



IOT FINAL PRESENTATION IMAGE TRAFFIC IDENTIFICATION USING DEEP LEARNING

Kevin Huang, Matthew Caro, Ankith Nagabandi

RECAP

- **CARLA:** simulate and model driving situations with a variety of factors/features
 - Collect this data and store it as a centralized dataset, perform classification analysis on it
- **Machine Learning Models:** develop a model to test on our data
 - Evaluate models for a variety of factors:
 - Speed
 - Accuracy/F1-score
 - ETC...

FINAL OBJECTIVES

- Create a CARLA dataset
- Build models to identify different objects
- Correctly classify different objects into classes

DATASET I

- Classification
 - Bike
 - Motorbike
 - Traffic_light
 - Traffic_sign
 - Vehicle
- Yolo labels
- Split data:
 - Train samples: 2556 (58.9%)
 - Test samples : 1839 (41.8%)

LABELING

Original



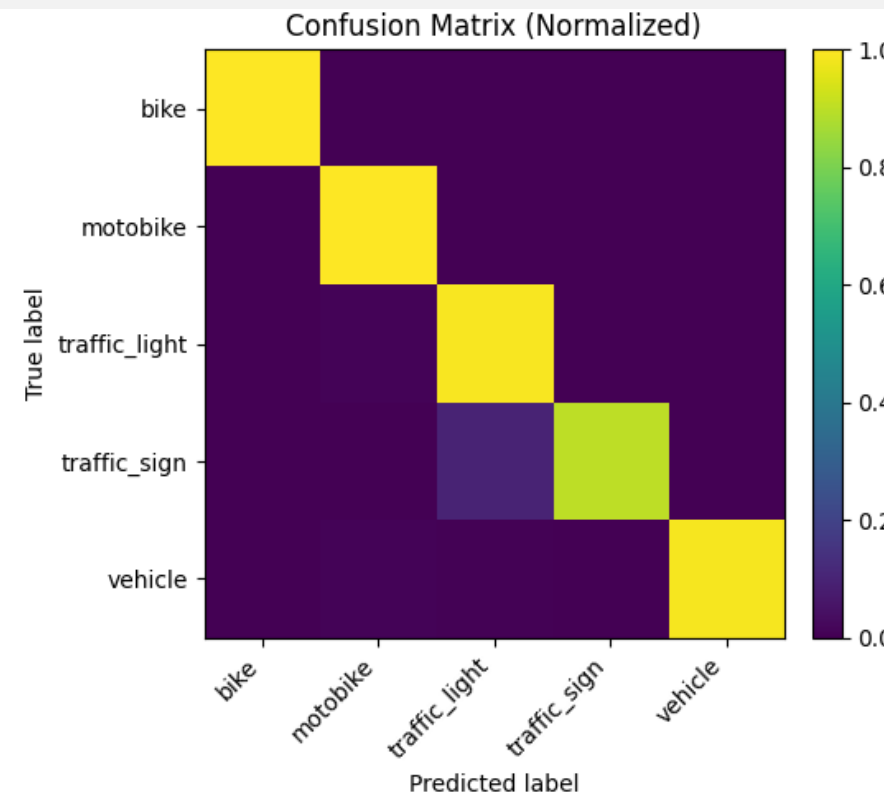
With Boxes



CNN I FOR OBJECT CLASSIFICATION

	precision	recall	f1-score	support
bike	1.000	1.000	1.000	30
motobike	0.385	1.000	0.556	10
traffic_light	0.999	0.991	0.995	1586
traffic_sign	1.000	0.900	0.947	10
vehicle	0.995	0.985	0.990	203
accuracy			0.990	1839
macro avg	0.876	0.975	0.898	1839
weighted avg	0.995	0.990	0.992	1839

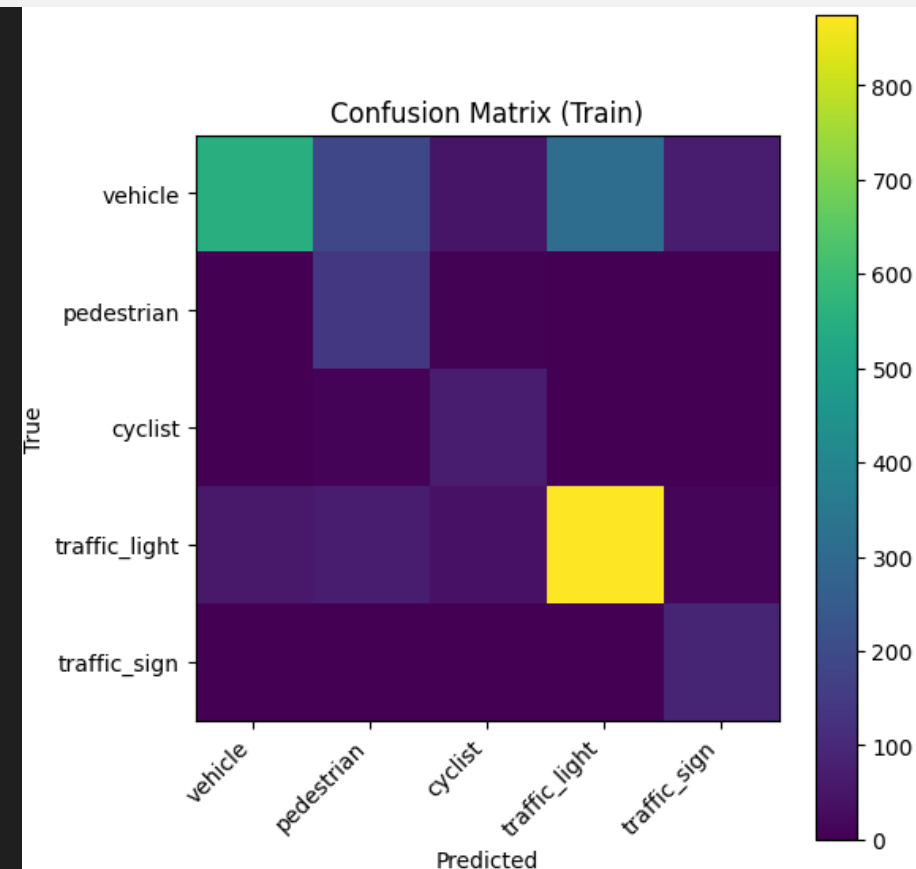
Sample 00: true=vehicle, pred=vehicle
Sample 01: true=vehicle, pred=vehicle
Sample 02: true=bike, pred=bike
Sample 03: true=vehicle, pred=vehicle
Sample 04: true=traffic_light, pred=traffic_light
Sample 05: true=traffic_light, pred=traffic_light
Sample 06: true=traffic_light, pred=traffic_light
Sample 07: true=traffic_light, pred=traffic_light
Sample 08: true=bike, pred=bike



DATASET GNN

	precision	recall	f1-score	support
vehicle	0.894	0.470	0.616	1172
pedestrian	0.351	0.940	0.511	151
cyclist	0.401	0.827	0.540	81
traffic_light	0.734	0.827	0.778	1057
traffic_sign	0.517	0.968	0.674	95
accuracy			0.675	2556
macro avg	0.580	0.807	0.624	2556
weighted avg	0.767	0.675	0.677	2556

	precision	recall	f1-score	support
vehicle	0.833	0.172	0.286	203
pedestrian	0.179	0.900	0.298	30
cyclist	0.250	0.600	0.353	10
traffic_light	0.905	0.915	0.910	1586
traffic_sign	0.500	0.900	0.643	10
accuracy			0.831	1839
macro avg	0.533	0.697	0.498	1839
weighted avg	0.879	0.831	0.826	1839



GNN CONTINUE

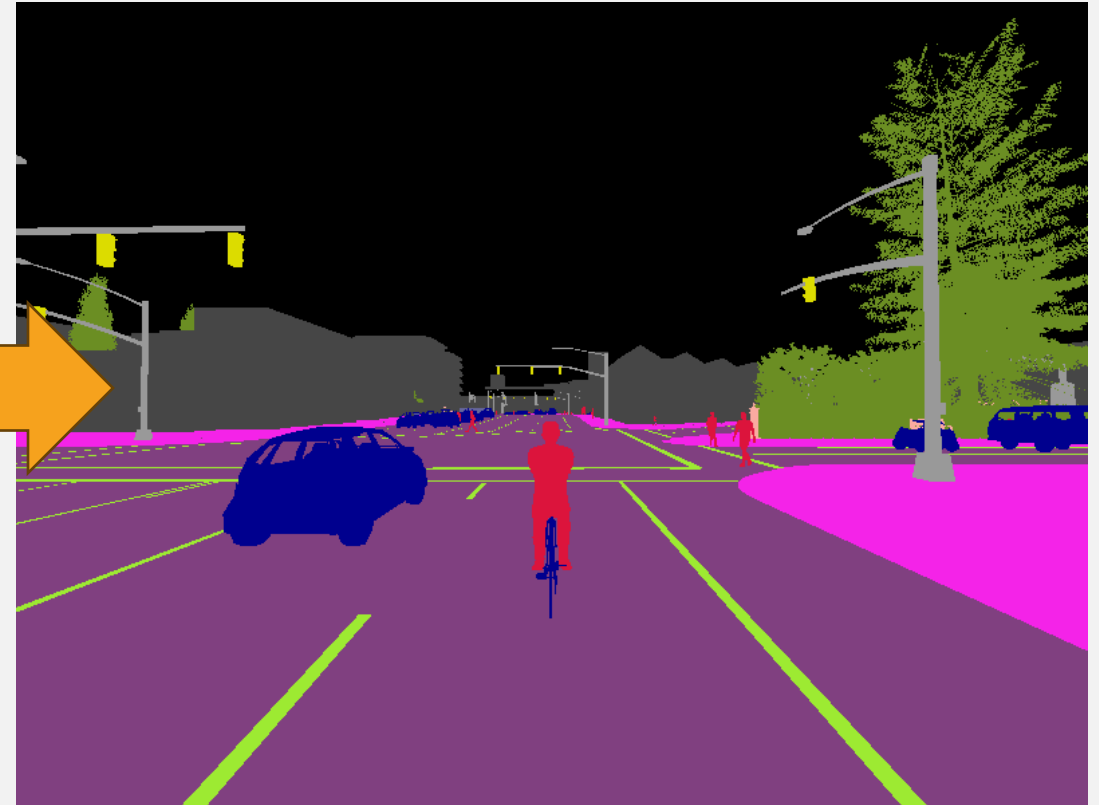
```
Graph idx: 0
Image path: carla-object-detection-dataset\images\train\Town01_001440.png
Num nodes: 1
True labels:      [4]
Predicted labels: [4]
Recommended action: GO
-----
Graph idx: 546
Image path: carla-object-detection-dataset\images\train\Town03_017580.png
Num nodes: 5
True labels:      [1, 2, 0, 3, 0]
Predicted labels: [2, 2, 2, 2, 2]
Recommended action: SLOW
-----
Graph idx: 134
Image path: carla-object-detection-dataset\images\train\Town01_012480.png
Num nodes: 1
True labels:      [3]
Predicted labels: [3]
Recommended action: GO
-----
Graph idx: 124
Image path: carla-object-detection-dataset\images\train\Town01_011820.png
Num nodes: 2
True labels:      [0, 3]
...
Num nodes: 1
True labels:      [0]
Predicted labels: [0]
Recommended action: GO
```


DATASET 2

- Split data:
 - Total samples: 109755
 - Train size: 76828 (70%)
 - Test size: 32927 (30%)

Classification	RGB codes
Traffic Sign	(220, 220, 0)
Building	(70, 70, 70)
Fence	(190, 153, 153)
Other	(250, 170, 160)
Pedestrian	(220, 20, 60)
Pole	(153, 153, 153)
Road Line	(157, 234, 50)
Road	(128, 64, 128)
Sidewalk	(244, 35, 232)
Vegetation	(107, 142, 35)
Car	(0, 0, 142)
Wall	(102, 102, 156)
Unlabeled	(0, 0, 0)

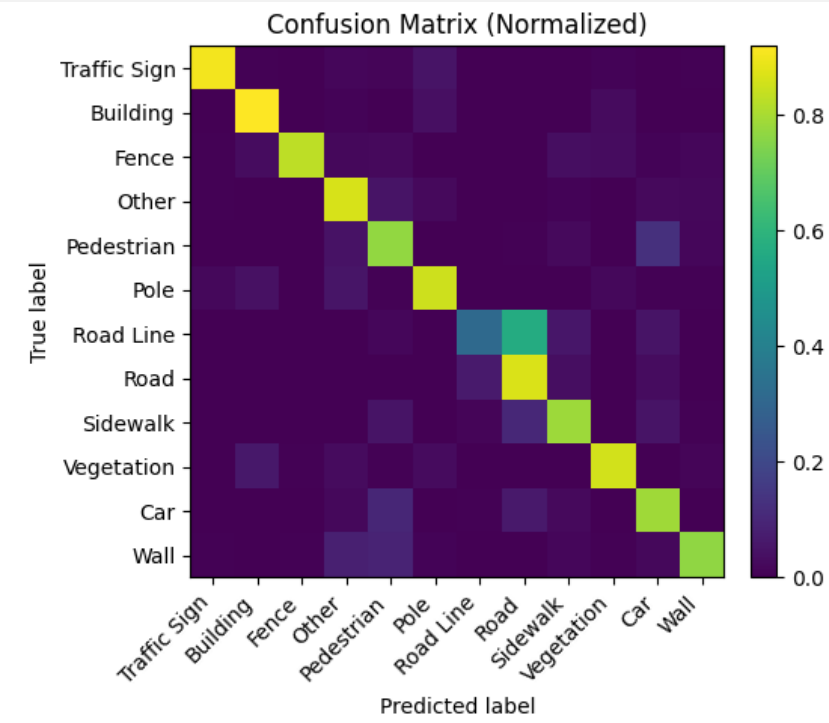
LABELING



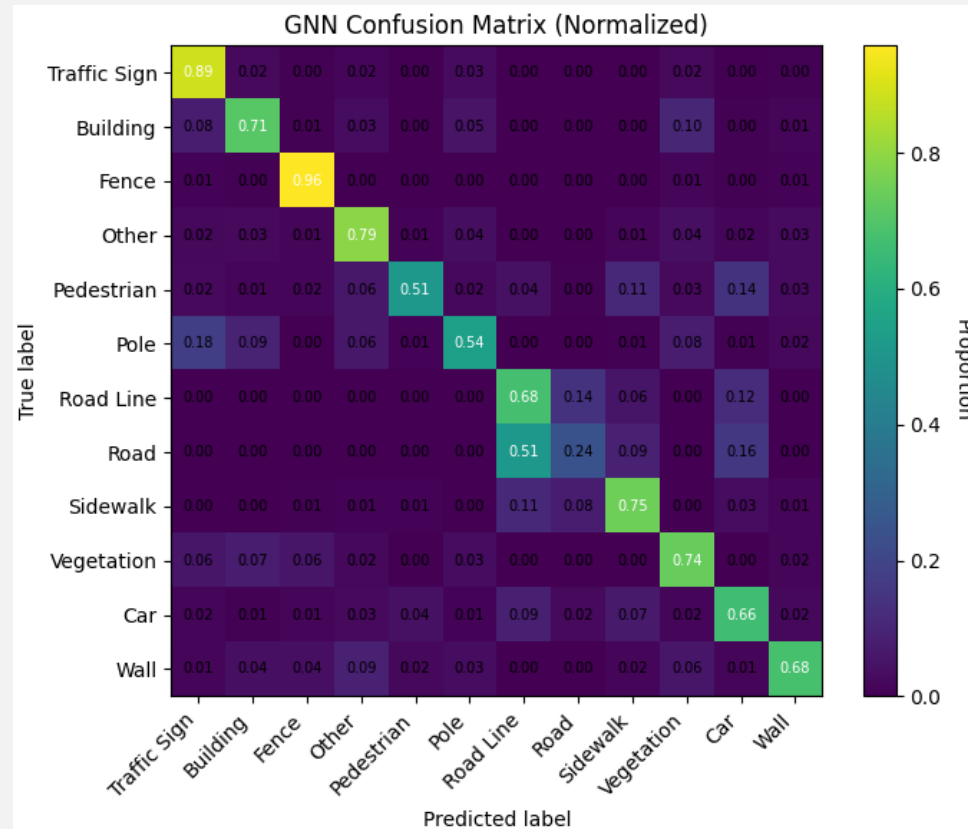
CNN 2 FOR OBJECT CLASSIFICATION

	precision	recall	f1-score	support
Traffic Sign	0.941	0.905	0.922	2015
Building	0.896	0.920	0.908	3207
Fence	0.932	0.826	0.876	800
Other	0.760	0.864	0.809	2656
Pedestrian	0.718	0.772	0.744	3117
Pole	0.870	0.850	0.860	3161
Road Line	0.791	0.315	0.450	3207
Road	0.540	0.866	0.665	3206
Sidewalk	0.828	0.784	0.805	3207
Vegetation	0.921	0.858	0.888	2786
Car	0.728	0.789	0.757	3209
Wall	0.901	0.767	0.829	2356
accuracy			0.785	32927
macro avg	0.819	0.793	0.793	32927
weighted avg	0.804	0.785	0.780	32927

```
Sample 00: true=Traffic Sign, pred=Traffic Sign
Sample 01: true=Car, pred=Road Line
Sample 02: true=Pedestrian, pred=Pedestrian
Sample 03: true=Other, pred=Other
Sample 04: true=Pedestrian, pred=Pedestrian
Sample 05: true=Road, pred=Road
Sample 06: true=Vegetation, pred=Vegetation
Sample 07: true=Building, pred=Building
Sample 08: true=Other, pred=Other
```

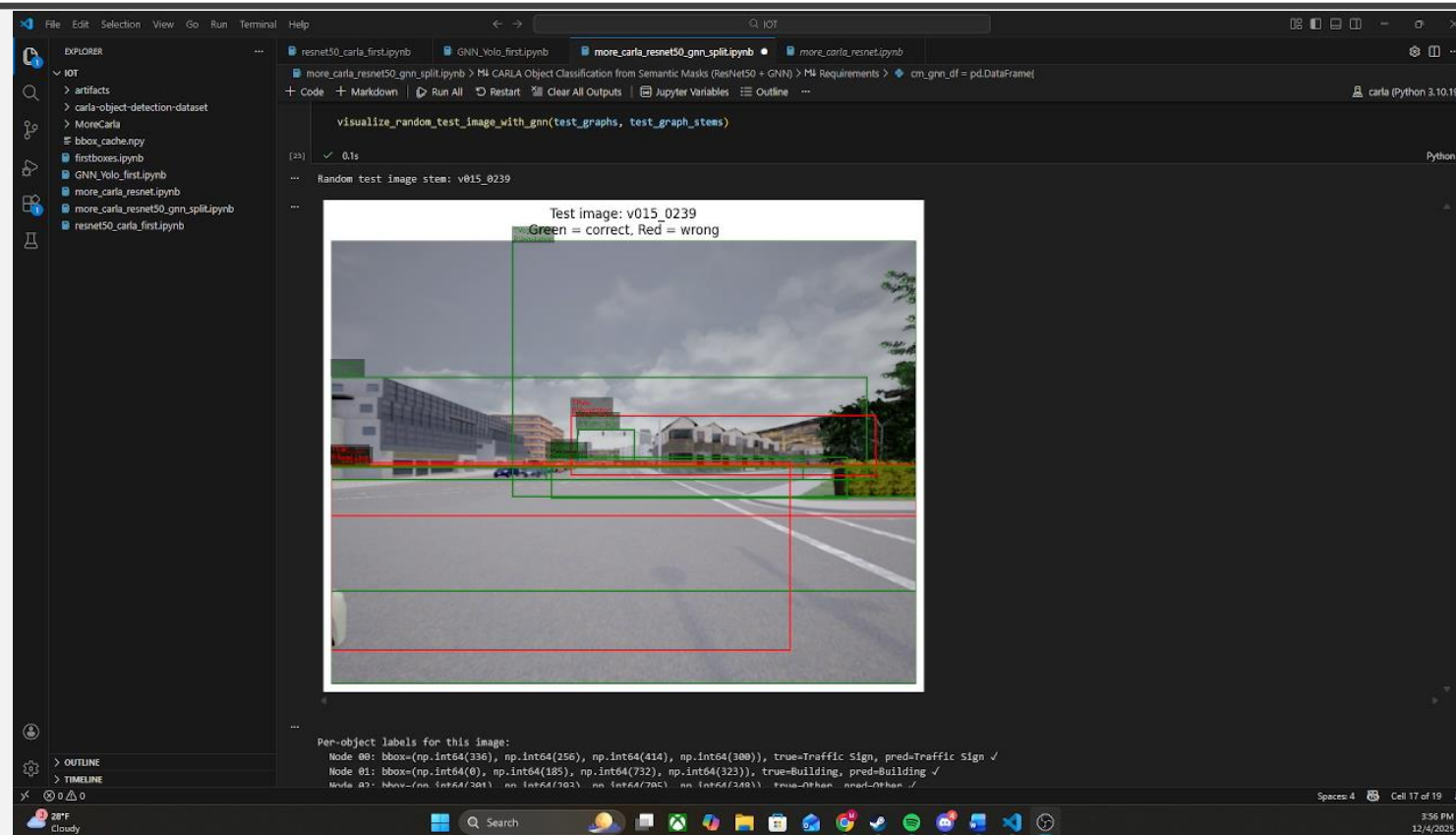


CNN + GNN COMBO



	precision	recall	f1-score	support
Traffic Sign	0.599	0.889	0.716	2009
Building	0.746	0.710	0.728	3200
Fence	0.620	0.958	0.753	780
Other	0.701	0.791	0.743	2637
Pedestrian	0.840	0.512	0.637	3109
Pole	0.747	0.541	0.628	3157
Road Line	0.473	0.676	0.556	3200
Road	0.492	0.243	0.325	3200
Sidewalk	0.681	0.747	0.713	3200
Vegetation	0.667	0.737	0.700	2749
Car	0.579	0.664	0.619	3211
Wall	0.785	0.678	0.727	2359
accuracy			0.649	32811
macro avg	0.661	0.679	0.654	32811
weighted avg	0.662	0.649	0.640	32811

DEMO



TIMELINE

Week	Objective
11	Switched Development to CNN (ResNet50) - Completed
12	Combine CNN with GNN - Completed
13	Final Presentation/ Start Final Report – In-Progress
14	Finish and Revise Final Report – In-Progress

PROJECT MANAGEMENT

Team Member	Task
Kevin	Generated Dataset + Ran CARLA
Matt	Trained Models
Ankith	Data Engineering Tasks – Cleaning + Prep for training

CHALLENGES

- Version issues
 - Python and CARLA must be on same or similar versions
- Ports/localhost issue
 - Set inbound rules in Windows Defender Firewall
 - Script refuses to connect to carla
- Training time
 - With our large amount of data, training requires a long time without interruption

CONCLUSION AND NEXT STEPS

- Finish CNN Development + Finetuning
- Utilize CNN in relation to a GNN to see relations between driver and other nodes from CNN
- Test and evaluate against newly generated set
- Record results in Paper

RESOURCES

- <https://www.kaggle.com/datasets/suraj520/carla-object-detection-dataset?resource=download>
- <https://www.kaggle.com/datasets/albertozorretto/carla-densely-annotated-driving-dataset?resource=download>