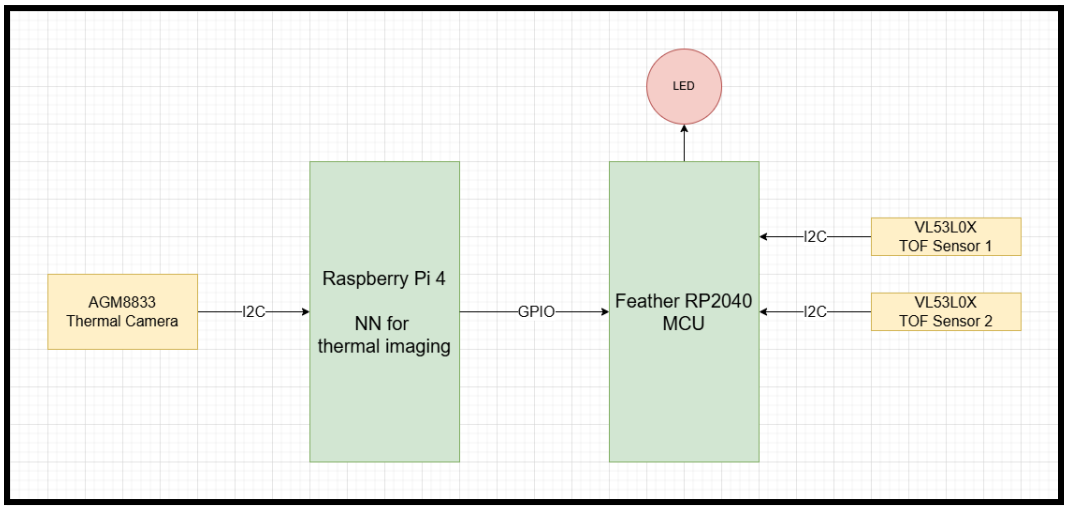
18848 Fall 2024
Final Project Report: Deer Detection

Amaan Kazi Akshat Sahay Swamynathan Siva

Abstract
The goal of this project is to design a real-time deer detection system for vehicles using embedded technology. We use sensor fusion (thermal imaging, TOF sensors) and onboard processing to catch deer near the road and warn drivers faster than they can react. If the system catches a deer, it can trigger alerts like flashing lights or braking assist to help drivers stay safe.

High-Level Block Diagram



Dataset Source, Method of Collection, and Data Cleaning
The source of this dataset were thermal images and distance measurements, which were collected via the thermal camera and TOF sensors, respectively. The first part of the data collection process was setting up the sensors on an RC car to simulate the car driving on the road. The thermal camera was attached to the windshield of the RC car while the TOF sensors were attached to the front bumper. A visualization of this setup is **linked here.** The second part of the data collection process was setting up the different scenarios for encountering a deer on the road. Specifically, we placed the deer at 3 different positions and collected the data by driving the car up to the deer. The table below shows the different scenarios tested.

Deer position	Deer Motion perpendicular to road* (from driver's perspective)	Collision possible
Right edge of road	Stationary	Maybe
Right edge of road	Right (away from road)	No
Right edge of road	Left (into road)	Yes
Center of road	Stationary	Maybe

Center of road	Right (toward car)	Yes
Center of road	Left (away from car)	No
Left edge of road	Stationary	No
Left edge of road	Right (toward car)	Maybe
Left edge of road	Left (away from car)	No

Additionally, a visualization of one of the scenarios for data collection is **linked here.** The thermal camera was connected to the Raspberry Pi 4, a script was run to collect the images, and the images were exported via a NumPy file and a MP4 video for visualization. Each scenario was collected 3 times where the deer's orientation was changed for every other collection. Additionally, we collected a similar number of images with the deer not being present. The script was set up to allow for collecting image data at 8 frames per second. The thermal images were converted to grayscale by normalizing the values to be between 0 and 255 and converted to 8-bit unsigned integers. For model training, the images were normalized to have values between 0 and 1.

Feature Extraction

There was no explicit feature extraction done for this project as the model was trained on the images themselves. For the TOF sensors, we discovered that there was no need to train a machine learning model to detect the deer (more discussion for this in sections below).

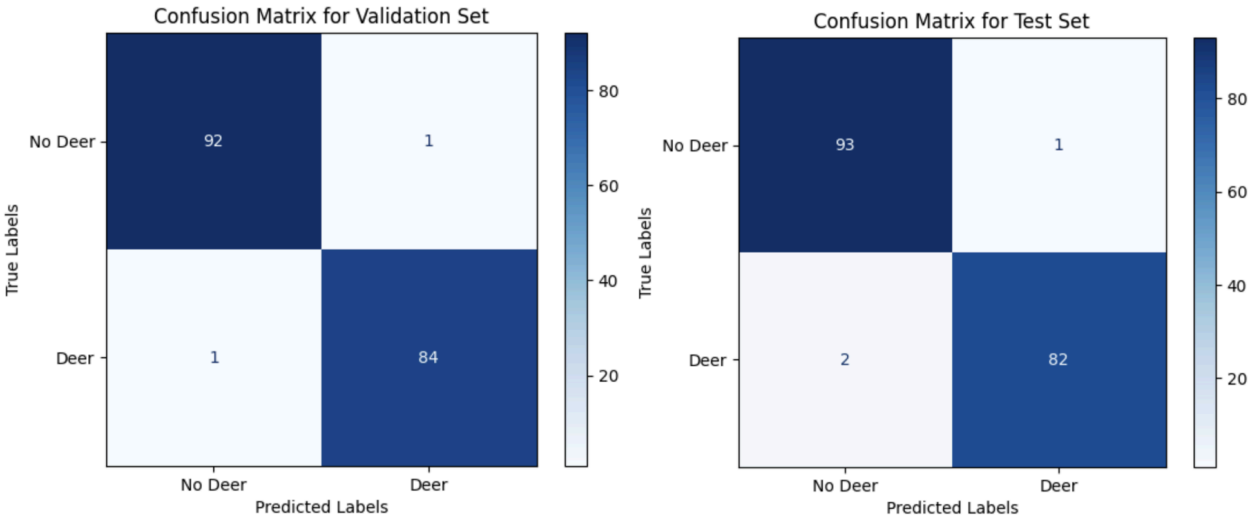
Classifier Architecture and Rationale

The classifier trained was a simple convolutional neural network (CNN). The architectures consists of a single 2D convolution layer with 64 filters and a kernel size of 3, a ReLU activation layer, and 2D max pooling layer with a kernel size of 2, a flatten layer, and a dense layer with 2 output neurons, and a sigmoid layer. The classification is purely based on whether the deer is present or not. A visualization of the model architecture is **linked here.**

The rationale for this architecture was to ensure that the model is lightweight yet captures as much information as possible from the image.

Confusion Matrix and Overall Accuracy Metrics

For training the model, we performed a 60/20/20 train/validation/test split on the data. Additionally, we performed quantization-aware training to ensure the model is more lightweight. Below are confusion matrices for this model on the validation and test datasets:



The model performed very well on the validation and test datasets, achieving 98.9% and 98.9% accuracy on both datasets, respectively.

Deployment Method: Hardware Target and Software Techniques

For the thermal imaging system, we deployed the model on the Raspberry Pi 4 using Tensorflow Lite using a script that loads the model and performs inference on each image as it is read in.

For the TOF sensors connected to the RP2040, we initially considered using Edge Impulse for processing the data to predict deer movement and add to the inference made by the camera. However, the output of the sensors did not have enough precision to train a quality ML model. Adding ML to something that can be done with traditional data processing would add unnecessary latency to our system, thus we just use thresholding for the TOF sensors to verify the model's inference. If the deer is detected by the CNN and the TOF sensors can verify that there is an object a certain distance away, an LED is triggered denoting an early stop alert.

Significant Challenges and Lessons Learned

We faced significant challenges in the process of designing our overall system and the individual components that enabled us to learn more, but also hamstrung our progress. One of the ToF sensors was miscalibrated, and they would not have enough clarity and stability in their readings to give us the fine grain readings we would need to decipher things like motion of the deer across the road using the difference in readings between the sensors. Another challenge we faced was with setting up the Raspberry Pi environment and ssh access, as well as getting a sufficient amount of frames per second from the IR camera. We also faced challenges with our machine learning model because the IR camera's output was in relative temperature, and so the color assigned to the deer and the surroundings would change from what was trained in the model, and cause it to not predict correctly. To remedy this, we ended up needing to retrain our model on grayscale images, which remedied the problem. Overall we learned how to adapt to real-world problems, identify reasonable compromises, and implement solutions. We got to get practical experience in the agile work cycle for embedded ML solutions.

Here is a link to our GitHub repository: https://github.com/amaank123456/Deer-Detection