Traffic Light Detection and Tracking

Team 2A1

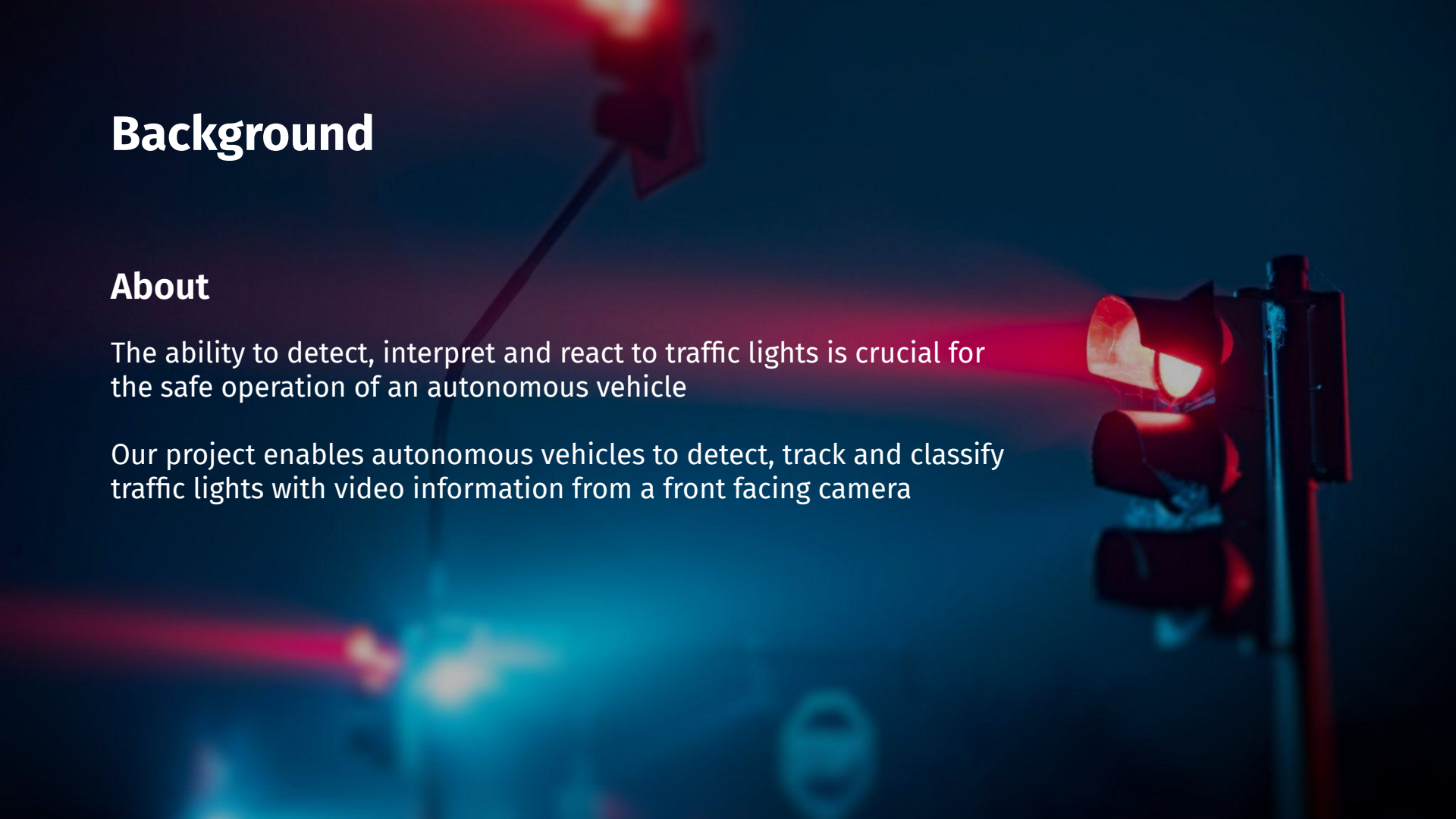
Clayton Gowan, Xiaohu (Max) Huang,
Robert Madriaga, Morgan Roberts, Aaryan Shenoy

Background

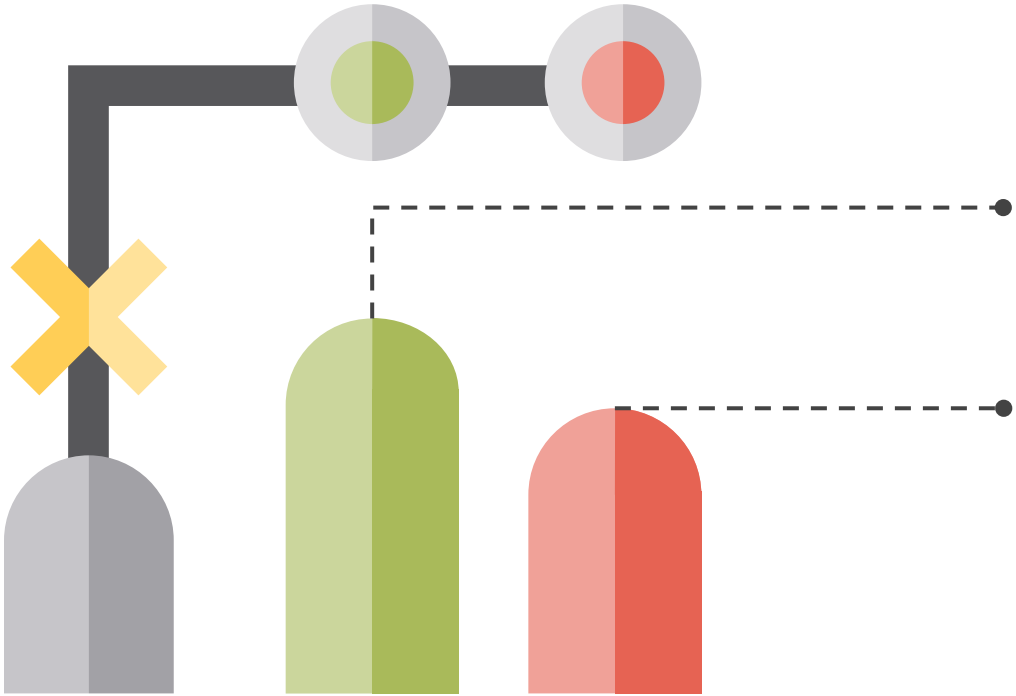
About

The ability to detect, interpret and react to traffic lights is crucial for the safe operation of an autonomous vehicle

Our project enables autonomous vehicles to detect, track and classify traffic lights with video information from a front facing camera



Needs Statement



Need

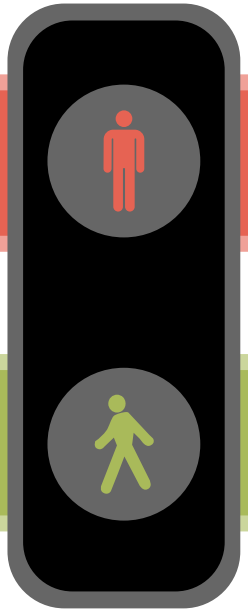
Detect, track, and classify traffic lights while prioritizing speed and accuracy.

Rationale

Self-driving cars necessitate speedy and consistent recognition of traffic lights.

Long response times and low precision are not acceptable when lives are at stake.

Goals and Objectives



01 **Goal**

Develop a speed-focused detection model for the state traffic lights and tracking across frames with ROS2 as an integration framework

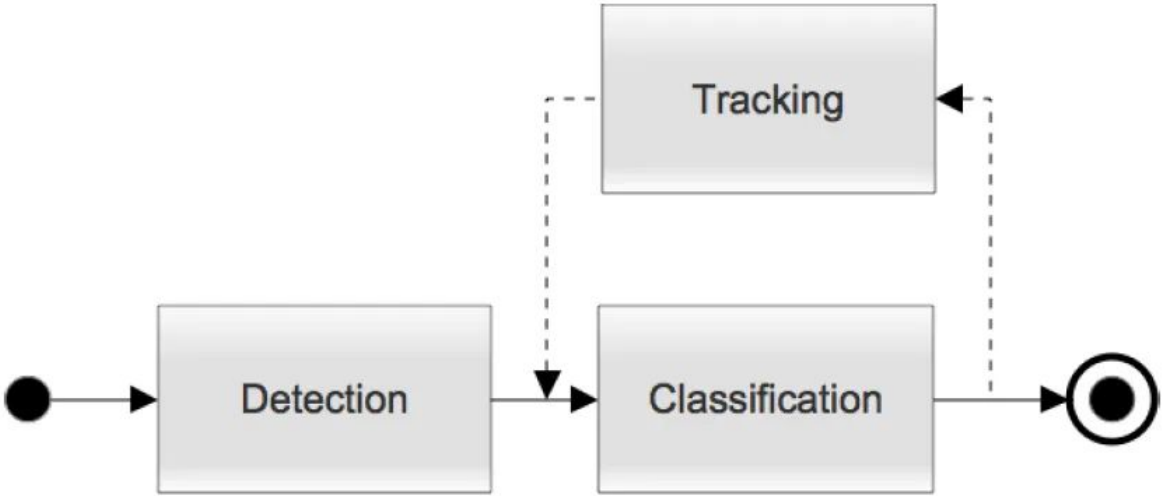
02 **Motivation**

- Fast detection and tracking speeds crucial in autonomous driving applications
- Modularization of detection and tracking components

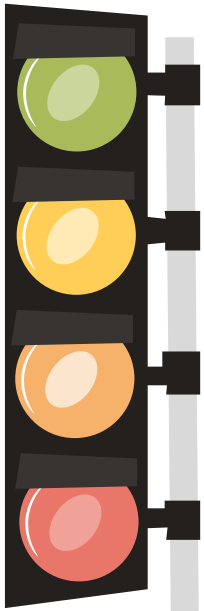
Problem Breakdown

Detection	Classification	Tracking
Scanning for objects of interest in the entire image	Categorizing the states of lights	Predicting the motion of objects

Problem Breakdown (cont.)



Economic Analysis/Budget



Budget

Each Team gets a $500 Budget

Software

Open Source/Free Tier
$0

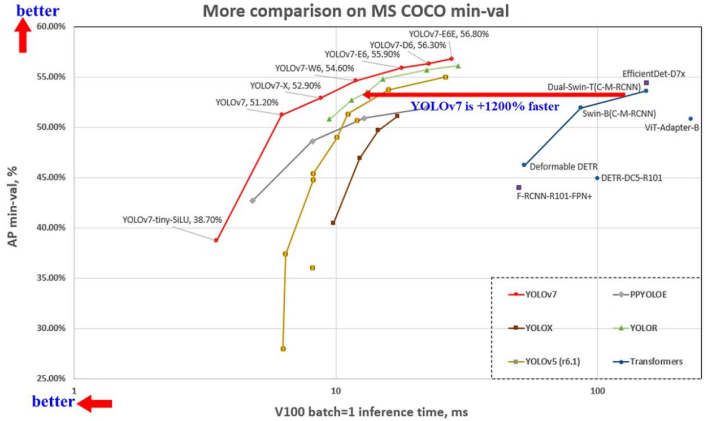
Human Labor

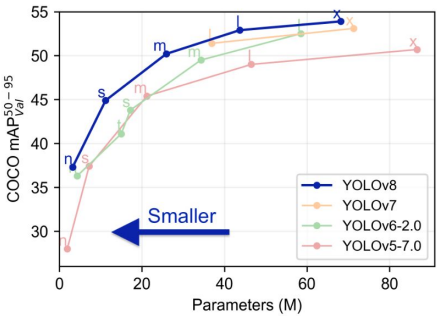
Students
$0

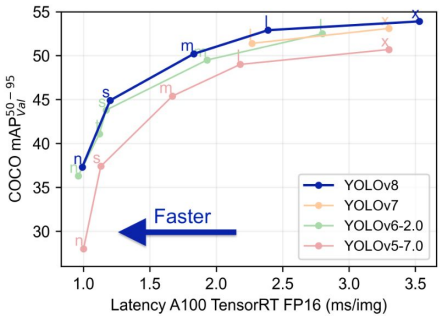
Hardware

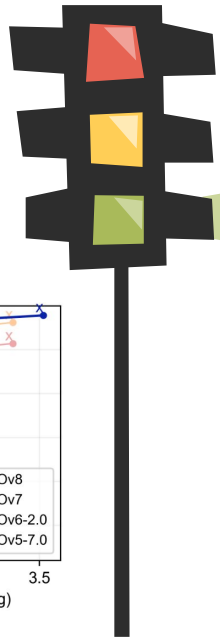
Samsung T7 Shield SSD 1TB
$74.99

Alternative Solutions - Detection Model









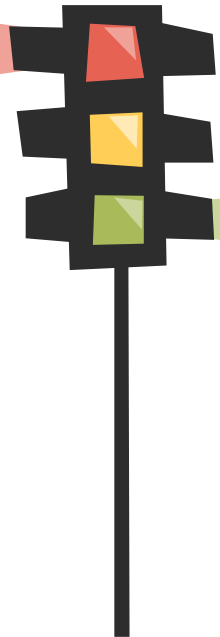
YOLOv8

Findings showed better detection performance and speed compared to older YOLO versions and other alternatives.

Alternative Solutions - Tracker

DeepSORT

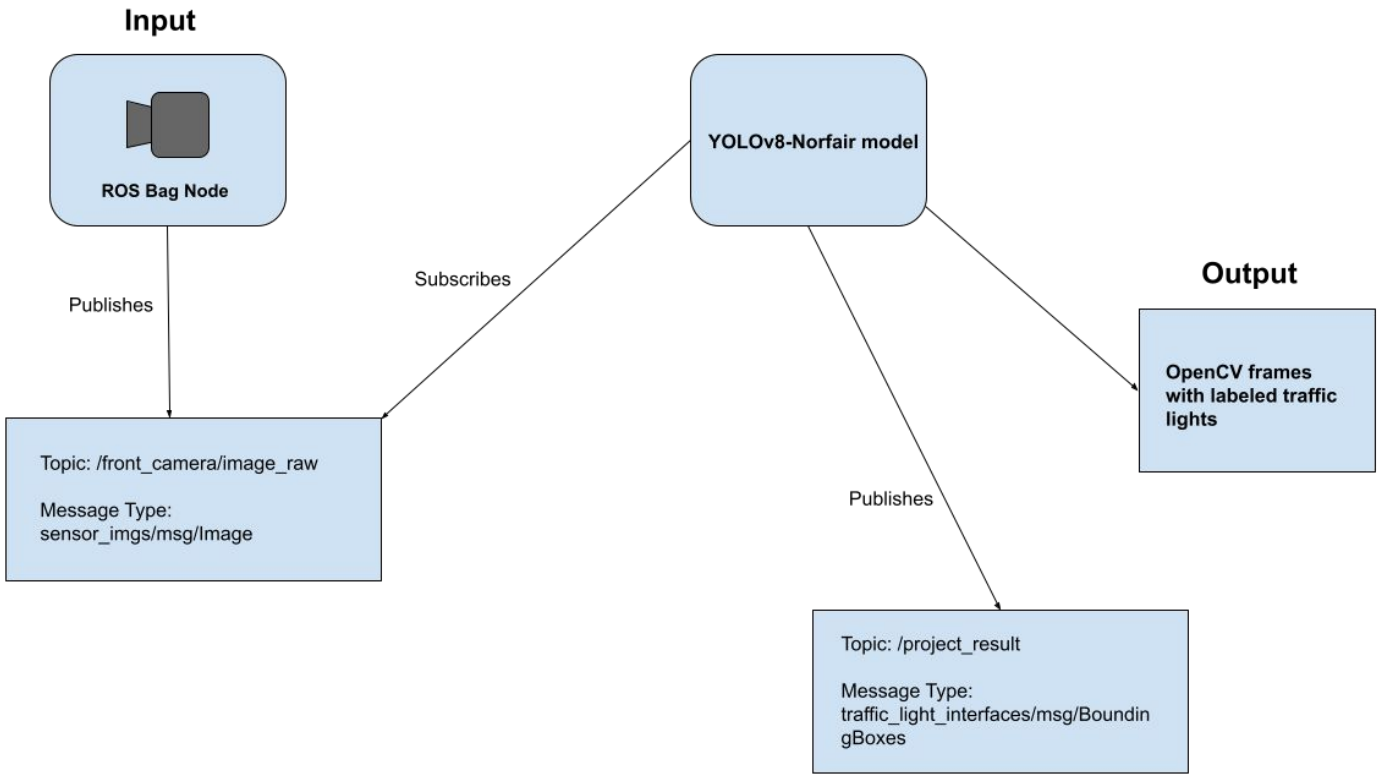
- Utilizes trajectory predictions and deep learning
- Lack of documentation + implementation hurdles



Norfair

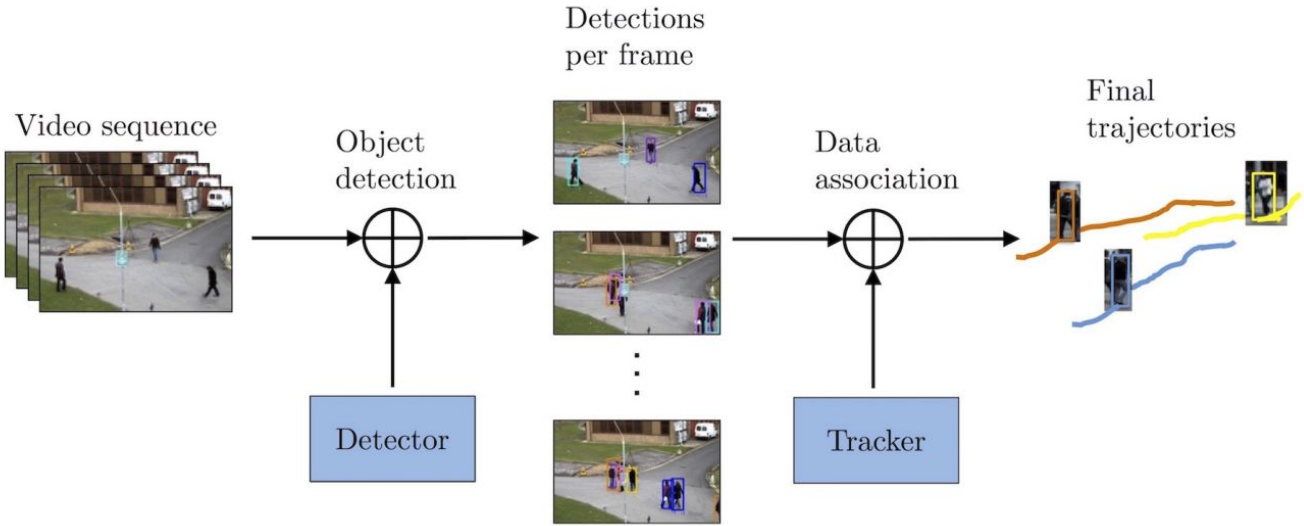
- Open source Python library
- Easy to Implement
- Good Performance

System-Level Description



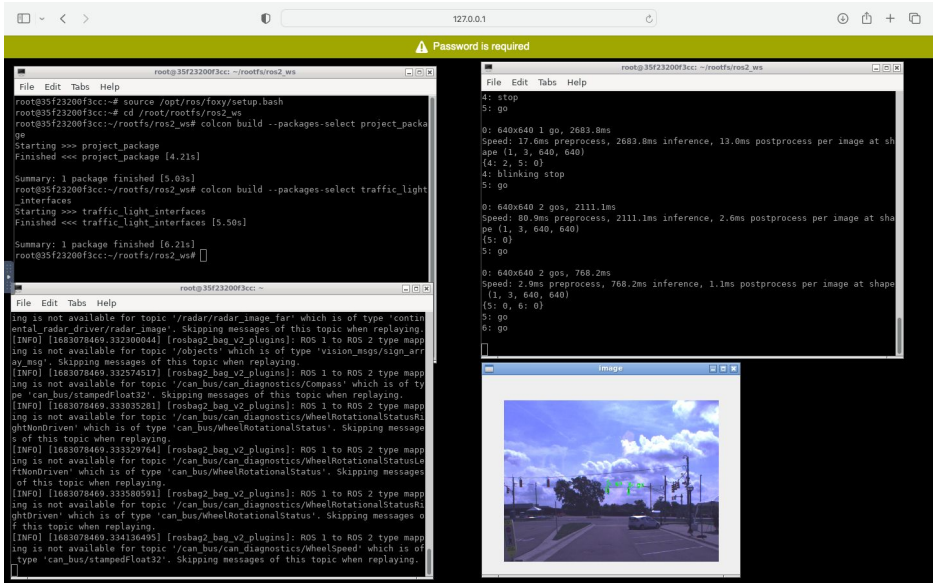
Design Module - YOLOv8 and Norfair

- Runs on each camera frame to produce bounding box detections
- Norfair utilizes YOLOv8 boxes to track traffic lights across frames

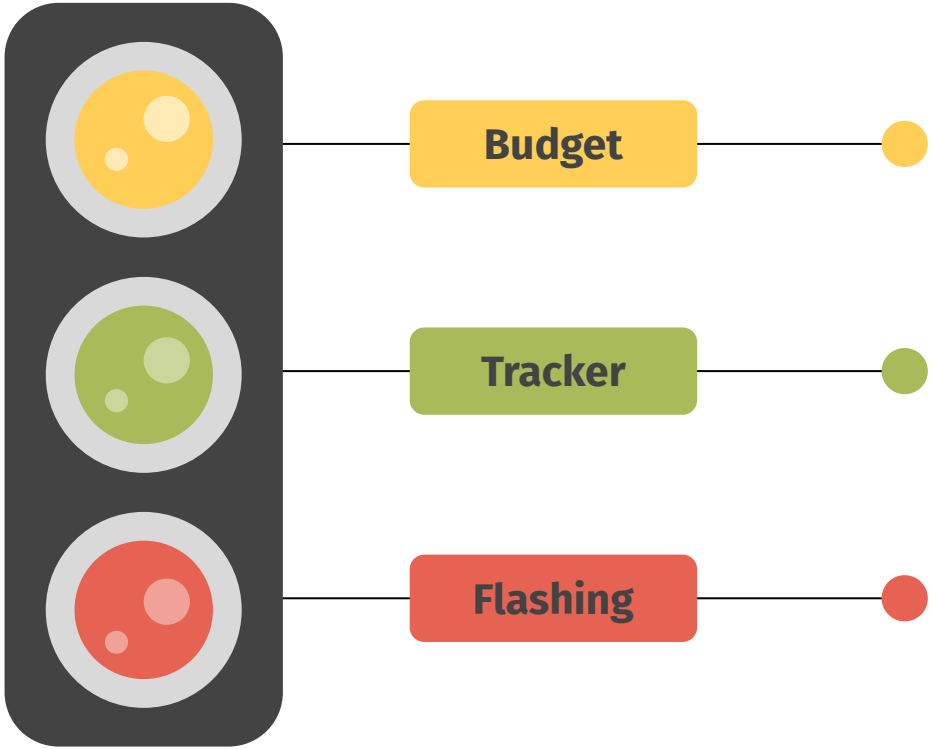


Design Module - Important Algorithms and Docker Hosting

- Flashing light algorithm
- Distance filtering algorithm
- Docker container hosts the project's noVNC GUI



Design Updates

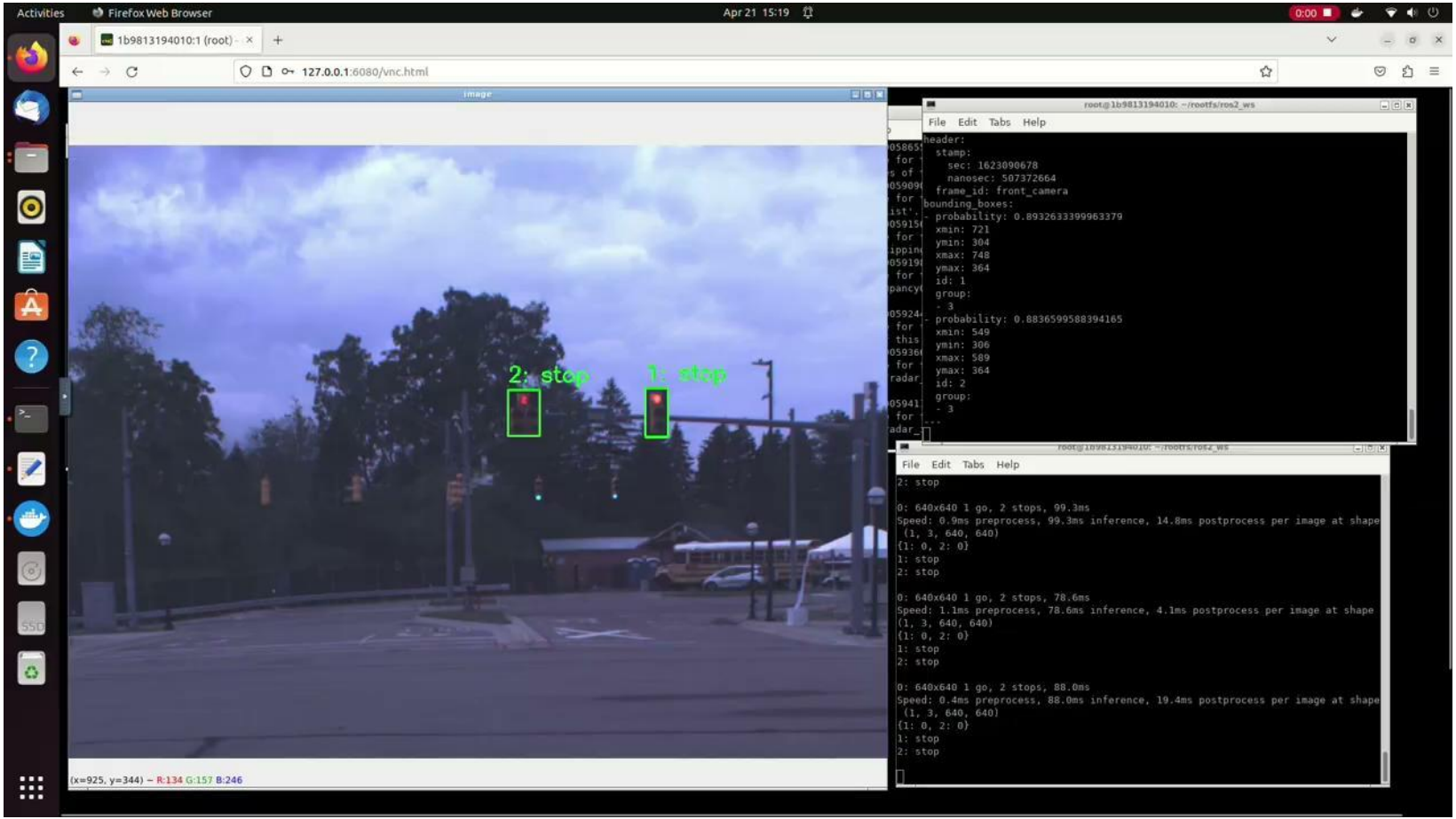


We purchased a hard drive for booting and testing on different machines

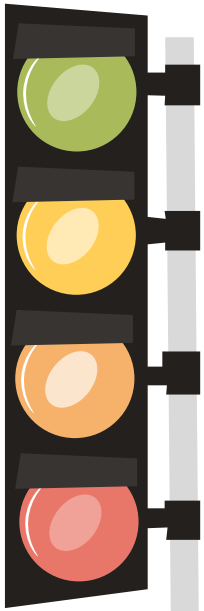
YOLOv8 → DeepSort → Norfair

Our tracker is equipped to recognize flashing lights, which are rather common

Demo


Activities
Firefox Web Browser
Apr 21 15:19
1b9813194010:1 (root)
127.0.0.1:6080/vnc.html
image
2: stop
1: stop
(x=925, y=344) ~ R:134 G:157 B:246
root@1b9813194010: ~/rootfs/ros2_ws
File Edit Tabs Help
header:
stamp:
sec: 1623090678
nanosec: 507372664
frame_id: front_camera
bounding_boxes:
- probability: 0.8932633399963379
xmin: 721
ymin: 304
xmax: 748
ymax: 364
id: 1
group:
- 3
- probability: 0.8836599588394165
xmin: 549
ymin: 306
xmax: 589
ymax: 364
id: 2
group:
- 3
...
File Edit Tabs Help
2: stop
0: 640x640 1 go, 2 stops, 99.3ms
Speed: 0.9ms preprocess, 99.3ms inference, 14.8ms postprocess per image at shape
(1, 3, 640, 640)
{1: 0, 2: 0}
1: stop
2: stop
0: 640x640 1 go, 2 stops, 78.6ms
Speed: 1.1ms preprocess, 78.6ms inference, 4.1ms postprocess per image at shape
(1, 3, 640, 640)
{1: 0, 2: 0}
1: stop
2: stop
0: 640x640 1 go, 2 stops, 88.0ms
Speed: 0.4ms preprocess, 88.0ms inference, 19.4ms postprocess per image at shape
(1, 3, 640, 640)
{1: 0, 2: 0}
1: stop
2: stop

Validation and Testing



Detection Speed

We should be able to run detection quickly (< 100 ms)

Accuracy and Confidence

Our model should have a high degree of accuracy and confidence

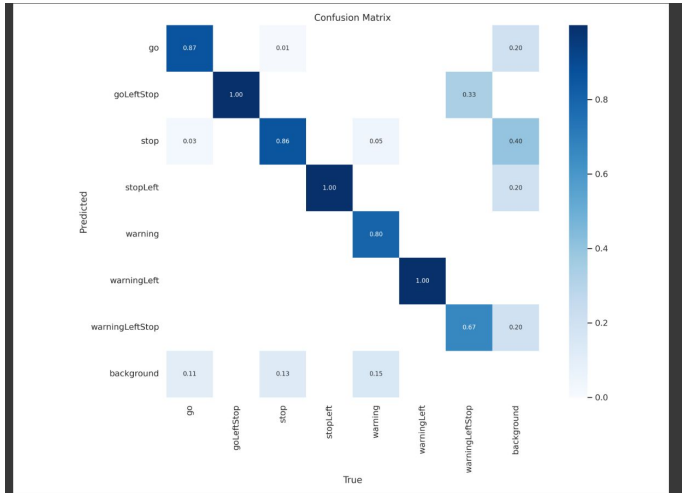
Scenario Variety

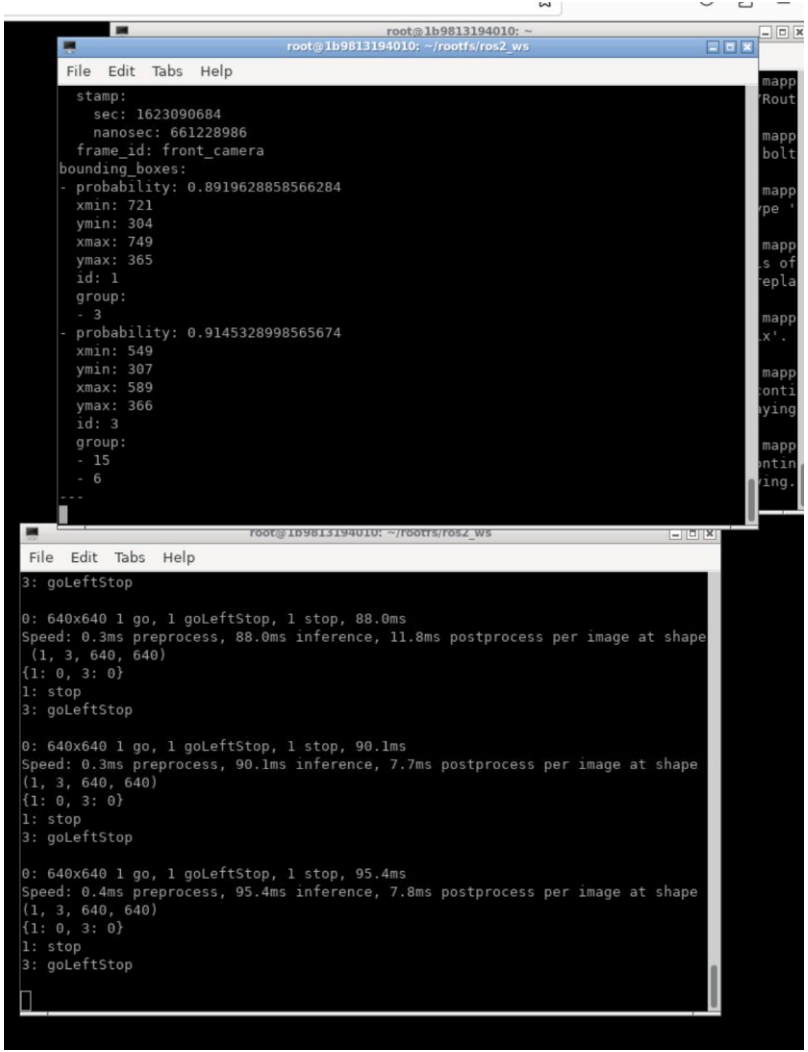
Our code should function as expected in a variety of scenarios

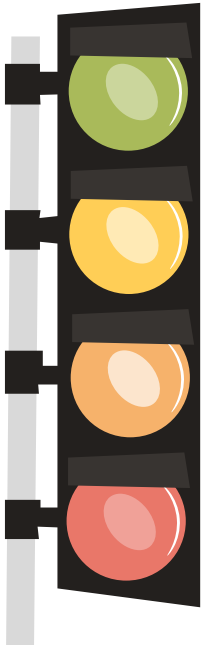
Pipeline

Our ROS pipeline should be correctly setup and publishing messages in the correct format

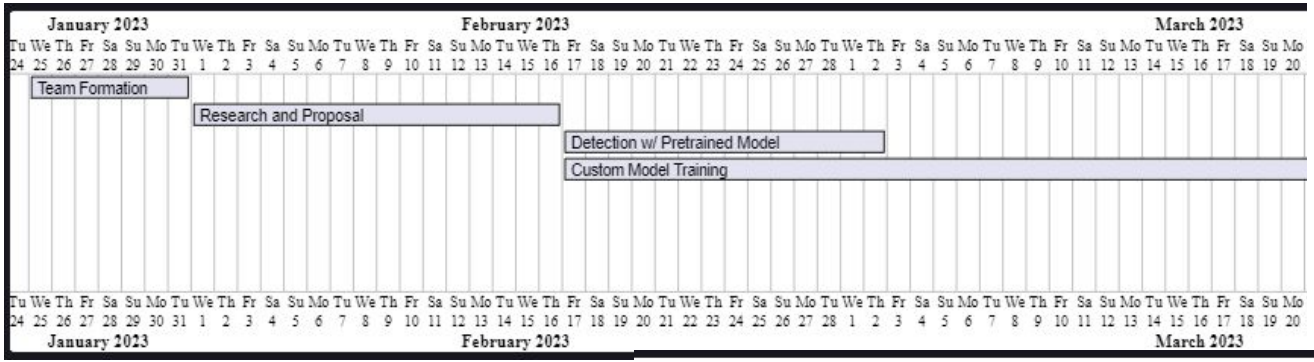
```
      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      50/50       13.8G     0.5782     0.3309     0.7987         46       1280: 100% 46/46 [01:10<00:00,  1.53s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 3/3 [00:09<00:00,  3.21s/it]
                   all         69        176      0.873      0.908      0.969      0.756

50 epochs completed in 2.149 hours.
Optimizer stripped from runs/detect/train/weights/last.pt, 22.7MB
Optimizer stripped from runs/detect/train/weights/best.pt, 22.7MB

Validating runs/detect/train/weights/best.pt...
Ultralytics YOLOv8.0.53 🚀 Python-3.9.16 torch-1.13.1+cu116 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 11128293 parameters, 0 gradients, 28.5 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 3/3 [00:08<00:00,  2.91s/it]
                   all         69        176      0.915      0.884      0.965      0.755
                    go         69         75          1      0.775      0.958      0.661
            goLeftStop         69          4      0.914          1      0.995      0.895
                  stop         69         71      0.944      0.718       0.93      0.593
              stopLeft         69          1      0.619          1      0.995      0.895
               warning         69         20      0.996        0.8      0.885       0.73
           warningLeft         69          2      0.933          1      0.995      0.798
       warningLeftStop         69          3          1      0.895      0.995      0.708
Speed: 6.2ms preprocess, 28.2ms inference, 0.0ms loss, 2.0ms postprocess per image
Results saved to runs/detect/train
```

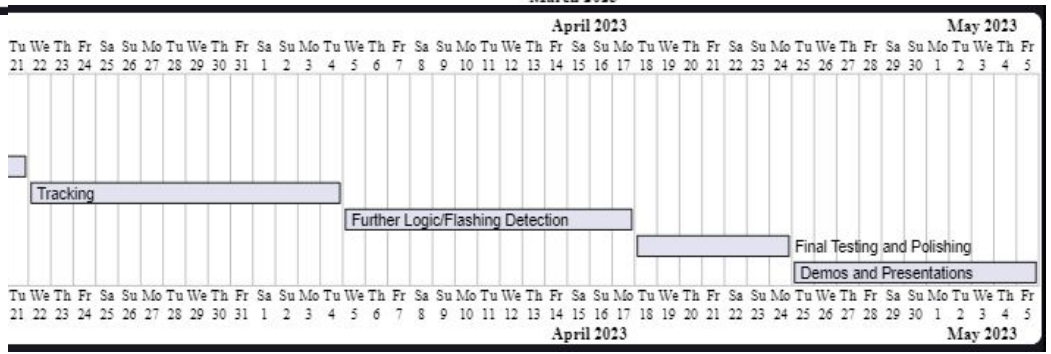






Schedule of Tasks





Social, Political, and Ethical Concerns

Social

Autonomous vehicles may displace transportation workers
Must demonstrate safety to be trusted for use by the public and become a viable product

Political

Regulatory bodies may have doubts about safety and reliability of self driving vehicles

Ethical

Need to properly communicate capabilities and limitations to regulators and the public
Ensure that the overall product is safe for use before delivering

Project Management and Teamwork

Docker/Git

The team worked in the same docker container
Git was used for synchronization

Tri-Weekly Meetings

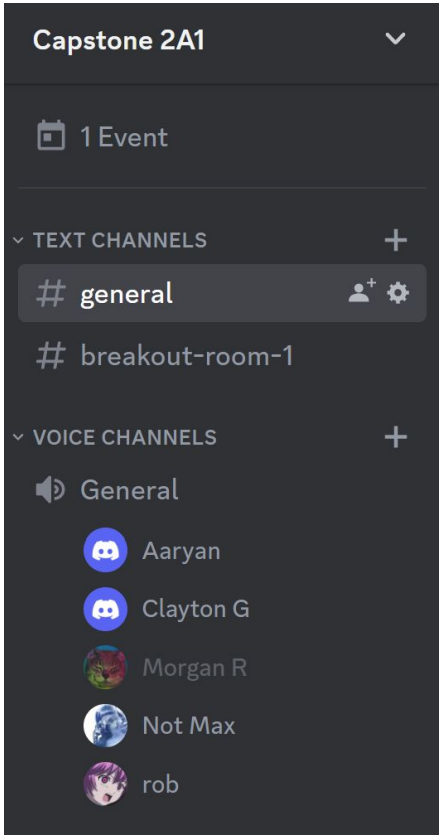
in-person on Tuesday and Thursday
online on Sunday

Communication

Discord was our primary channel

Distribution of Labor

Team was split into two parties to work on the model and docker/ROS



Health, Safety, and Environmental Analysis

Health Benefits

Autonomous vehicles can improve quality of life and increase accessibility

Environmental Benefits

Optimal navigation leads to a reduction of emissions

Safety Concerns

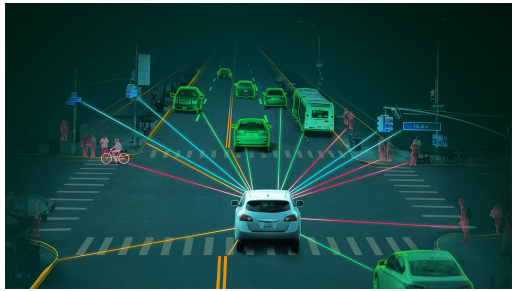
- Slightest of errors can result in catastrophic system failure
- Failure results in loss of life and property

Safety Benefits

Under ideal operating conditions, cars can avoid danger and drive better than humans

Privacy Concerns

Cars are rolling surveillance devices



Manufacturability and Sustainability Hardware

Limitations

Performance of the model is limited by frame rate consistency

Design

Designed for a high resolution camera at 60fps

Requirements

Hardware handling the computations needs to be higher end

Manufacturability and Sustainability Software

Limitations

Updates to software can cause compatibility errors as dependencies change/ features deprecated

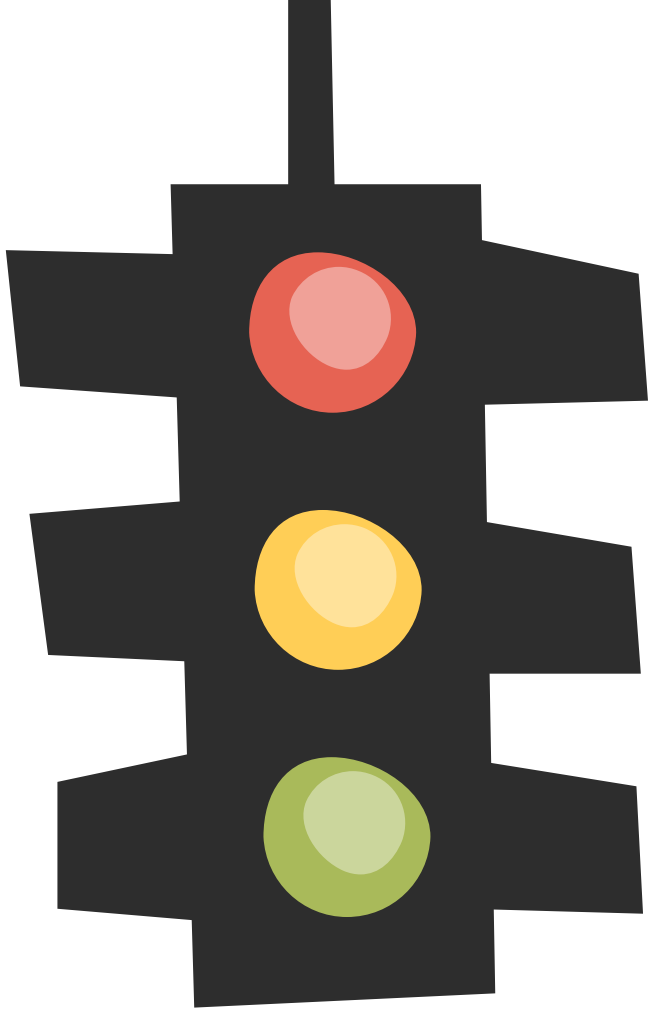
Design

Made to run in an ROS2 environment with Python3

Requirements

Versions of all software used need the most recent of 5/2/23

ROS 2™



Questions?