B4: Ride-AR

Team Members:
- Fayyaz Zaidi
- Chad Taylor
- Ethan Wang

Use-Case - The Problem

- Over 40,000 bicycle accidents and 800 deaths occurred in the US in 2019
- A large number of bicycle accidents occur because they fail to yield or see a bicyclist and end up colliding with them.

Solution Approach

- Improve biker's situational awareness using sensors
- Use visual warnings to alert bicyclists

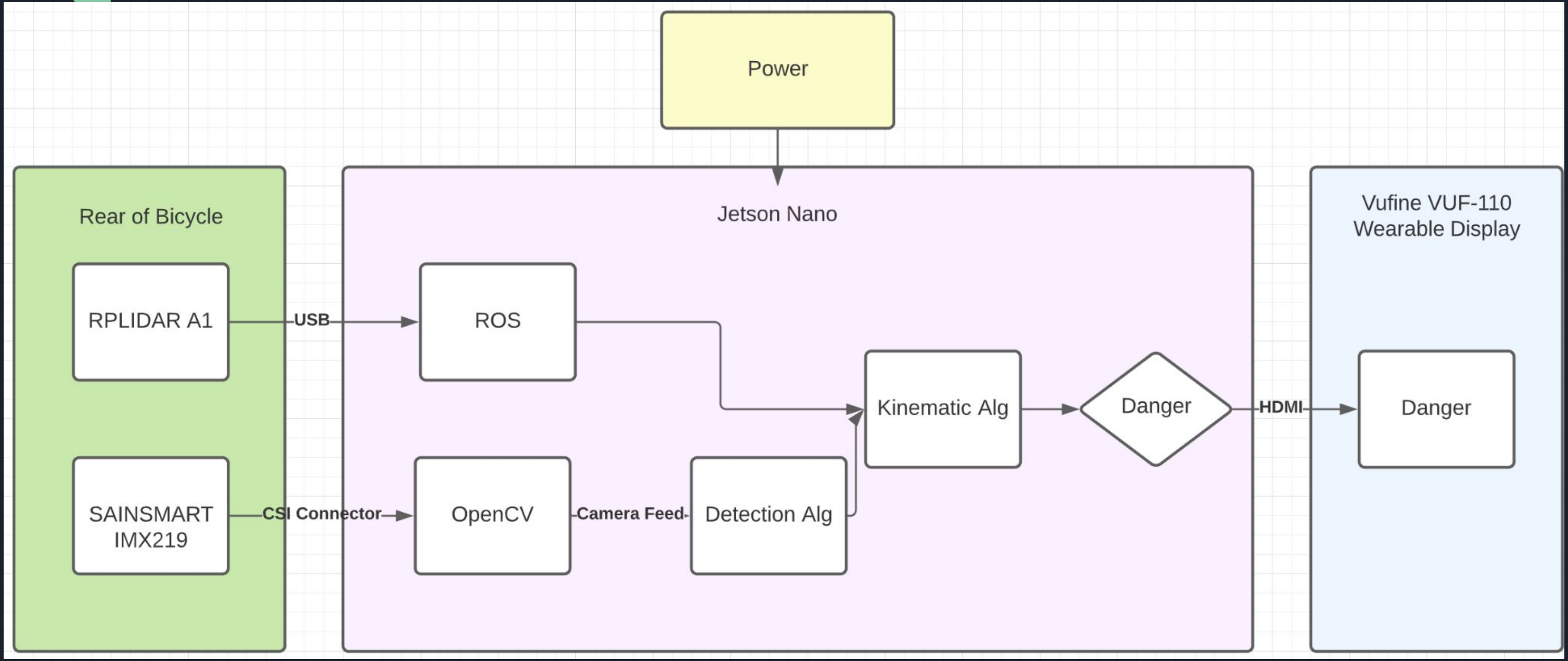
Requirements

- Camera and LIDAR > 120 degree field of view
- LIDAR detect car within 12m*
    - Distance to car within 1m
- Object recognition detect car within 60m
    - <10% error rate
- Camera detection up to 60 m
- Display danger on HUD glasses

Requirements

- Calculate speed of car within 5mph
- Process inputs and alert user within 0.3 seconds of detection
- Battery life of 6 hours
- <3lbs
- <0.5ft^3

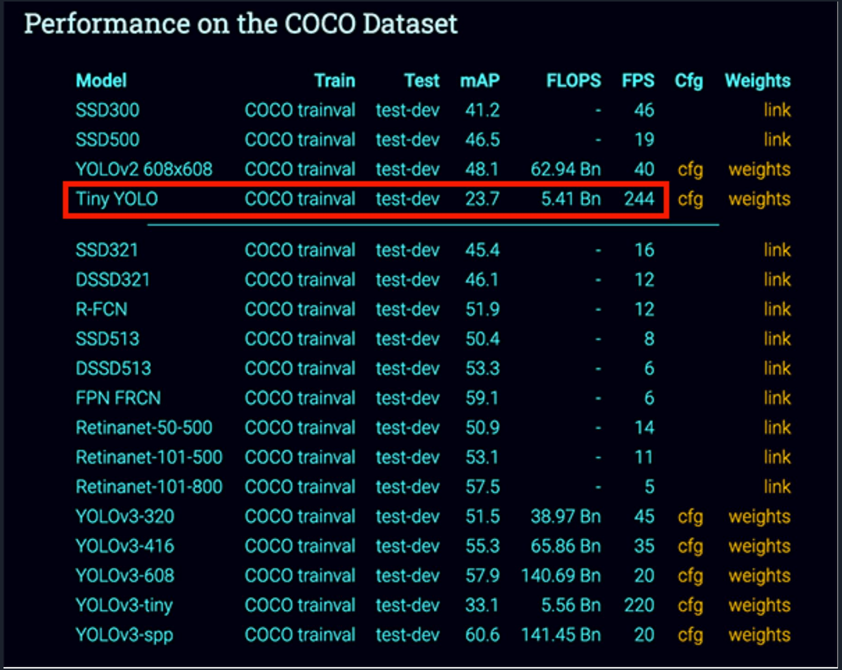
System Specifications



System Specifications

YOLO and SSD:

- Tiny-YOLO is much lighter and faster than YOLOv3 but comes at a cost
- SSD and YOLO both used for real-time object detection
- YOLO is more popular than SSD because of how fast and accurate it is
- YOLO uses darknet while SSD can be implemented with TensorRT package



Performance on the COCO Dataset

Model	Train	Test	mAP	FLOPS	FPS	Cfg	Weights
SSD300	COCO trainval	test-dev	41.2	-	46		link
SSD500	COCO trainval	test-dev	46.5	-	19		link
YOLOv2 608x608	COCO trainval	test-dev	48.1	62.94 Bn	40	cfg	weights
Tiny YOLO	COCO trainval	test-dev	23.7	5.41 Bn	244	cfg	weights
SSD321	COCO trainval	test-dev	45.4	-	16		link
DSSD321	COCO trainval	test-dev	46.1	-	12		link
R-FCN	COCO trainval	test-dev	51.9	-	12		link
SSD513	COCO trainval	test-dev	50.4	-	8		link
DSSD513	COCO trainval	test-dev	53.3	-	6		link
FPN FRCN	COCO trainval	test-dev	59.1	-	6		link
Retinanet-50-500	COCO trainval	test-dev	50.9	-	14		link
Retinanet-101-500	COCO trainval	test-dev	53.1	-	11		link
Retinanet-101-800	COCO trainval	test-dev	57.5	-	5		link
YOLOv3-320	COCO trainval	test-dev	51.5	38.97 Bn	45	cfg	weights
YOLOv3-416	COCO trainval	test-dev	55.3	65.86 Bn	35	cfg	weights
YOLOv3-608	COCO trainval	test-dev	57.9	140.69 Bn	20	cfg	weights
YOLOv3-tiny	COCO trainval	test-dev	33.1	5.56 Bn	220	cfg	weights
YOLOv3-spp	COCO trainval	test-dev	60.6	141.45 Bn	20	cfg	weights

Implementation Plan

- Lidar, Jetson, battery pack, HUD Glasses and camera are being ordered.
- We will be downloading the YOLO and SSD real-time detection algorithms and experiment with both
- Cooling fan may be bought depending on performance of Jetson Nano

Testing, Verification, Metrics

Use Case 1: Car Recognition

- Enter empty space and perform numerous trials placing camera in front of a vehicle at a number of different orientations
    - Measure both the percentage of accurate car detection as well as the time needed to recognize the vehicle. Also change visibility to see how camera reacts

Use Case 2: Distance Detection

- Move car or like object near lidar to see if accurately collecting data 12m maximum range with a margin of error of 1m through console output

Use Case 3: Danger Detection

- Two step approach. Either use preset speed and timings and use basic kinematic to determine speed from different distances from bike or use large space drive car at slower speeds and read calculations from sensor

Testing, Verification, Metrics

Use Case 4: Latency Testing

- Move objects closer and farther from the sensors and measure the time it takes to display the visuals on the glasses

Use Case 5: Object Detection

- We will experiment with the YOLO and SSD object detection algorithms. We will judge the detection algorithms based on their performances such as accuracy and detection speed.

Project Management

Fayyaz:

1. Integrate camera and lidar together with jetson such that they can speak to each other
2. Design and implement circuit to integrate glasses display when the algorithm determines danger

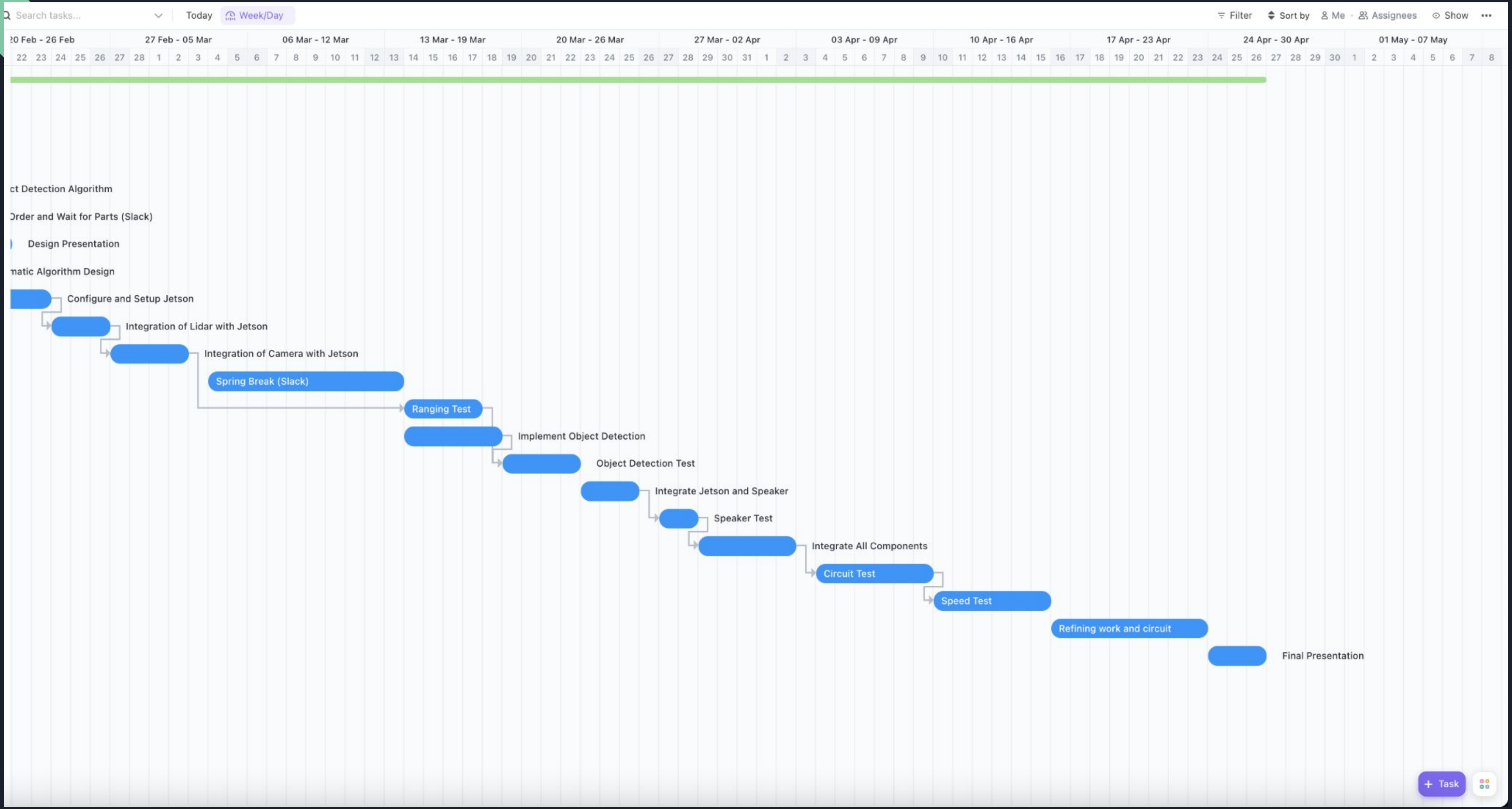
Ethan:

3. Design effective module to attach to bike
4. Configure lidar sensor to ensure that we read precise measurements

Chad:

5. Install and implement a computer vision algorithm to detect cars
6. Create algorithm that uses kinematics, lidar output, and object detection to determine surrounding and potential danger

Schedule


Search tasks...
Today
Week/Day
Filter
Sort by
Me
Assignees
Show
20 Feb - 26 Feb
27 Feb - 05 Mar
06 Mar - 12 Mar
13 Mar - 19 Mar
20 Mar - 26 Mar
27 Mar - 02 Apr
03 Apr - 09 Apr
10 Apr - 16 Apr
17 Apr - 23 Apr
24 Apr - 30 Apr
01 May - 07 May
ct Detection Algorithm
Order and Wait for Parts (Slack)
Design Presentation
matic Algorithm Design
Configure and Setup Jetson
Integration of Lidar with Jetson
Integration of Camera with Jetson
Spring Break (Slack)
Ranging Test
Implement Object Detection
Object Detection Test
Integrate Jetson and Speaker
Speaker Test
Integrate All Components
Circuit Test
Speed Test
Refining work and circuit
Final Presentation
Task