the device is powered on, we estimate the system will take 1 minute to boot up and be ready for action.

3.4.2. Maintainability

3.4.2.1. The System shall require the admin to verify images taken by the Raspberry Pi camera.

3.4.3. Performance

3.4.3.1. The system shall be based upon a Raspberry Pi Model 3 B+ hardware; the system will capture a 30 frames per second for analysis. Portability is limited to computers in the Raspberry Pi Model 3 B+ series using the corresponding proprietary peripherals.(speeds no greater than 30mph)

3.4.3.2. The system shall have performance of different algorithms quantified using a weighted matrix based upon detection rates, resource (processor, RAM, and storage) utilization, and false positive rate.

3.5. Context Diagram:

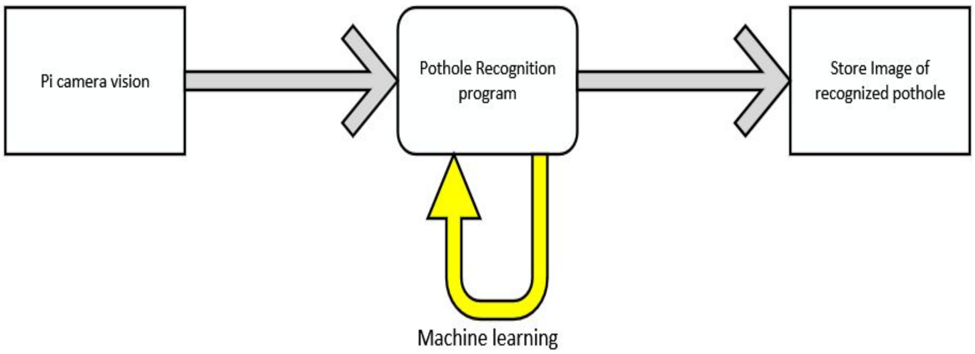


Figure 3.5- *This image depicts the context diagram in relation to the software*

3.6. Functional Decomposition:

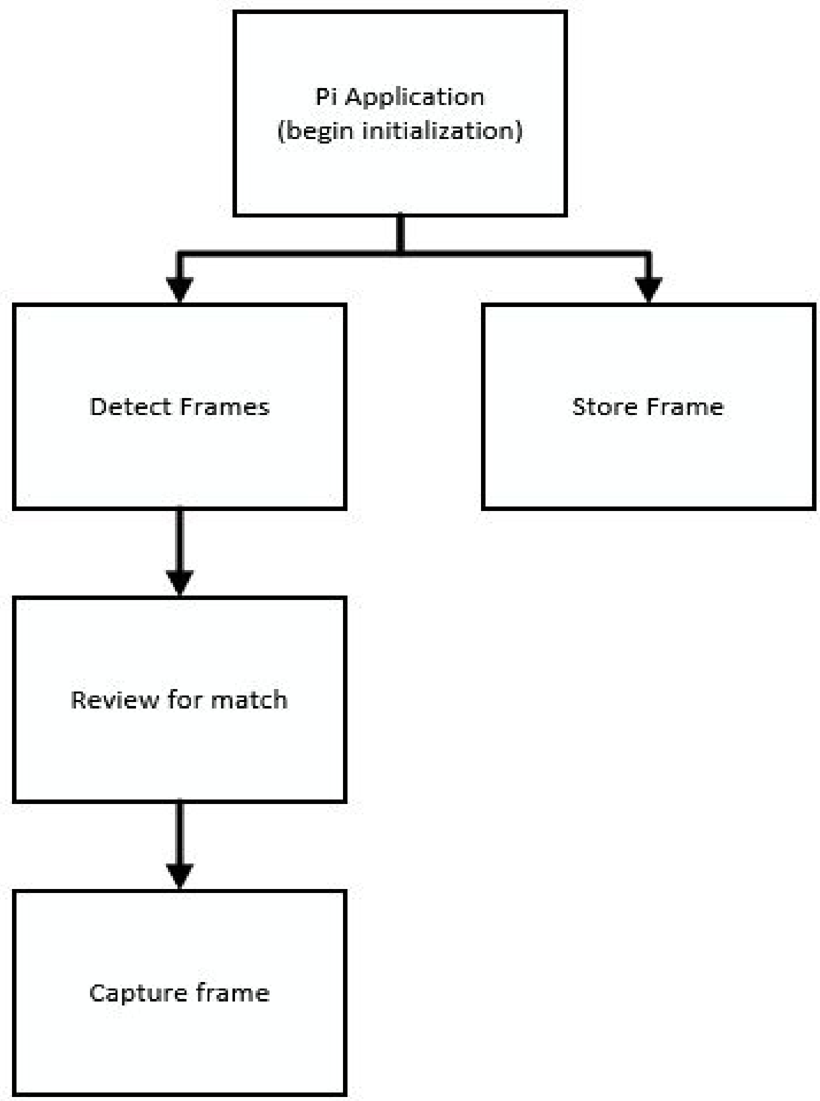


Figure 3.5- *Functional decomposition of the pothole detection software*

3.7. Use Cases:

3.7.1. Driver

3.7.2. Infrastructure Use Case

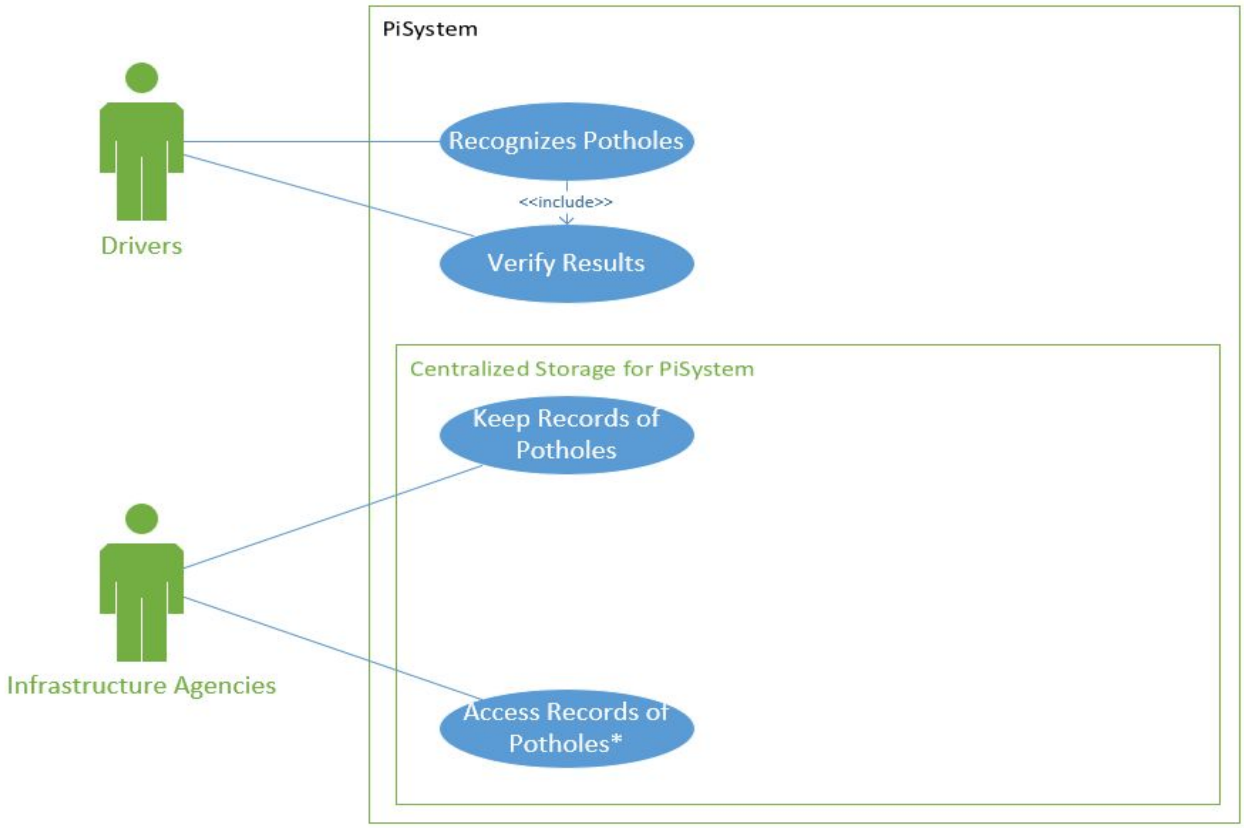


Figure 3.7 - *Use case scenarios for motorist and agencies*

3.8. Time Sequence Diagram:

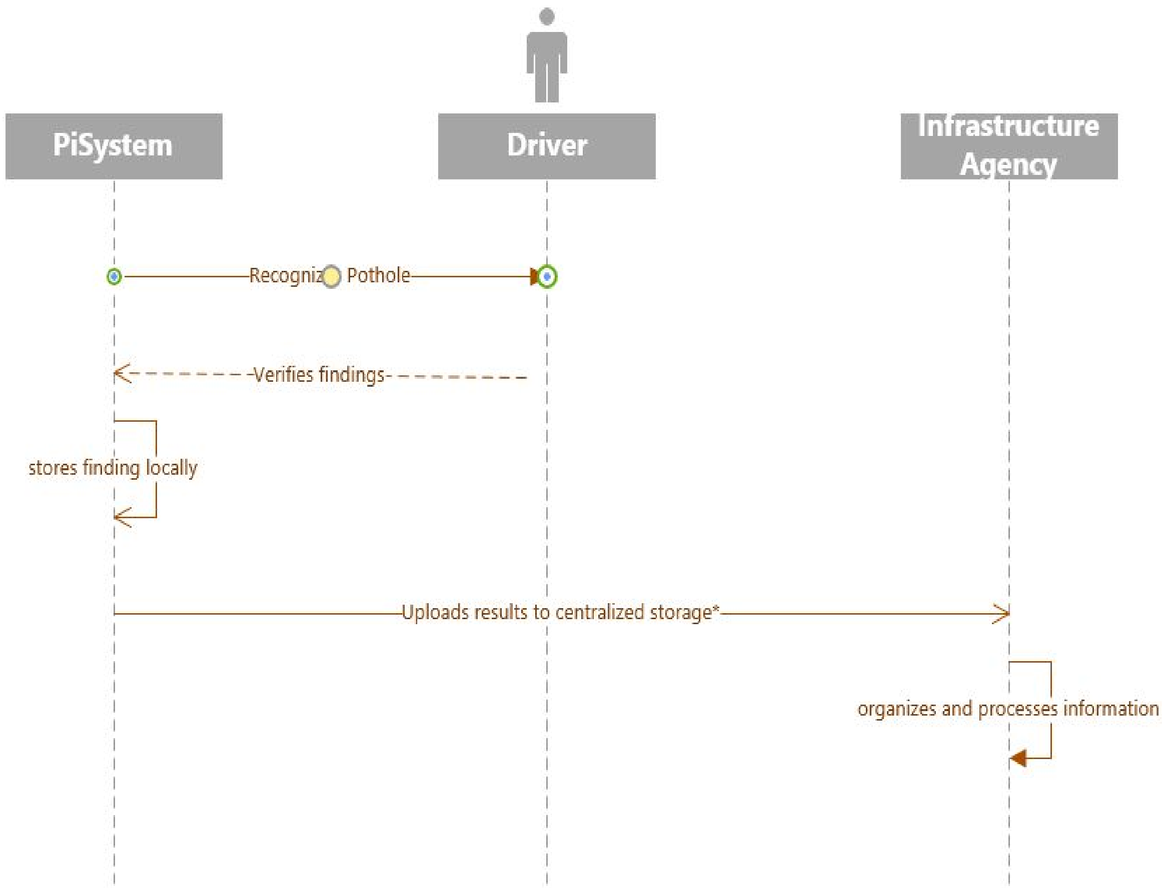


Figure 3.8- *Time sequence diagram of the RoadRunner software*

4. Conceptual Design

4.1. Mock UI

A web application will be built in order to communicate with the Raspberry Pi(it will also be hosted on the Pi). The website will be accessible from the Pi or another device as long as both devices are on the same network. Figure 4.1(A) is the homescreen for the web application. There will be two actions a user may perform: Roader will allow the admin to verify the images captured by the Raspberry to see if in fact they are potholes. This is crucial because here we can determine how well the model is actually learning. Gallery will display all verified images that contain potholes.

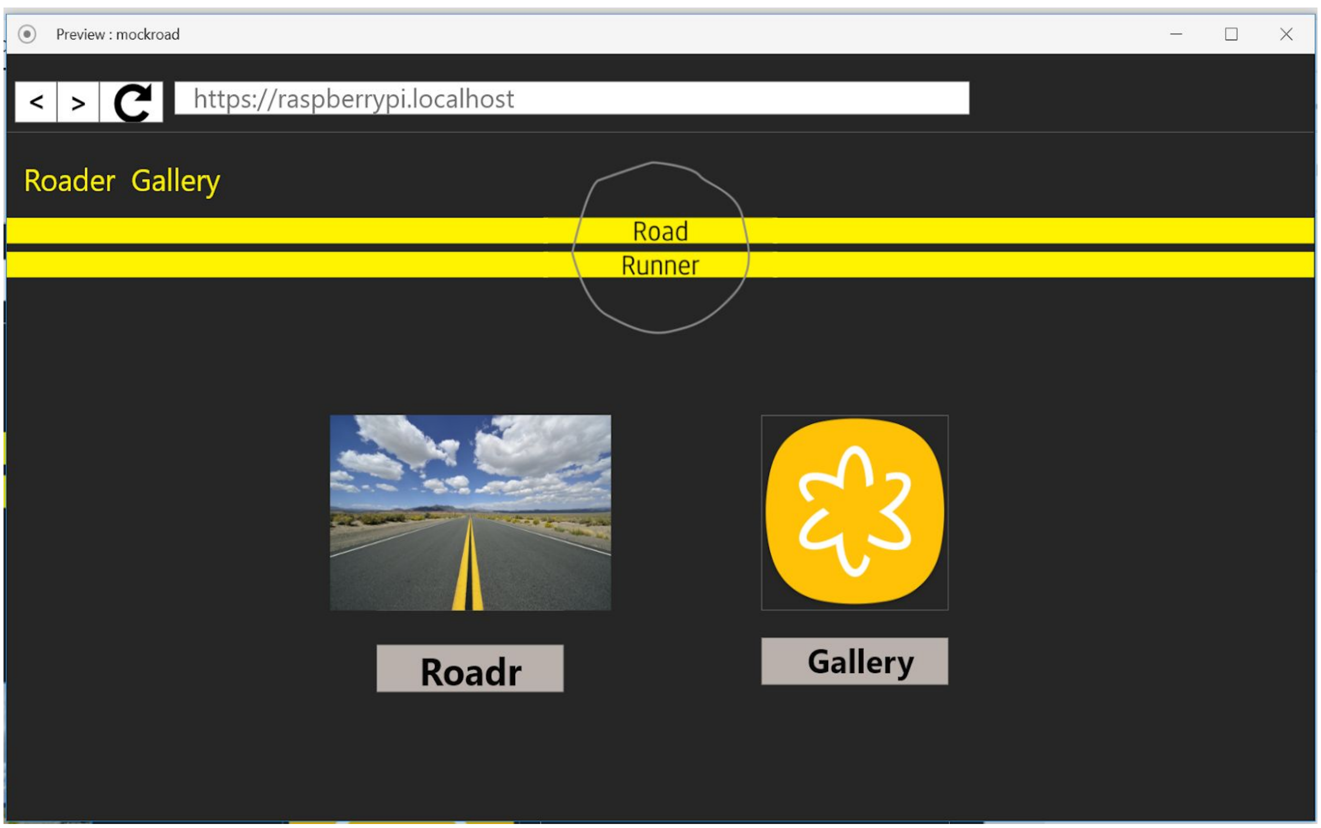


Figure 4.1 (A)

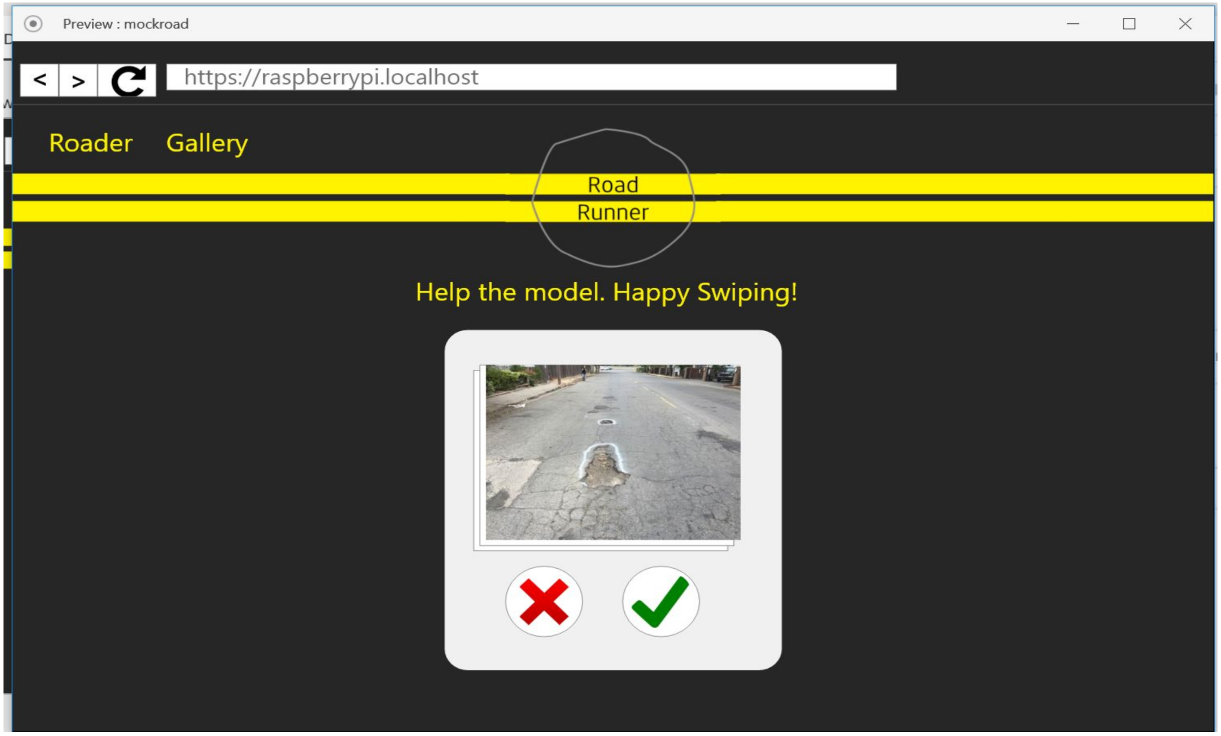


Figure 4.1 (B)- *Since the model is using assisted learning, here the admin is able to help the model decide if the selected image contains a pothole if the algorithm generates a result with less than 50% confidence.*

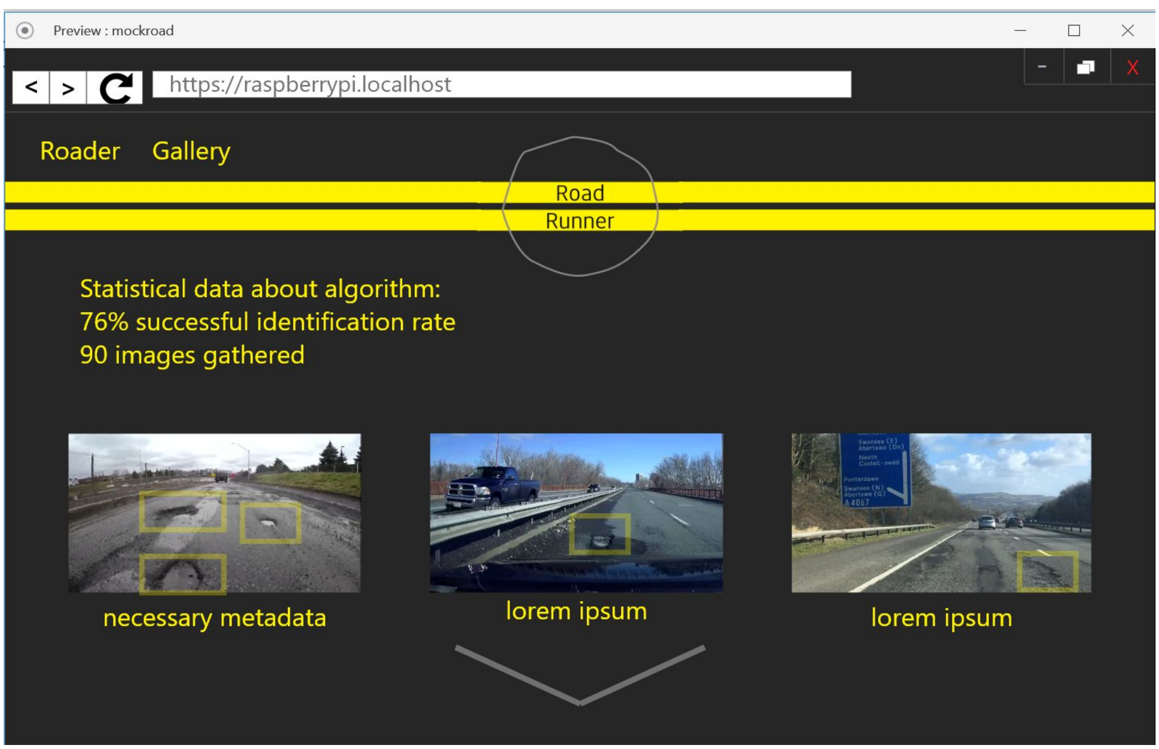


Figure 4.1 (C) - *A gallery of images can be observed. The yellow boxes highlight the identified potholes.*

5. Architecture Design

5.1. Class diagram

There will be two classes needed for this application. PotholeRecognizer will take data in from the Pi camera frame by frame and check for matches. If a match is found saveFrame() and getTimeNDate() will be called. This class will run when the Raspberry Pi is headless will be on the road scanning for potholes. ImageServer will be used when an admin goes to access the data stored locally by the Pi. As defined above in the mock UI, there will be several functions including one to view images taken by the Pi.

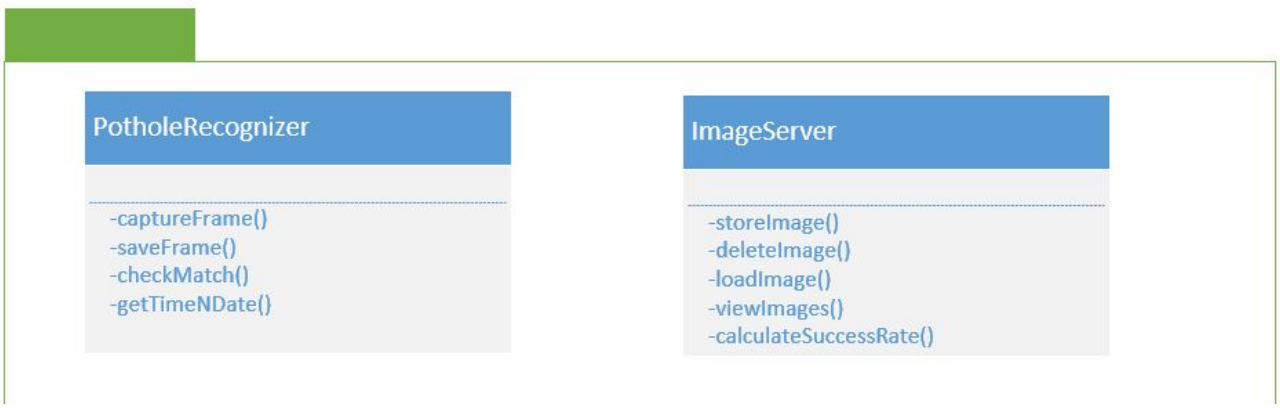


Figure 5.1

5.2. Infrastructure Physical design

The Raspberry Pi system will have two peripheral devices connected to it one being the camera and the other a micro-SD card.The raspberry Pi also contains a network adapter which will allow us to connect to it directly via wifi direct. The system will host its own server in which any device connected to it will be able to access it. The current setup will all the user to use the system as a dashcam as well as a server during its operation.

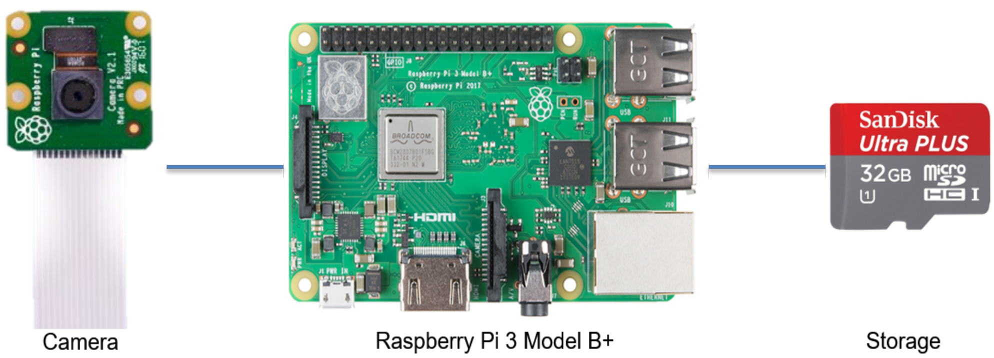


Figure 5.2

6. Tools and Technologies Used

6.1. Software Tools and Technologies

6.1.1. Raspbian 4.14 for the Raspberry Pi Operating system.
6.1.2. OpenCV for real time image recognition.
6.1.3. Github for collaboration amongst teammates.
6.1.4. Python for the programming language

6.2. Hardware Technologies

6.2.1. We are using a Raspberry Pi 3 Model B+ system with the following specs:
6.2.1.1. Quad core 64-bit processor clocked at 1.4GHz
6.2.1.2. 1GB SDRAM
6.2.1.3. Micro SD port with and SD card size of 32gb
6.2.1.4. USB and HDMI port
6.2.1.5. Linux Platform(Raspbian 4.14)
6.2.1.6. Raspberry Pi Camera Module v2
6.2.1.6.1. Resolution of 1080p30 or 720p60

7. Summary

7.1. In this project we have learned so far:

7.1.1. How to identify key stakeholders in a project
7.1.2. How to identify functional and nonfunctional requirements
7.1.3. The business process of identifying a project and documenting all necessary attributes within
7.1.4. How to successfully deliver assignments with team members by having clear communication

7.2. Expectations for Senior Design part 2

During the Senior design 2, we expect to start coding our solution according to the requirements outlined in this document. Collaboration will occur over github. The project tasks and progress will tracked with a Trello board. Once a prototype is complete, the application will be tested for bugs and to ensure all requirements have been met according to stakeholders. In addition, some more documents will be created such as a user manual to help users navigate this application.

References

Li, Y. (2018). *Road Pothole Detection System Based on Stereo Vision.* (Electronic Thesis). Retrieved from https://etd.ohiolink.edu/

Products. (n.d.). Retrieved December 4, 2018, from https://www.raspberrypi.org/products/

Quigg, B., Chin, D., Catalano, M., & Mir, M. (n.d.). Pothole Tracker. Retrieved December 4, 2018, from http://www.ecs.umass.edu/ece/sdp/sdp16/team05/documents/PDR.pdf