Smart Helmet with Computer Vision

Kevin Gager, Aryan Barik

Advisors: Tyler Edmiston, Hossein Asghari, Lei Huang

Departments of Computer Science + Electrical and Computer Engineering, Loyola Marymount University, Los Angeles, CA 90045

Introduction

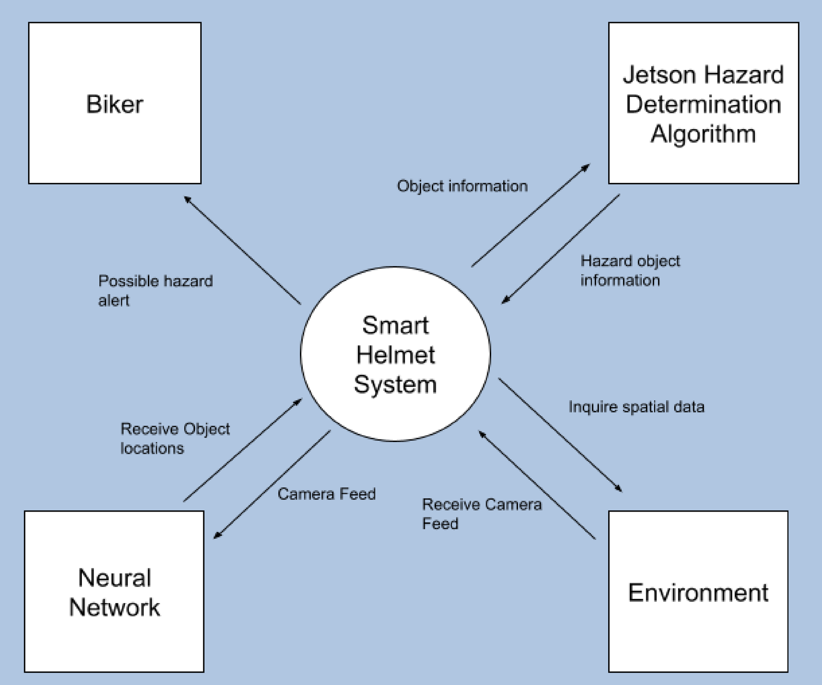
Many people are looking for alternative means of transportation. Bikes are one of the most efficient forms of non-polluting. However, bicyclists who share the road with cars take on a level of risk. This factor is amplified when car-dominated infrastructure has bicycle riding as an afterthought. Having to keep track of vehicles from potentially multiple directions can be distressing and dangerous, discouraging bike riding. Thus, a smart bike helmet that can intelligently determine possible threats in all directions is required. Using cameras embedded into the helmet, the helmet will receive a 360-degree video feed of a biker's surrounding areas. Objects in the feed will be evaluated on whether they are threats and the user will be alerted if they are. The proposed outcome of this project is to mitigate the chance of a biker being in a dangerous situation.

Proposed Solution

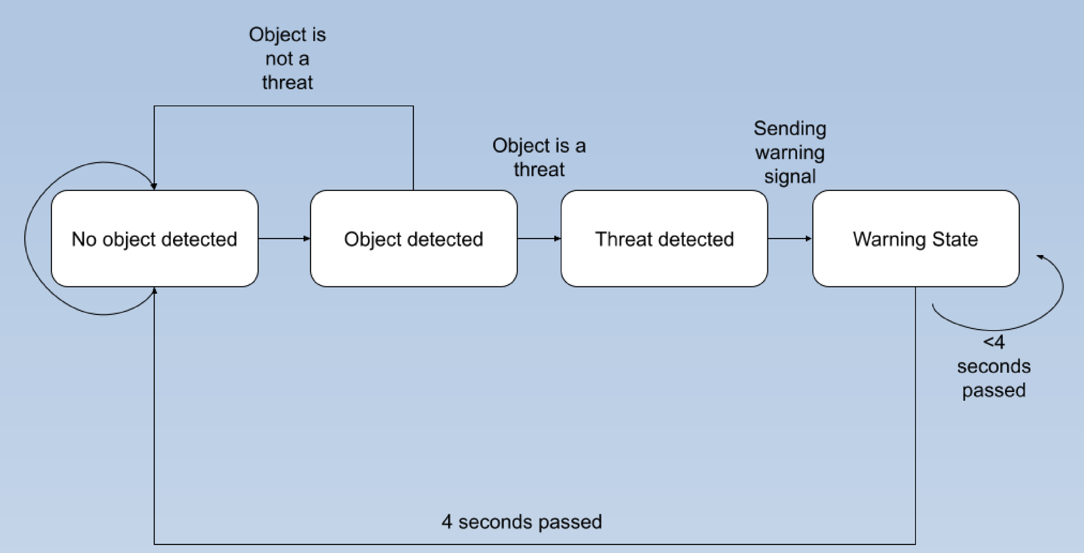
We wanted the system to prevent the most substantial threats that are preventable to the bicyclist. We determined these to be objects falling under 3 unique categories--cars, people, and potholes.

We planned on mounting a computing board onto a pre-existing bicycle helmet. The computing board would be attached to a set of video cameras that input a real time video feed of the bicyclist's surroundings. The computer board then uses that information to run a threat detection algorithm. If it indeed determines a threat, the computing board will light up a set of LED indicators, warning the user of the possible threat.

Context Diagram



State Transition Diagram



Technology Used

We decided to use the YOLOv5s objection detection model in order to detect the inquired objects. Unfortunately, YOLOv5's base COCO dataset does not include potholes as one of their detectable classes. As a result, we decided to gather and train our own pothole image dataset using Roboflow and Google Collab.

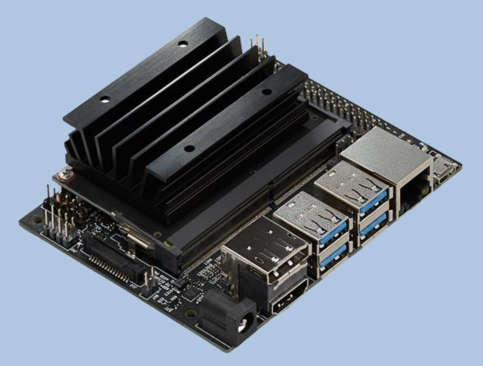
YOLOv5

We managed to gather over 2,000 images from self-collection and previously made datasets found on Roboflow Universe.

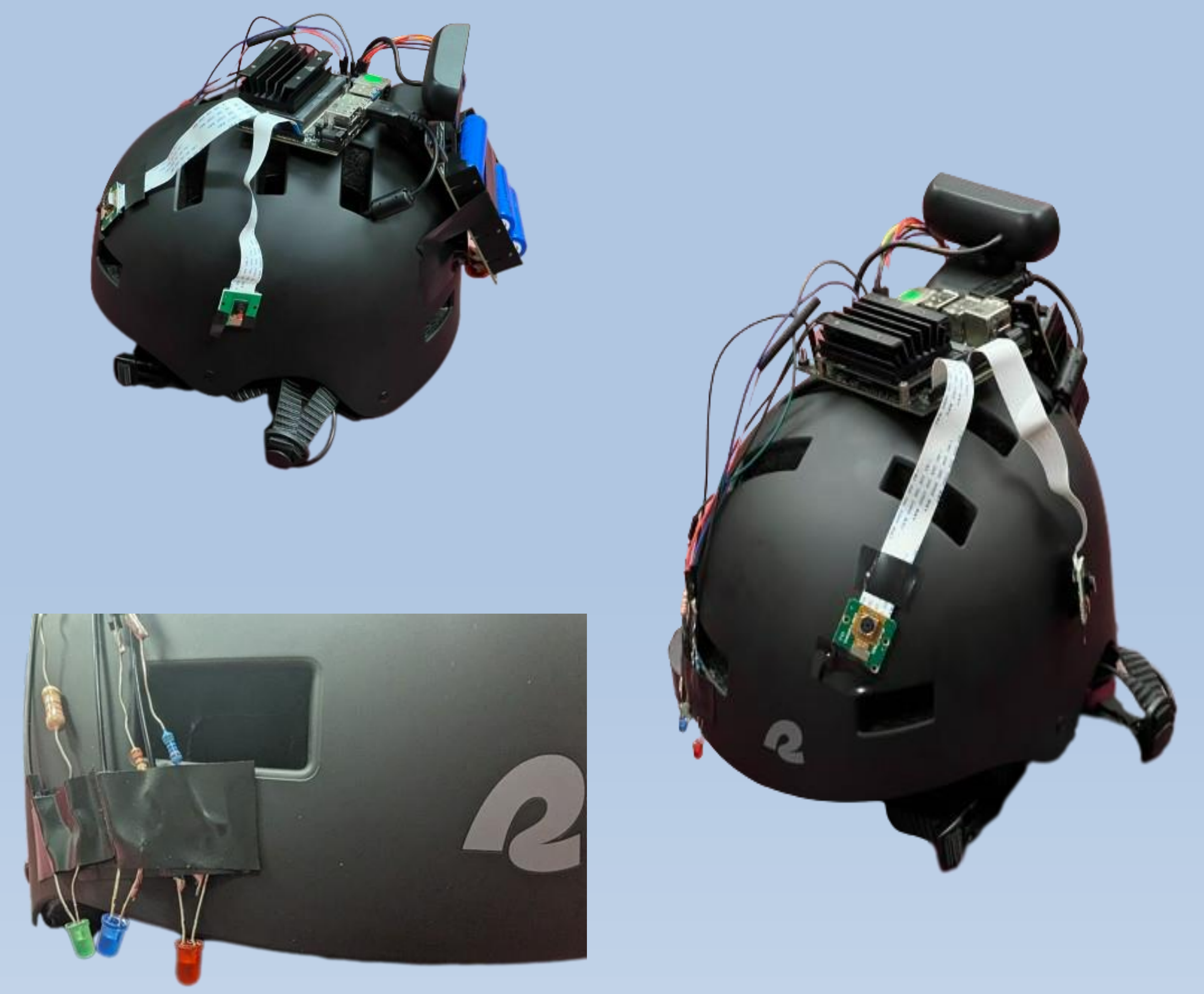
TRAIN / TEST SPLIT

Training Set (80%)	Validation Set (13%)	Testing Set (7%)
1.8k images	294 images	148 images

For our computing board we used an NVIDIA Jetson Nano. We used python for all our programing.



Final Product

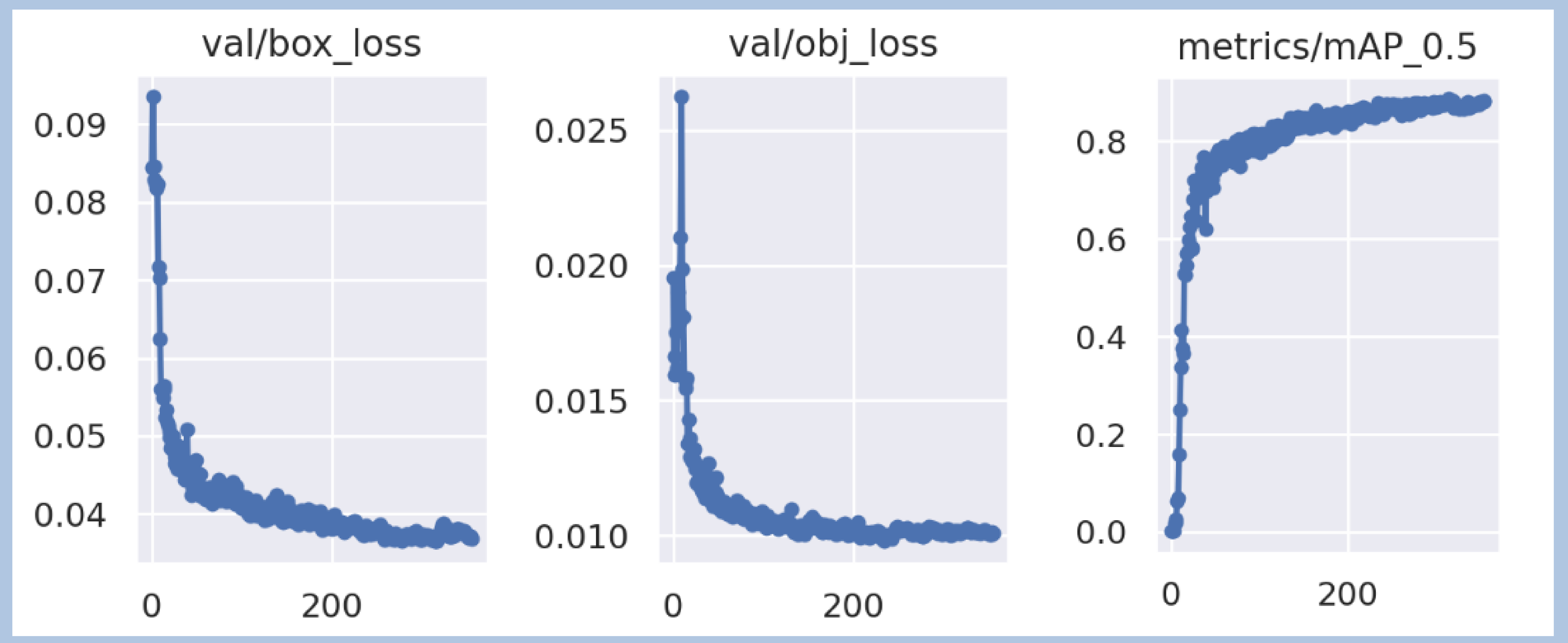


EXPERIMENTAL RESULTS

We used 148 images for our testing set (images not included in our training or validation set.) Our image processing model yielded a precision of 85.0% which is the percent of true positive detections, meaning the chance that a positive detection generated from the model is correct. Our model also has a recall rate of 85.2%, which is the percent of the model being able to correctly detect a desired object that appears in a photo or video. We achieved a mean average percent (mAP) of 88.7%, which combines the previous two metrics into one metric that is the agreed upon metric within the computer vision community to evaluate image processing models.

88.7%	85.0%	85.2%
mAP	precision	recall

The graphs bellows show how various metrics including the mAP changed over the course of adding more validation images.



Bellow are examples of objects detected using our model. Our product takes an average of 100 ms to inference an image.