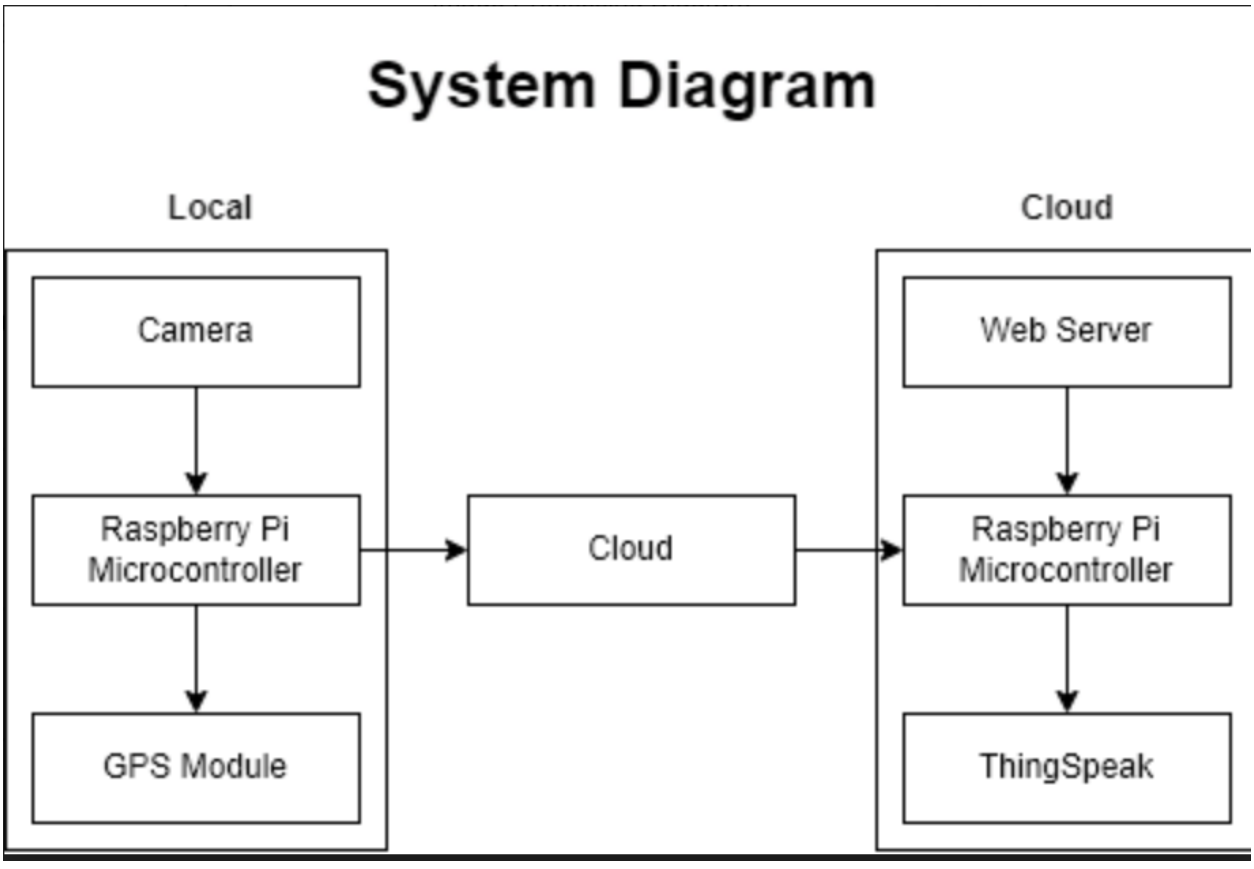
Senior Design Project Monthly Report for 9/2023

Pothole Detection System

Braeden Kurz (Computer Engineering)

Faculty advisor: Dr. Regentova

Current updated system diagram



Functions, modules, parts that have been accomplished in the past month.

Progress was made to improve the accuracy and speed of the pothole detection function. Rather than use Roboflow for training, I used the Google Colab notebook https://github.com/freedomwebtech/yolov4tinyrpi4/blob/main/yolov4_tiny_custom_TRAINING.ipynb for training. The process took over three hours with over 3000 iterations of training, resulting in a Mean-Average-Precision (mAP) of 70.1% compared to the previous 59.1%.

test.py

D: > yolov4tinyrpi4 > test.py

```
1   from cvlib.object_detection import YOLO
2   import cv2
3
4   cap=cv2.VideoCapture(0)
5   weights="yolov4-tiny-custom_best.weights"
6   config="yolov4-tiny-custom.cfg"
7   labels="obj.names"
8   count=0
9   while True:
10      ret,img=cap.read()
11      count += 1
12      if count % 10 != 0:
13          continue
14      img=cv2.resize(img,(680,460))
15
16      yolo = YOLO(weights, config,labels)
17      bbox, label, conf = yolo.detect_objects(img)
18      img1=yolo.draw_bbox(img, bbox, label, conf)
19      cv2.imshow("img1",img)
20      if cv2.waitKey(1)&0xFF==27:
21          break
```

The test.py script is used for testing the video recording detection function. The script reads the YOLOv4-Tiny weights generated from training the dataset, the config file, and the obj.names file which contains the label ‘potholes’ in reference to the annotated images containing potholes. Since we are detecting and labeling potholes in real time the video is checked every time the count value is divisible by ten. A single frame is picked from the video and the pothole detection unit analyses and labels the detected pothole. The script currently does not work due to an issue with an outdated tensorflow python package. The operating system needs to be updated to the Debian Bullseye operating system to support the more recent tensorflow python package. Additionally, Tensorflow 1.14.0 had an issue with a missing object file called libhdfs.so that could only be resolved by updating the Tensorflow to a more recent version, but current operating system did not support it. The SD card also corrupted, and while the YOLOv4-Tiny files and training data was saved, the operating system needed to be replaced and updated.

Table of specification & design restrictions

Function module	Specification (use bullets)	Why chooses this specification?
Camera Module	• **Input Power:** 1.5V to 3.0V DC • **Peak Current:** 300mA • **Sensor:** OV5647 • **Connection:** 15 cm flat ribbon cable to 15-pin MIPI Camera Serial Interface (CSI) connector • **Lens:** Fixed Focus Lens • **Angle of View:** 54° Horizontal x 41° Vertical • **Field of View:** 2.0 x 1.33m @ 2m • **Resolution:** 5 Megapixels, 2592 x 1944 • **Frame Rates:** 30fps@1080P, 60fps@720P, 90fps@480P • **Fixed Focus:** 1m to Infinity • **Dimensions:** 4.7 inches x 0.2 inches x 0.1 inches • **Weight:** 0.32 ounces • **Operating Temperature:** -30 to+70 °C	• Necessary to operate the camera module • Same as previous • Primarily daytime use • Cable necessary for interfacing on the Raspberry Pi 3 Model B+ • Allows for crisper images to capture. • Images need not be large • The potholes need to be detected at a reasonable distance
Raspberry Pi 3 Model B+ Microcontroller	• **Processor:** Broadcom BCM2837B0,	• **Processor:** Necessary for quickly processing signals

	Cortex-A53 64-bit SoC @ 1.4GHz • **Memory:** 1GB LPDDR2 SDRAM • **Connectivity:** 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless, LAN, Bluetooth 4.2, BLE, Gigabit Ethernet over USB 2.0 (maximum throughput 300Mbps), 4 × USB 2.0 ports • **Access:** Extended 40-pin GPIO header • **Video & Sound:** 1 × full size HDMI, MIPI DSI display port, MIPI CSI camera port, 4 pole stereo output and composite video port • **Multimedia:** H.264, MPEG-4 decode (1080p30); H.264 encode (1080p30); OpenGL ES 1.1, 2.0 graphics • **SD Card Support:** Micro SD format for loading operating system and data storage • **Input Power:** 5V/2.5A DC via micro USB connector, 5V DC via GPIO header, Power over Ethernet (PoE)–enabled (requires separate PoE HAT)	• **Memory:** Storing the OS • **Connectivity:** Connecting to the internet to send GPS coordinates of detected pothole to cloud • **Access: N/A** • **Video & Sound:** Necessary for interfacing with the Raspberry Pi 3 Model B+ to install the OS and camera module • **Multimedia:** Necessary for interfacing the camera module • **SD Card Support:** The SD card contains the OS for operating the Raspberry Pi • **Input Power:** The Raspberry Pi needs this amount of power to function • **Dimensions:** Small and lightweight for easy transportation • **Weight:** Small and lightweight for easy transportation • **Operating Temperature:** Safe operating temperature to avoid damaging the board and harming users

	• **Dimensions:** 85 x 56 x 17 millimeters • **Weight:** 7.1 ounces • **Operating Temperature:** 0 to +50 °C	
GSM Module	• **Input:** 3.3V to 5.0V @ 45.0mA DC • **Data Transmission Protocol:** GSM, TCP/IP, SMS, Bluetooth • **Operating Temperature**: -40°C to 85°C • **Dimensions:** 30.2mm x 65.0mm • **Weight:** 4.6 ounces	• **Input:** Power needed to operate the GSM module • **Data Transmission Protocol:** Uses GSM to transmit data to cloud network. • **Operating Temperature:** Safe operating to avoid damaging the module and harming the user • **Dimensions:** Small and compact to improve portability • **Weigh:** Lightweight to improve portability
Battery Pack	• **Input:** 6-AA Batteries • **Output:** ~9.0V DC (~5.0V DC w/ DC-DC Converter) • **Dimensions:** 3.625 inches x 2.25 inches x 0.625 inches • **Weight:** 0.634 ounces	• **Input:** Requires 6-AA batteries to supply Raspberry Pi • **Output:** Need to know how much the output must be reduced to avoid damaging the board. A DC-DC converter is needed to meet power requirements of the Raspberry Pi • **Dimensions:** Bulky and takes up most space • **Weight:** Lightweight to improve portability
Cloud Service	• **Input:** Coordinates of identified potholes • **Output:** Visual representation of pothole data • **Power Requirements: N/A** • **Data Transmission Protocol:** WebSocket Protocol • **Measurement**	

	Range: N/A • **Constraints:** Data should be transmitted in intervals of 30 seconds to prevent data congestion.	
Entire prototype	• **Input:** User credentials, video camera recording, battery power supply w/switch (Micro-USB) • **Output:** Data uploading to cloud, alarm notification • **Power Requirements:** 5V, 2.5A DC • **Data Transmission Protocol:** WebSocket Protocol, RS232 and TTL UART, 802.11 b/g/n/ac Wireless LAN • **Pothole Detection Range:** Roughly 10 meters • **Weight:** 9.814 ounces • **Constraints:** There should be enough light to detect the pothole and the vehicle should be moving below 45mph for the system to get a good quality video of the road.	• **Input:** The system • **Output:** The system needs to alert drivers of upcoming potholes and their locations • **Power Requirements:** Need to know how much power the system needs to be supplied • **Data Transmission:** To ensure standard data transmission protocols • **Pothole Detection Range:** Early detection of potholes helps warn users with enough time to react • **Weight:** N/A • **Constraints:** The camera needs enough light and speed to detect incoming potholes

Table of necessary standards:

Standard	Description	Link or References	Why this standard is employed/necessary?
IEEE Standard for Sensor Performance	This standard ensure that sensor performance data is consistent and accurate, which is important for many applications, including industrial process control, environmental monitoring, and medical diagnostics.	https://standards.ieee.org/ieee/2700/6770/	It helps to ensure that sensors are properly characterized and that their performance is well understood, which can lead to improved system performance and reliability.
Standard for Harmonization of Internet of Things (IoT) Devices and Systems	This standard defines a method for data sharing, interoperability, and security of messages over a network, where sensors, actuators and other devices can interoperate, regardless of underlying communication technology.	https://standards.ieee.org/ieee/1451.99/10355/	The standard defines a communication protocol and data model that allows for the exchange of information between smart transducers in a wireless network, regardless of the manufacturer or communication technology used.
IEEE Draft Standard for Automotive System Image Quality	The standard provides a framework for evaluating various aspects of image quality, including resolution, color accuracy, noise, dynamic range, distortion, and more. It also includes guidelines for selecting appropriate test charts and equipment, as well as instructions for conducting tests in	https://standards.ieee.org/ieee/2020/6765/	Mainly used in the automotive industry to ensure that their imaging systems meet the necessary performance requirements for safety and reliability.

	different lighting conditions.		
CC BY 4.0 License	Anyone is free to share, copy, and redistribute a work in any medium or format, and to adapt, remix, transform, and build upon the work for any purpose, even commercially. However, they must give appropriate credit to the original creator(s) and provide a link to the license.	https://creativecommons.org/licenses/by/4.0/legalcode	This license allows others to share, copy, and redistribute a work in any medium or format, and to adapt, remix, transform, and build upon the work for any purpose, even commercially, as long as they give appropriate credit to the original creator(s). This ensures that the original creator(s) receive proper recognition for their work, while also allowing others to benefit from and build upon it.

Functions, modules, parts that are in progress

Pothole Detection: A collection of images containing potholes and normal/well maintained roads were annotated and compiled into a dataset. The images were resized as per the requirements for training on the YOLOv4-Tiny model and augmented by adjusting the brightness to accommodate for changes in the lighting and camera. Currently, the detect.py script executes with errors caused by unsupported pythone packages. Work is being made to update the software and demo the pothole detection function.

Camera Module: Raspberry Pi Camera Module purchased. Raspbian Buster installed on Raspberry Pi 3 Model B+. Functions properly and no action is necessary

Alarm System: Piezo buzzers have been purchased. Still need to test the alarm function when a pothole is detected.

GPS Function: GPS module has been purchased. Currently working on installing the necessary libraries and writing code that parses through the GPS data collected.

GSM Module: Since the GPS module is used for tracking the position of the Raspberry Pi and the potholes in its vicinity, the GSM Module was considered unnecessary.

Power Supply: The Car adapter supplies the Raspberry Pi 3 Model B+ with enough power to operate with no warnings with the camera module, but more testing needs to be done with additional components to be certain.

Challenges (if any):

- *Bullseye operating system, as well as the Tensorflow, and opencv-python libraries are outdated, causing errors in the detection.py script*
- *The more recent Tensorflow libaries (2.x.x) are unsupported on the Bullseye operating system*
- *SD card was corrupted. Although the YOLOv4-Tiny files were saved, the operating system needed to be reinstalled*
- *Designing the system in a more compact manner*
- *Minor overheating issues*

Potential solutions:

- *Update libraries and operating to their more recent and supported versions (some of the more recent libraries have issues with the Raspberry Pi 3 B+*
- *Take a modular approach by building and testing each module before putting them together*
- *Carefully handle the SD card or replace the current one*
- *Read through the documentation of the Google Collab notebook and update the code*
- *System fan for cooling the Raspberry Pi and components*

Questions (if any):

-

Total hardware & budget cost so far:

Part Description	**Function**	**Amount Needed / Unit Price**	**Subtotal**	**Purchase Link**	**Datasheet Link**

Raspberry Pi 3 Model B+	Central controller for reading sensor data, alerting the driver and passengers of an upcoming pothole, and uploads the data to a cloud database for future reference and data visualization.	1/$35.00	$35.00	https://www.adafruit.com/product/3775?src=raspberrypi	https://datasheets.raspberrypi.com/rpi3/raspberry-pi-3-b-plus-product-brief.pdf
Arducam 5MP Camera Module OV5647	Camera module for scanning the road and detecting potholes in low-light conditions	1/$9.99	$9.99	https://www.amazon.com/Arducam-Megapixels-Sensor-OV5647-Raspberry/dp/B012V1HEP4?ref_=ast_sto_dp&th=1&psc=1	https://docs.arducam.com/Raspberry-Pi-Camera/Native-camera/source/OV5647DS.pdf
Coolwell Raspberry Pi GSM HAT GSM/GPRS/GNSS/Bluetooth Module **(NO LONGER IN USE)**	GSM module with GPS support used to send pothole coordinates to the ThingSpeak cloud when a pothole is detected.	1/$37.20	$37.20	https://www.amazon.com/Coolwell-Raspberry-Bluetooth-Module-Supports/dp/B0BS42WPGQ/ref=sr_1_8?keywords=raspberry+pi+gsm+mo	https://www.waveshare.com/w/upload/4/4a/GSM_GPRS_GNSS_HAT_User_Manual_EN.pdf

	The module also receives a confirmation message when the data is received by the cloud.			dule&qid=1677371035&sprefix=raspberry+pi+gsm+mo%2Caps%2C323&sr=8-8	
Jumper Wires (4in and 8in pack)	Connecting the modules to their appropriate pins to perform pothole detection and data uploading.	1/$6.49	$6.49	https://www.amazon.com/dp/B01L5ULRUA?ref_=dp_atch_dss_base_image	N/A
Cylewet 10Pcs 5V Active Buzzer	Buzzer component for alerting the driver of upcoming pothole	1/$6.98	$0.14	https://www.amazon.com/gp/product/B01N7NHSY6/ref=ewc_pr_img_1?smid=A2O4FZXIRZDLHA&psc=1	N/A
6-AA 9V Battery Pack	Power supply of the system	1/$6.99	$6.99	https://www.amazon.com/Battery-Holder-Enclosure-Connector-Cable/dp/B01N2INSBR/ref=sr_1_8?crid=1LM4N7NY5TJD2&keywords=6+aa+battery+pack+hold	N/A

				er&qid=1679283954&sprefix=aa+battery+pack+holder%2Caps%2C203&sr=8-8	
Part Description	**Function**	**Amount Needed / Unit Price**	**Subtotal**	**Purchase Link**	**Datasheet Link**
Total:		$58.61			

Employed software/program resources (e.g., open source codes, APIs, etc.)

Function	Key features	Source link/references
Raspberry Pi Tensorflow Package	A Raspberry P package necessary for running Tensorflow on the Raspberry Pi	https://github.com/tensorflow/build/tree/master/raspberry_pi_builds
Picamera Python Package	Software for interfacing with the Raspberry Pi Camera module	https://github.com/waveform80/picamera/tree/release-1.13
Darkflow Repository	Used to translate the Darknet API to Tensorflow since	https://github.com/thtrieu/darkflow/

	Darknet is a C++ based framework.	
Roboflow Platform	Website for creating	https://docs.roboflow.com/
Kaggle	Creating the dataset for detecting potholes. The dataset is divided into two categories: normal or well maintained roads and road that contain potholes. This division will help the system differentiate between well maintained roads and alert the user for potholes.	https://www.kaggle.com/docs/datasets