Temporal Logic For Anomaly Detection In Traffic Videos

CS599 Final Project
Nicole Fronda

Agenda

- Background
- Problem Setup
- Implementation
- Results
- Conclusion

Background

NVIDIA AI City Challenge - Anomaly Detection

Problem Setup

Pipeline

- Annotate Videos
  - Identify and Label Vehicles in Video Frames
  - Identify Road Lanes
  - Calculate Car Velocities
- Create Traces
- Develop Signal Temporal Logic Rules to Detect Anomalies
- Project Traces Through Signal Temporal Logic Rules
- Cluster Traces
- Identify Anomalous Clusters

Pipeline

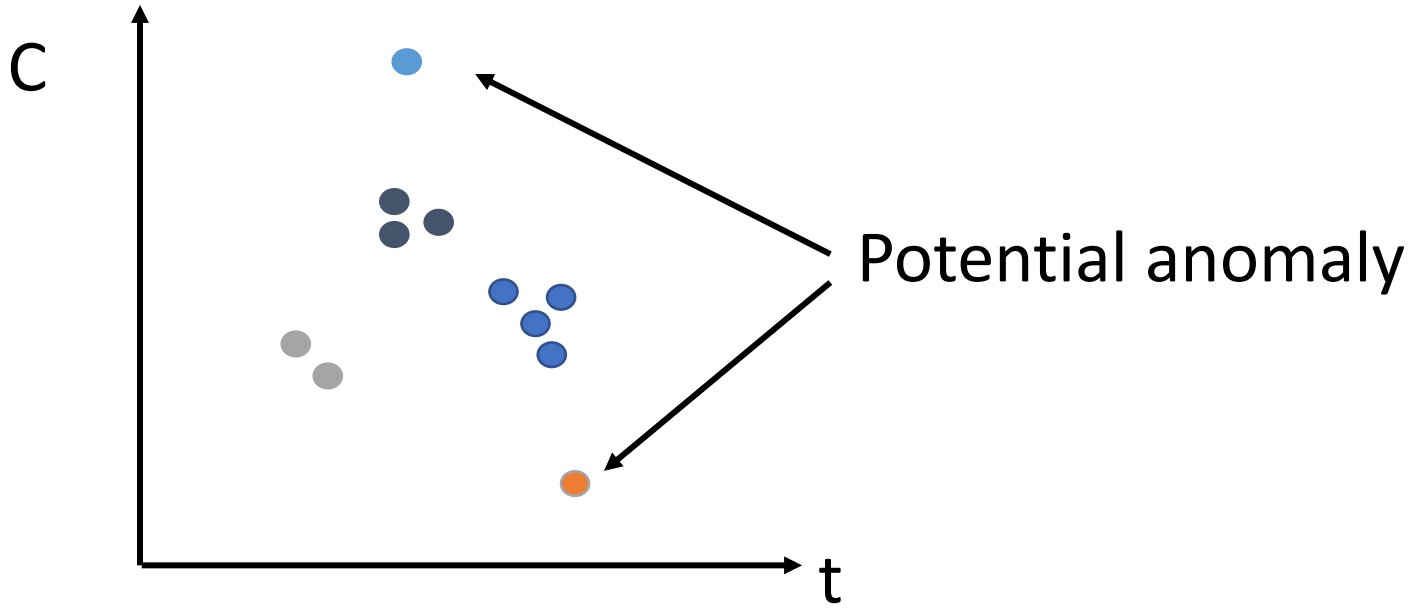
- Annotate Videos
  - Identify and Label Vehicles in Video Frames
  - Identify Road Lanes
  - Calculate Car Velocities
- Create Traces
- **Develop Signal Temporal Logic Rules to Detect Anomalies**
- Project Traces Through Signal Temporal Logic Rules
- Cluster Traces
- Identify Anomalous Clusters

Anomalies

- Cars moving faster than average
- Cars moving slower than average
- Cars suddenly braking
- Cars suddenly accelerating
- Cars parked in the shoulder

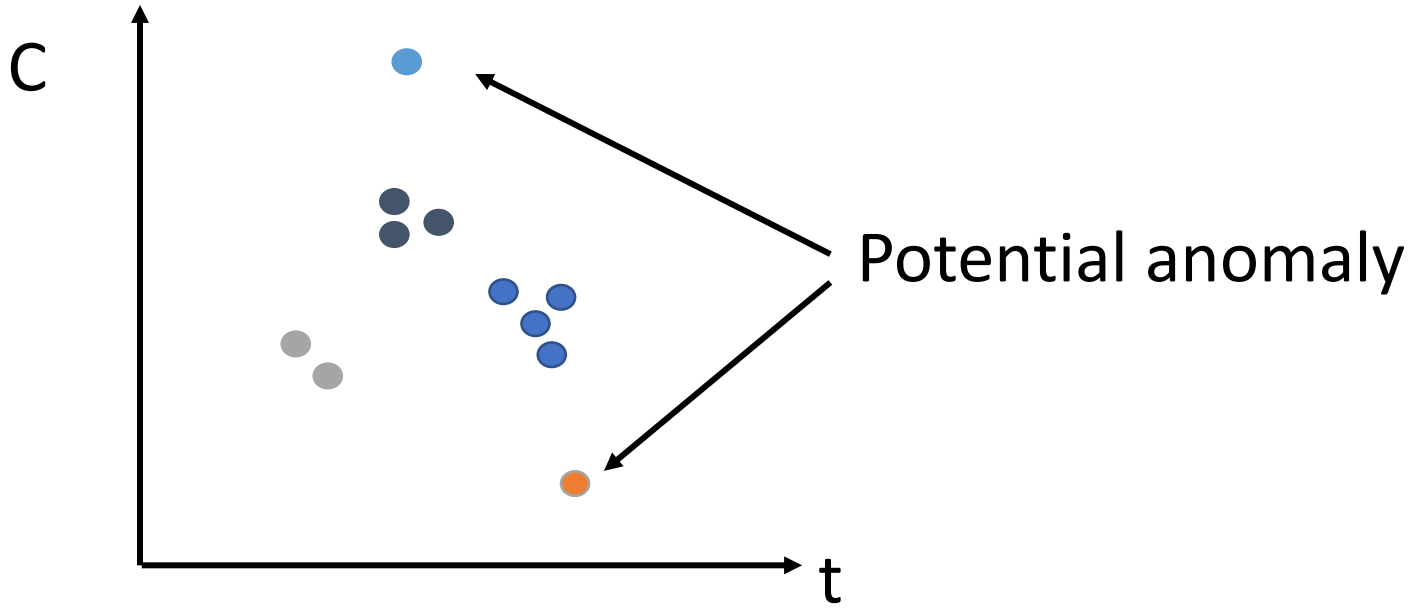
Anomalies in Signal Temporal Logic

- Cars moving faster than average
  - **F ((v[t] – avg_v) > C)**
    - Look for signal clusters with really high C
    - Clusters with really low C are cars moving slower than average



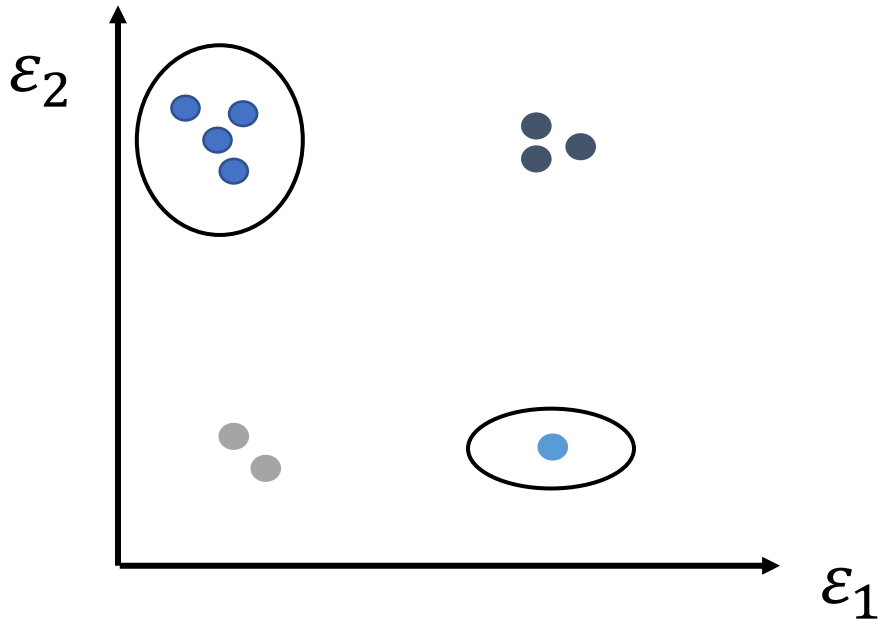
Anomalies in Signal Temporal Logic

- Cars moving slower than average
  - **F ((avg_v - v[t]) > C)**
    - Look for signal clusters with really high C
    - Clusters with really low C are cars moving faster than average



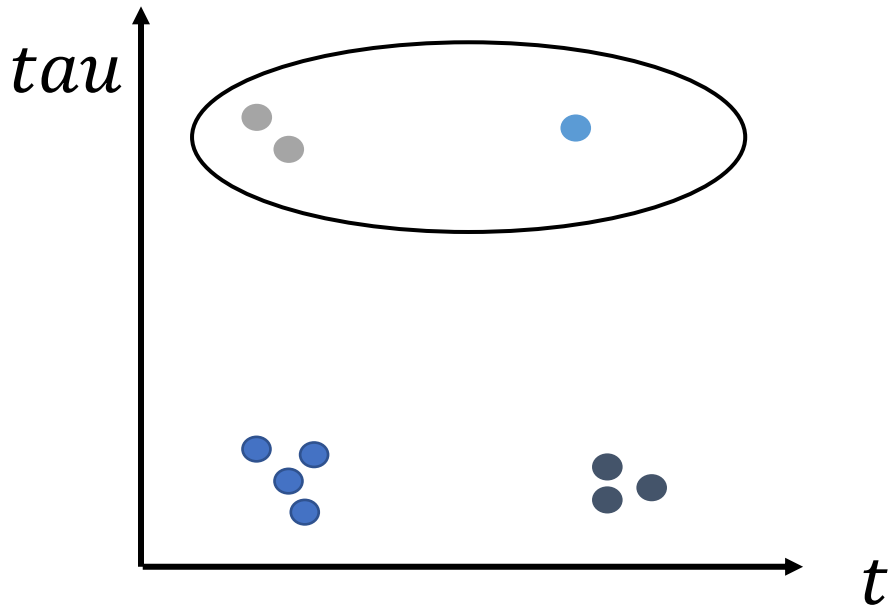
Anomalies in Signal Temporal Logic

- Cars suddenly braking or suddenly accelerating
  - **G (velocity(t) > ε_1=> $F_{[0,0.5]}$(velocity(t) < ε_2)**
    - Cars suddenly braking will be in clusters with large ε_1 and small ε_2
    - Cars suddenly accelerating will be in clusters with large ε_2 and small ε_1



Anomalies in Signal Temporal Logic

- Cars parked in the shoulder
  - $\boldsymbol{F_{[0,tau]}}$**(lane[t] == shoulder)**
    - Cars stopped in the shoulder will be in clusters with high tau



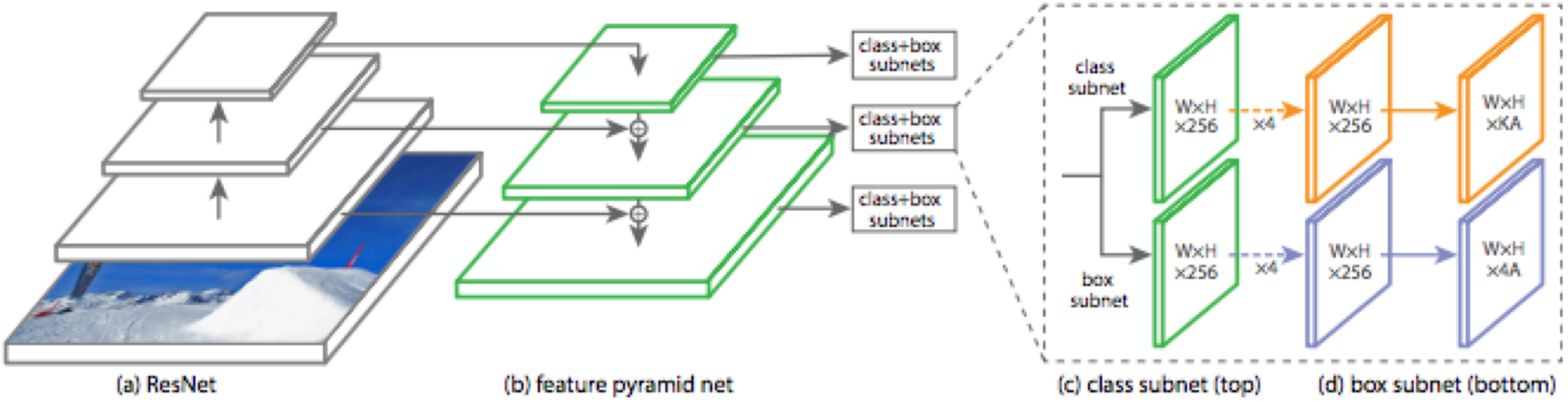
Implementation

Pipeline

- Annotate Videos
  - Identify and Label Vehicles in Video Frames
  - Identify Road Lanes
  - Calculate Car Velocities
- Create Traces
- Develop Signal Temporal Logic Rules to Detect Anomalies
- Project Traces Through Signal Temporal Logic Rules
- Cluster Traces
- Identify Anomalous Clusters

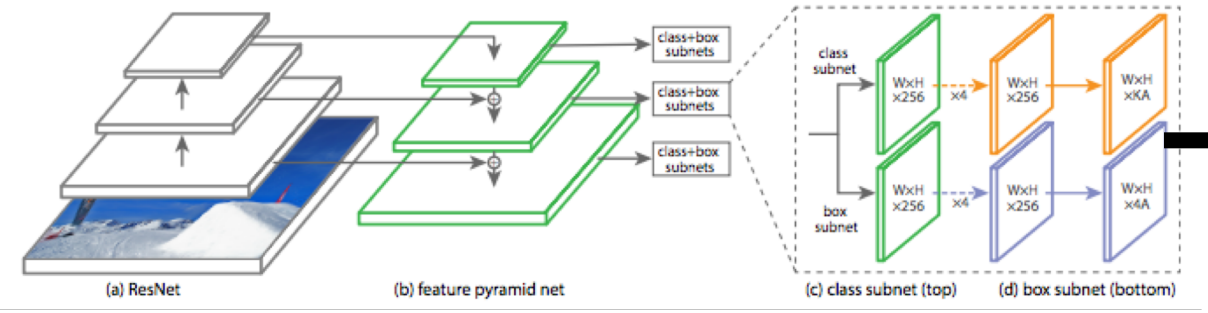
Annotate Videos: Identify Vehicles

- 2 Stage Neural Network



- [Focal Loss for Dense Object Detection]

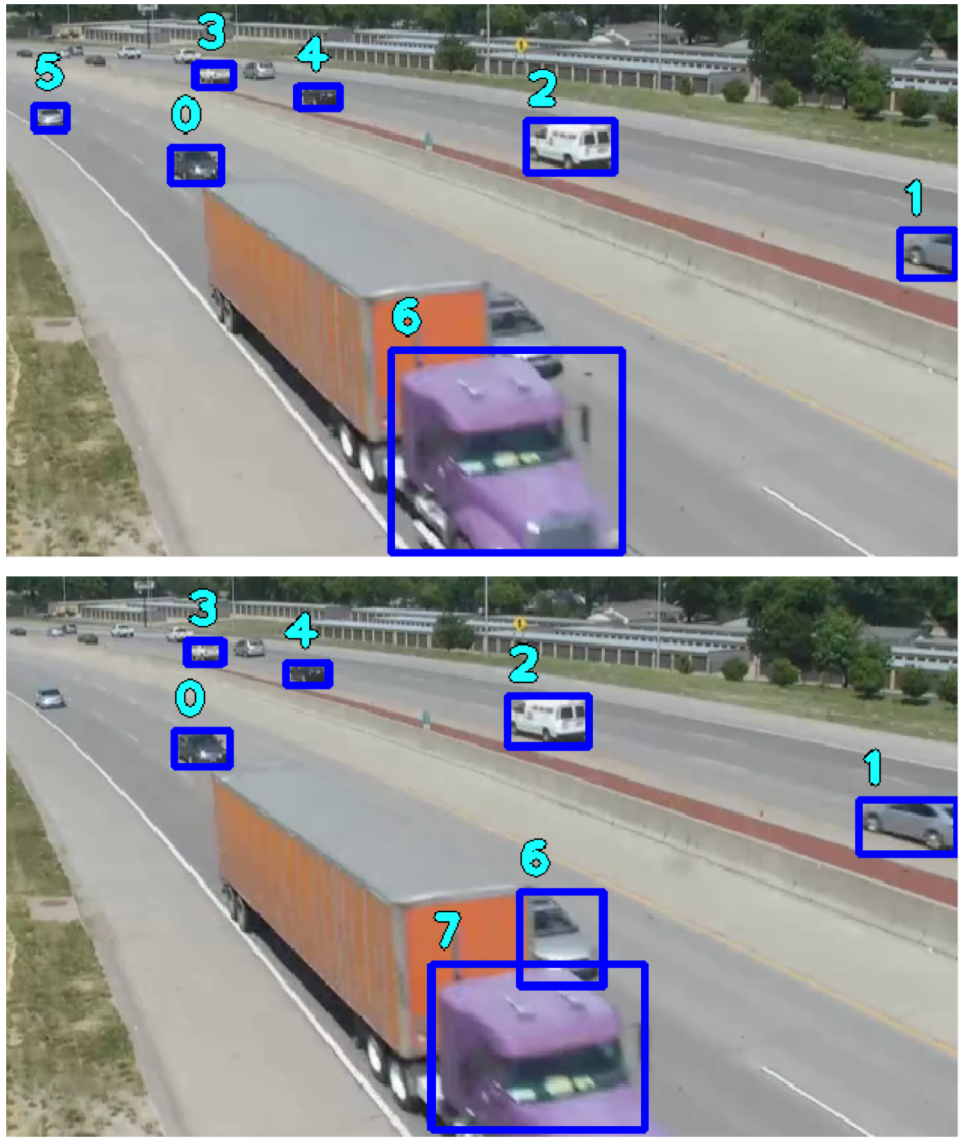
Annotate Videos: Label Vehicles



Annotate Videos: Label Vehicles

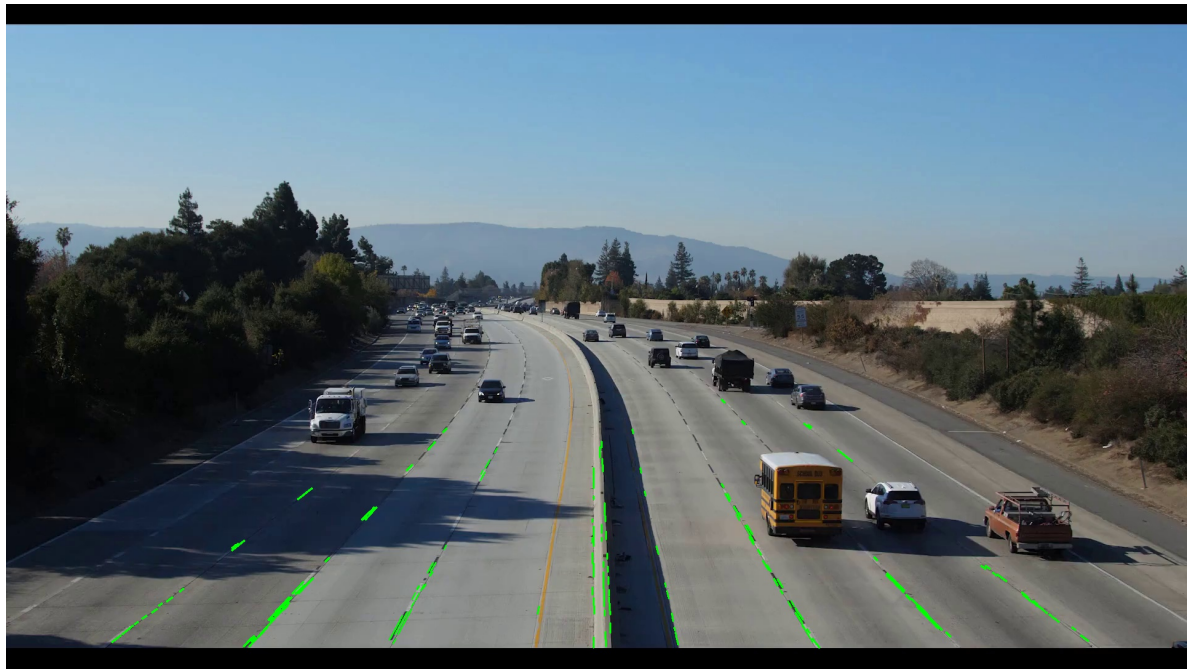
```
previous_positions = {}
for f in frames:
  for car in f:
    closest_car = find_closest(car(x, y),  previous_positions)
    if there is no closest_car:
      add new car to previous_positions
    else:
      update previous position of car
```

Previous_positions = {c0 : [(x1, y1), (x2, y2)],
c1 : [(x1, y1), (x2,y2)],
…
c5 : [(x1, y1)],
c6 : [(x1, y1), (x2, y2)],
c7 : [(xy, y2)]}



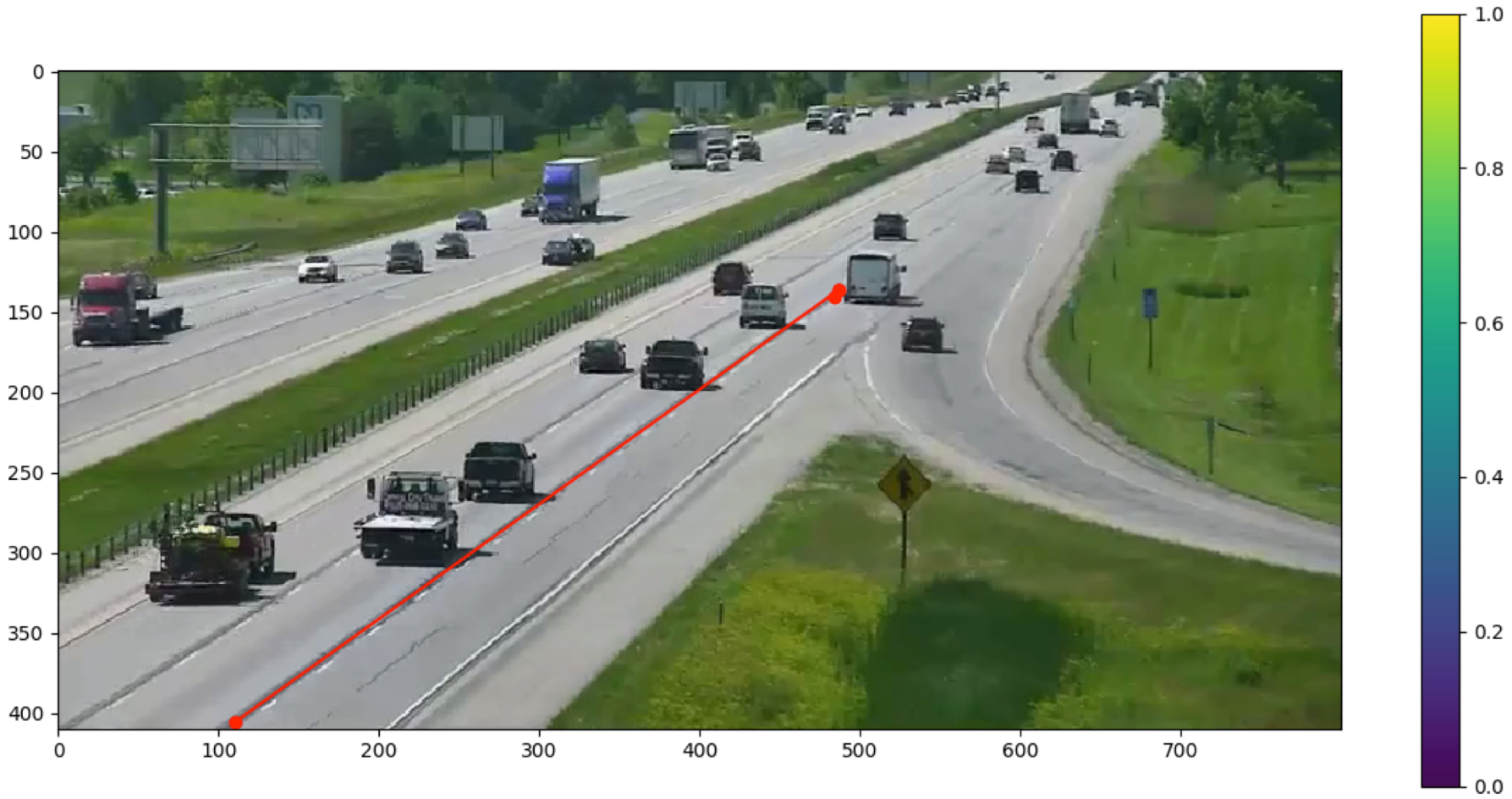
Annotate Videos: Identify Road Lanes

Canny Edge Detection



Hough Transform

Annotate Videos: Identify Road Lanes (alternative solution)

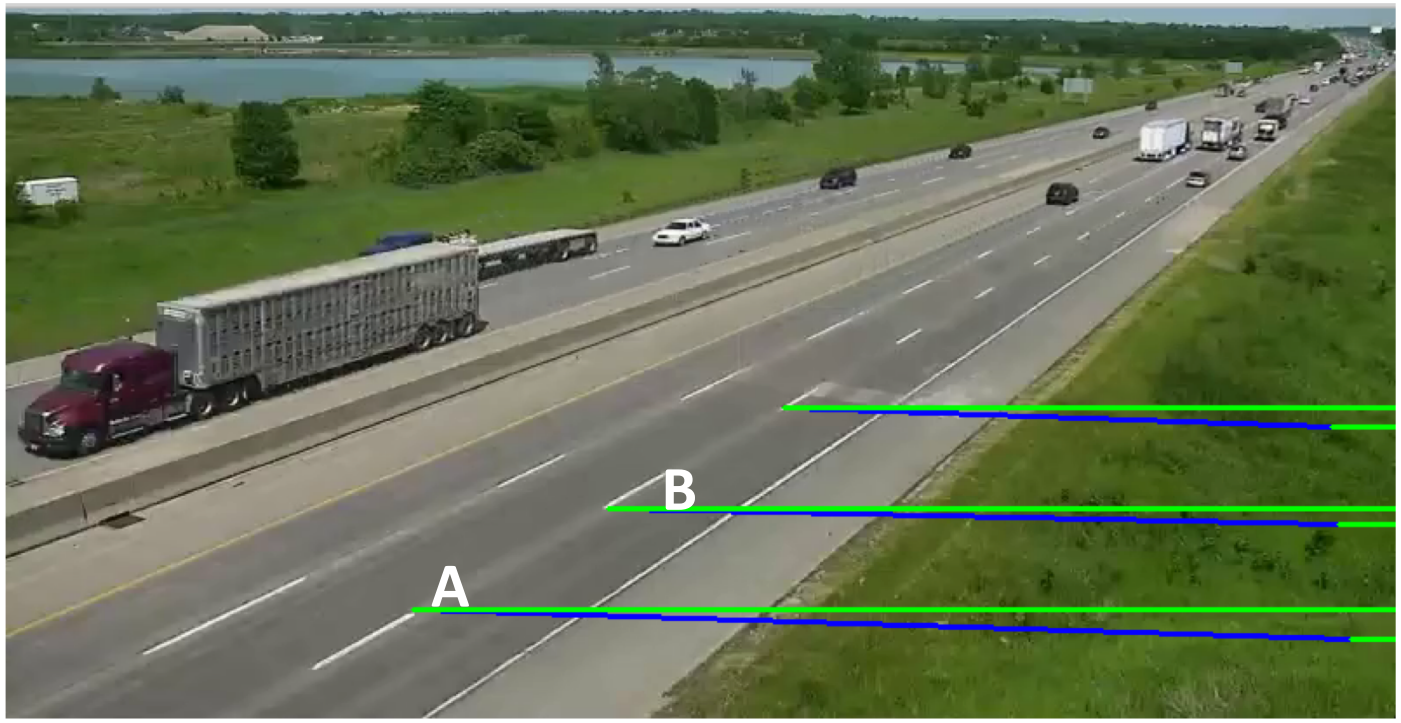


Annotate Videos: Calculate Velocity

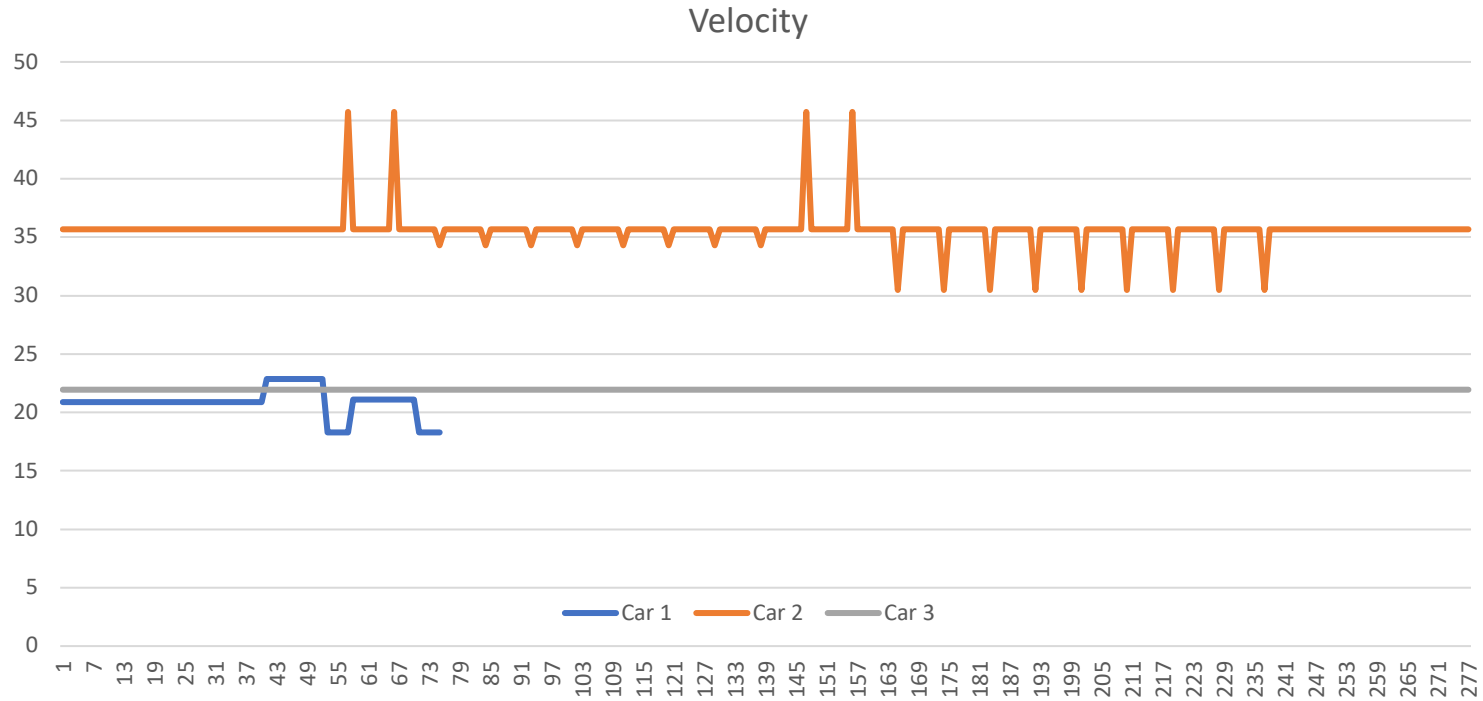
N = # frames between A and B

dist = actual distance between A and B

V = dist / (N / 30 fps)



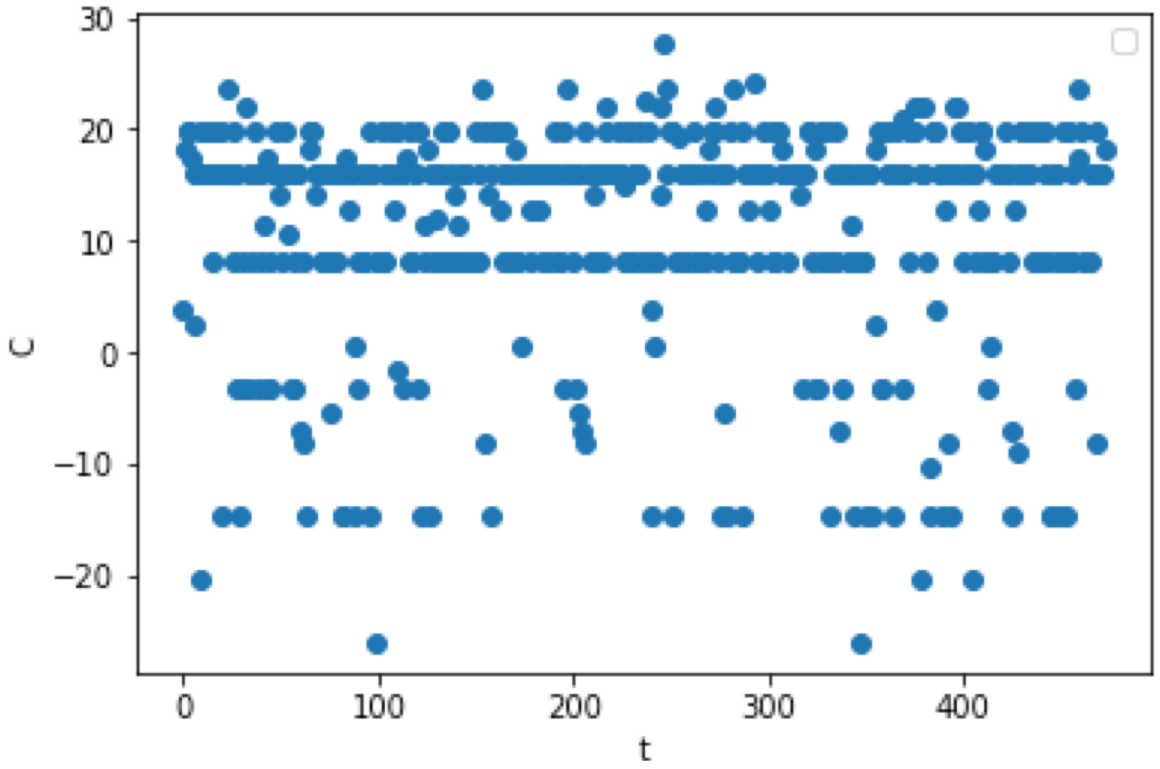
Project Traces Through Signal Temporal Logic

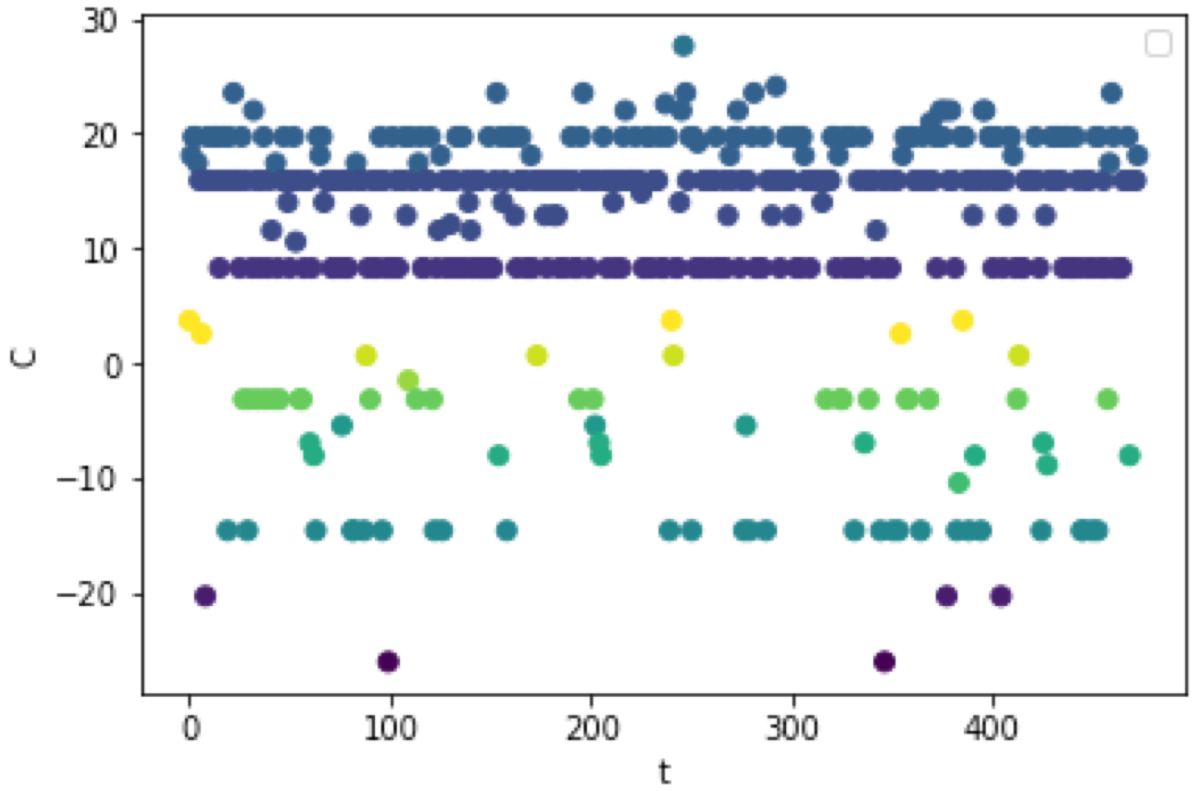



```
S = BreachTraceSystem({'v'},[times', velocity]);

%%% Cars Moving Slower than average (find high c)
spec2 = STL_Formula('mu', 'ev_[t0,T] ((avg_v - v[t]) > c)');
spec2 = set_params(spec2, {'t0', 'T', 'avg_v'}, [times(1) times(end) avg_velocity]);
P = ParamSynthProblem(S, spec2, {'c'}, [-50, 50]);
c = P.solve();
```

Cluster Traces & Identify Anomalous Clusters

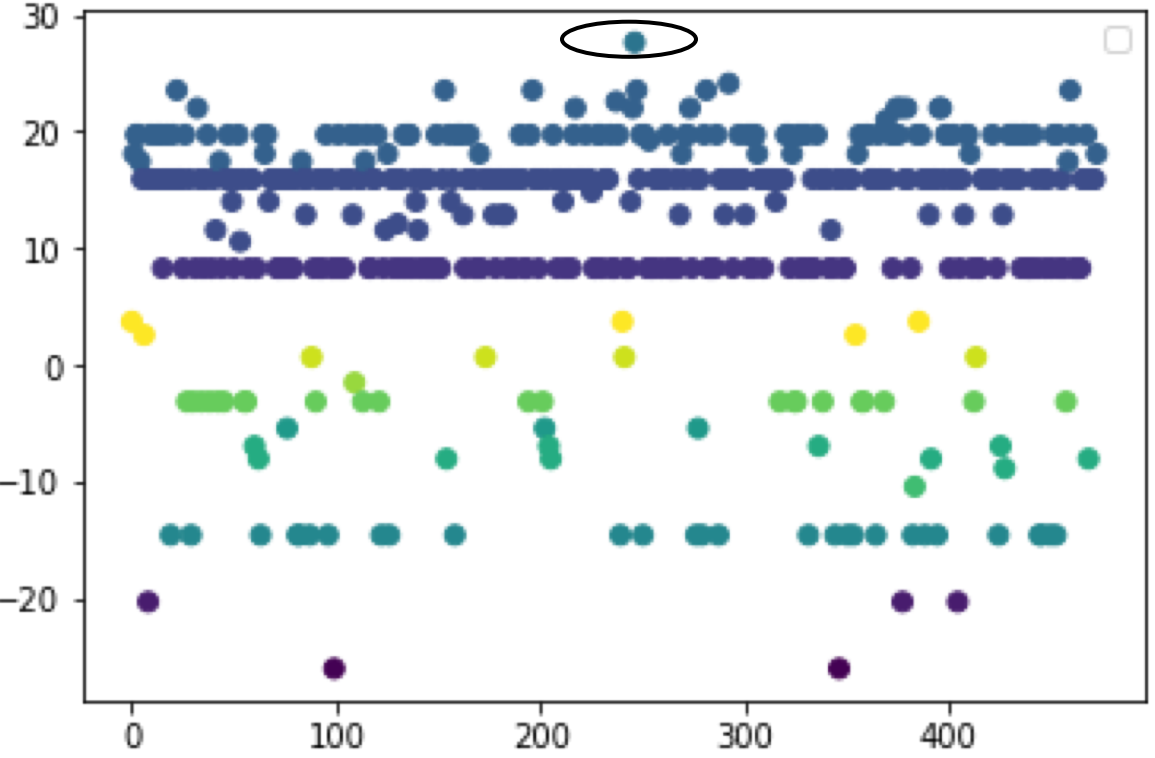




Results

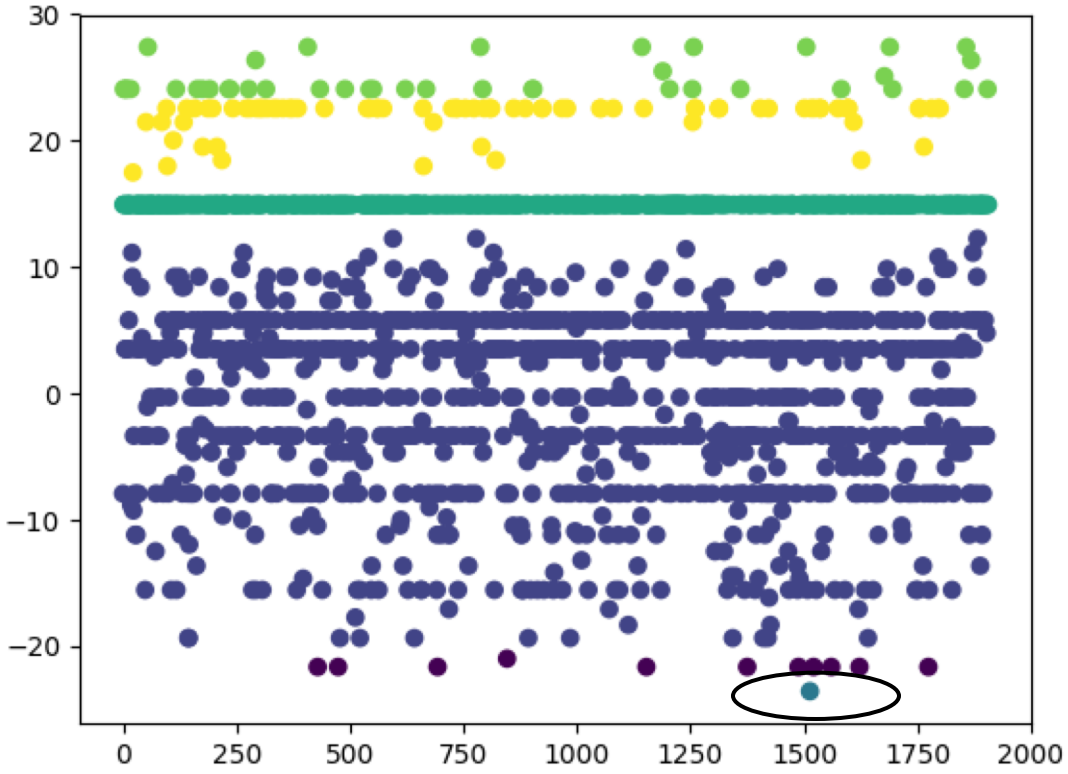
Cars Moving Slower Than Average

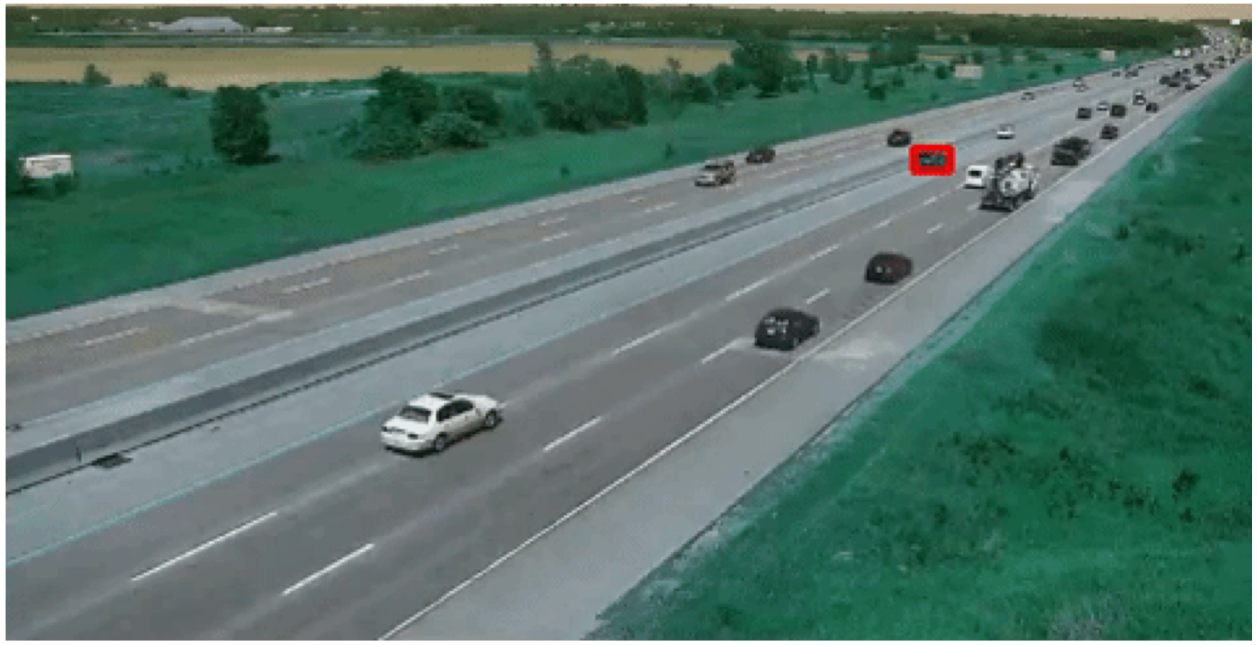
F ((avg_v - v[t]) > C)



Cars Moving Slower Than Average

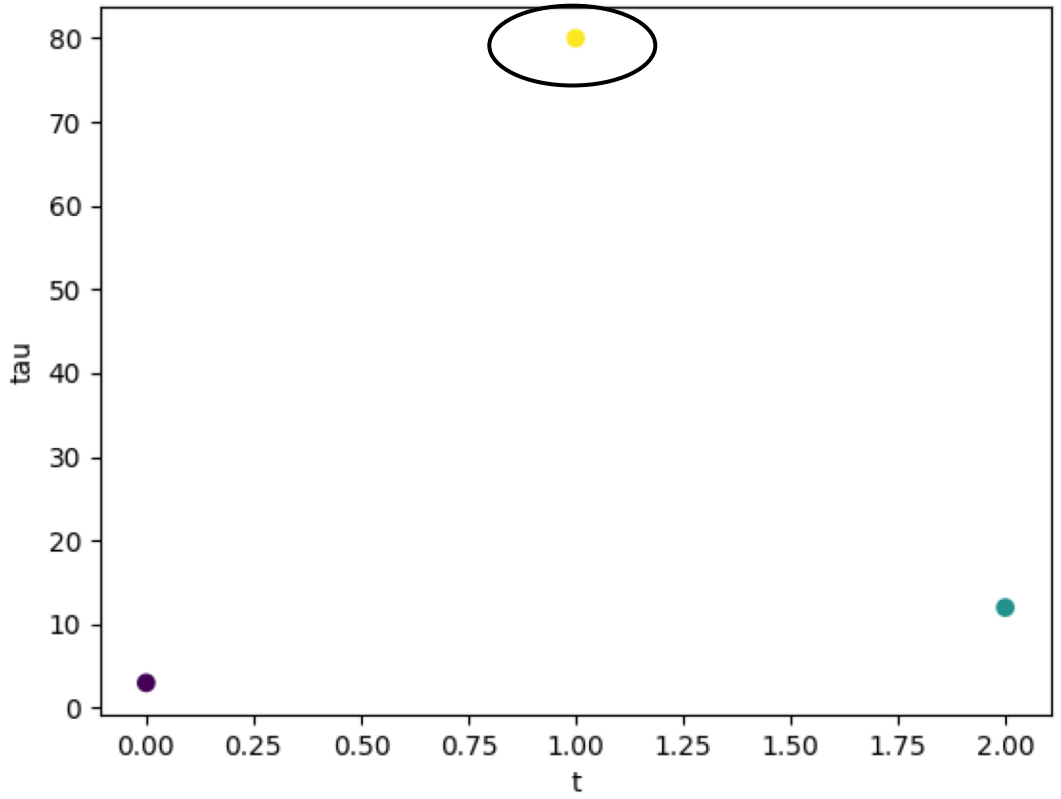
F ((v[t] – avg_v) > C)





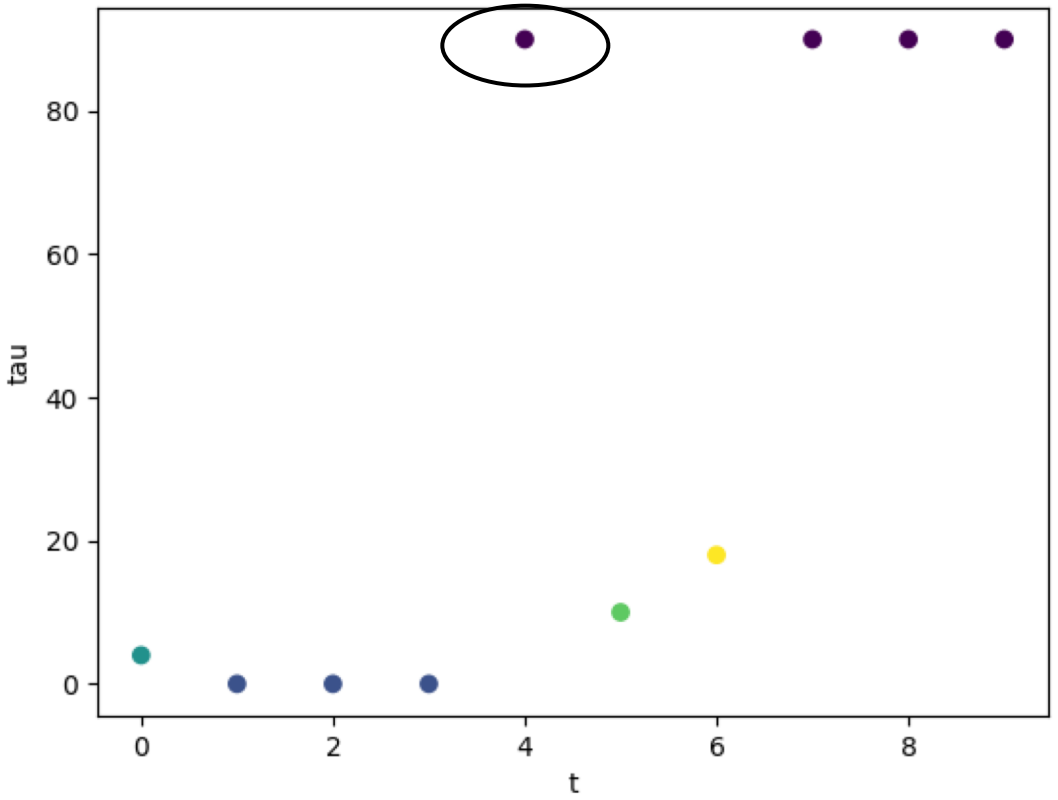
Cars parked in the shoulder

$F_{[0,tau]}$(lane[t] == shoulder)



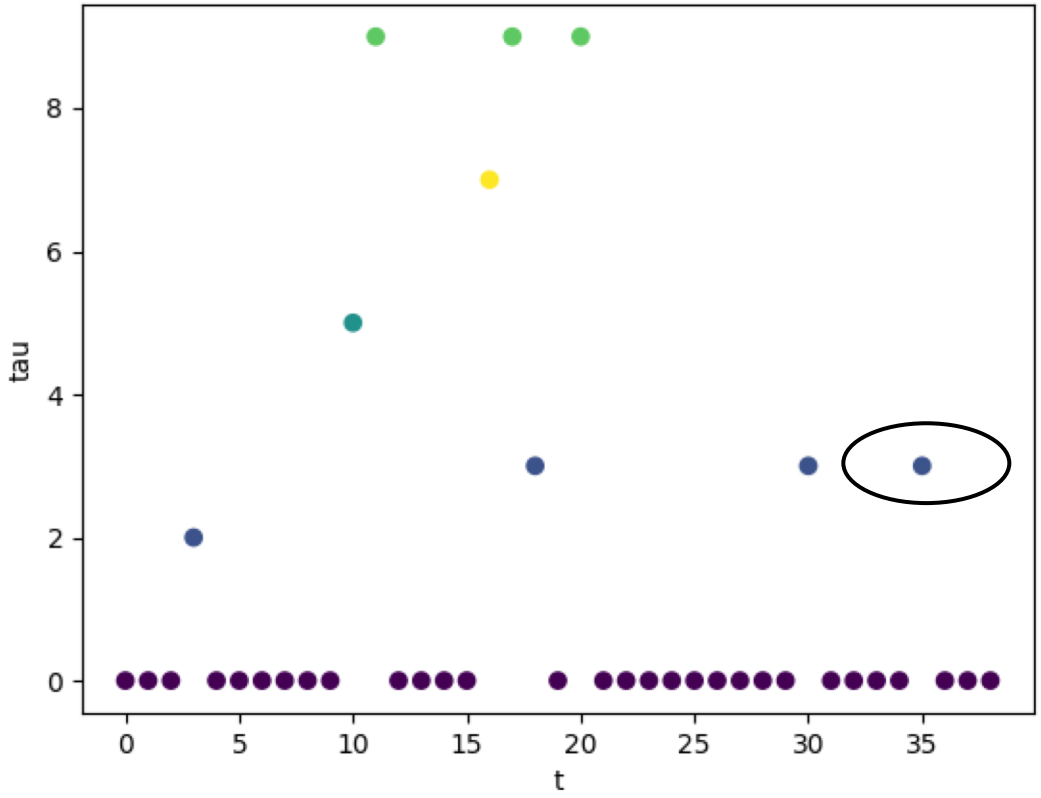
Cars parked in the shoulder

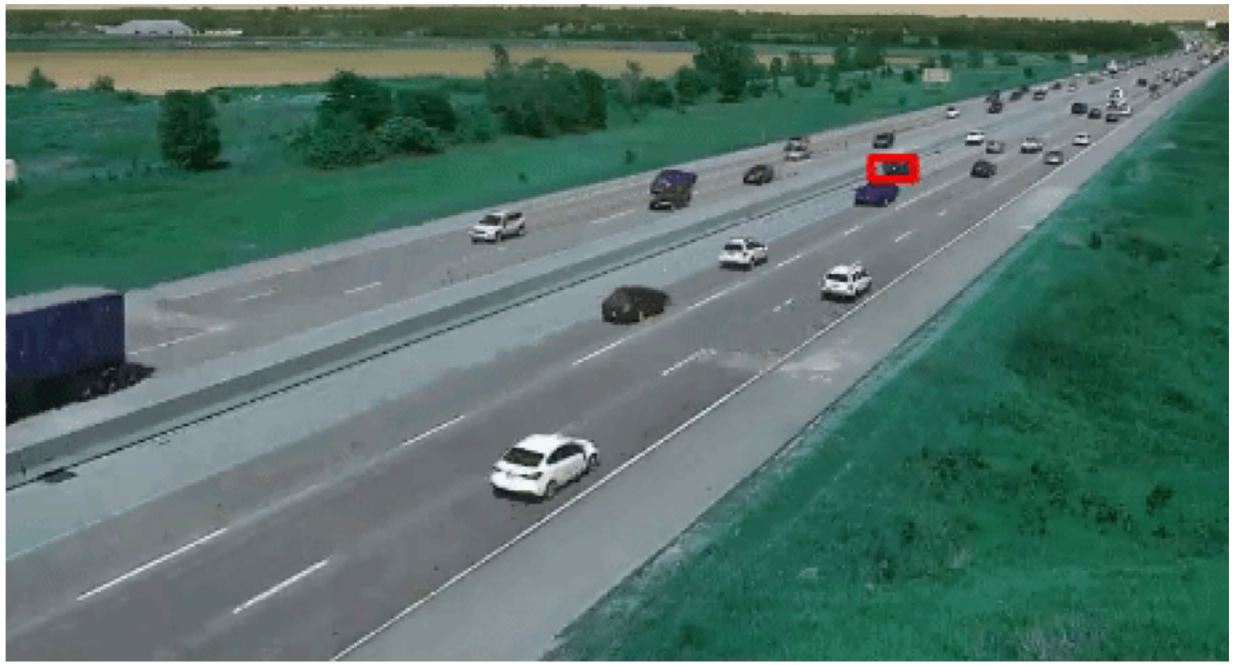
$F_{[0,tau]}$(lane[t] == shoulder)



Cars parked in the shoulder

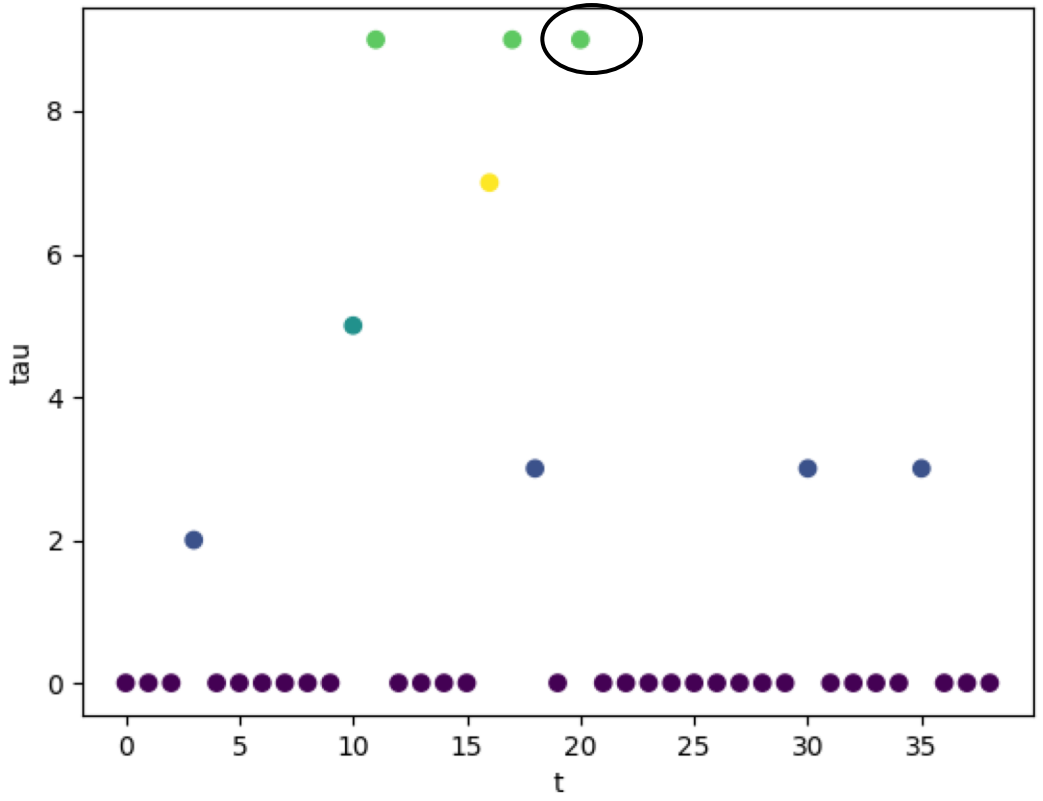
$F_{[0,tau]}$(lane[t] == shoulder)





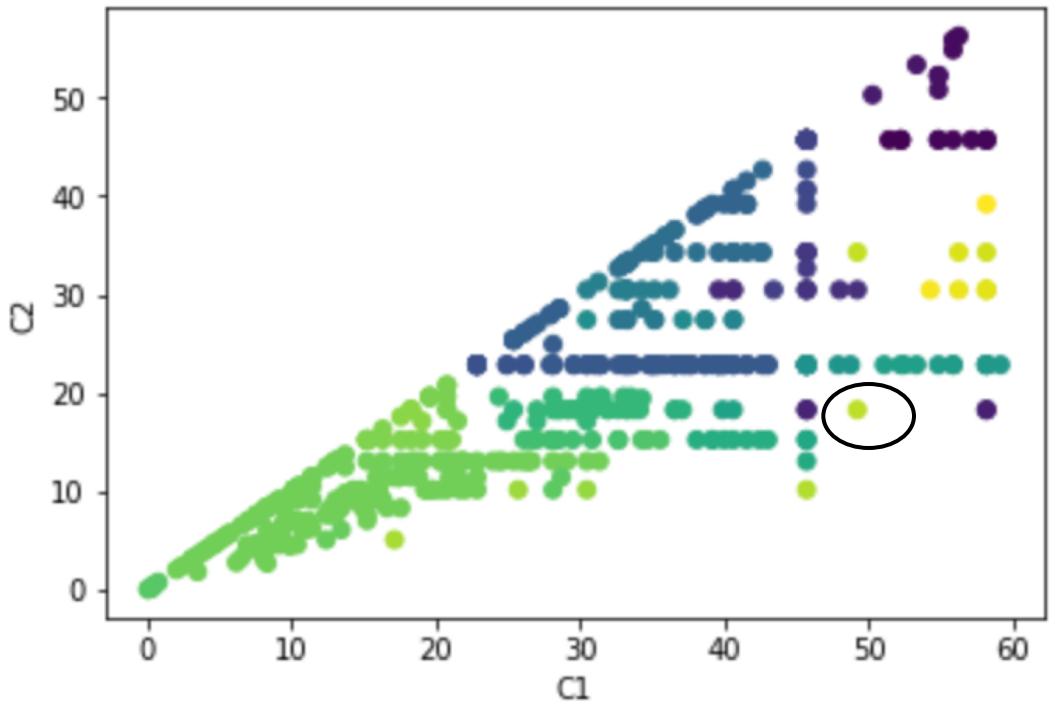
False Positives

$F_{[0,tau]}$(lane[t] == shoulder)



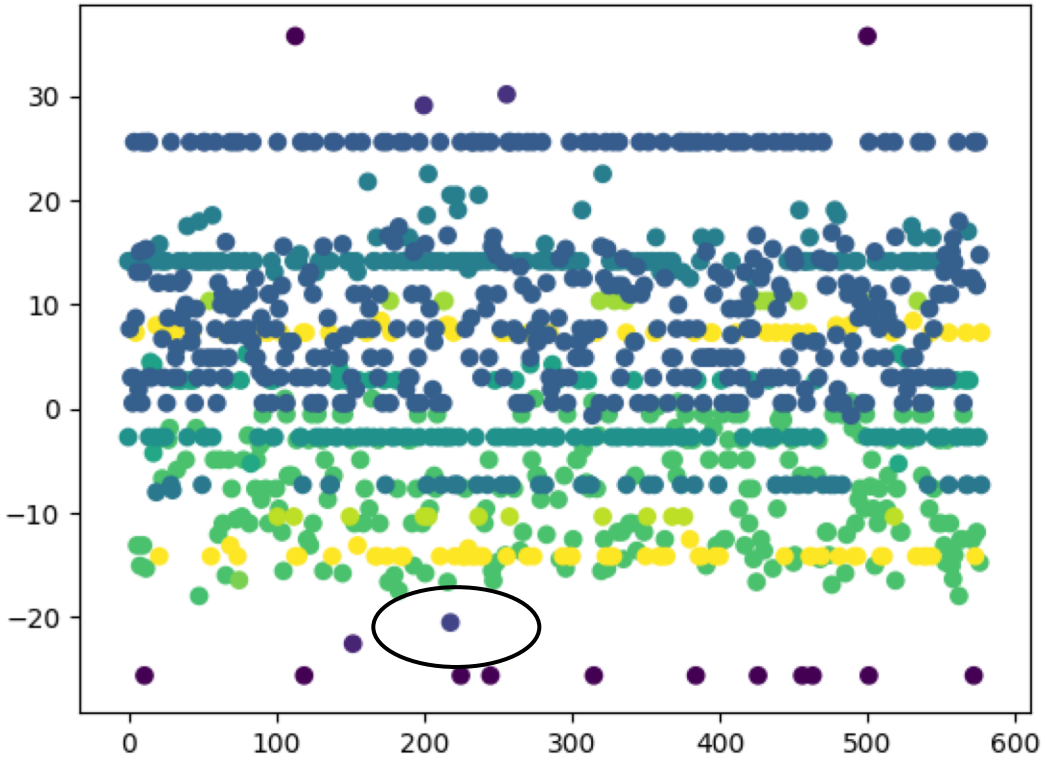
False Positives

G (velocity(t) > ε_1=> $F_{[0,0.5]}$(velocity(t) < ε_2)

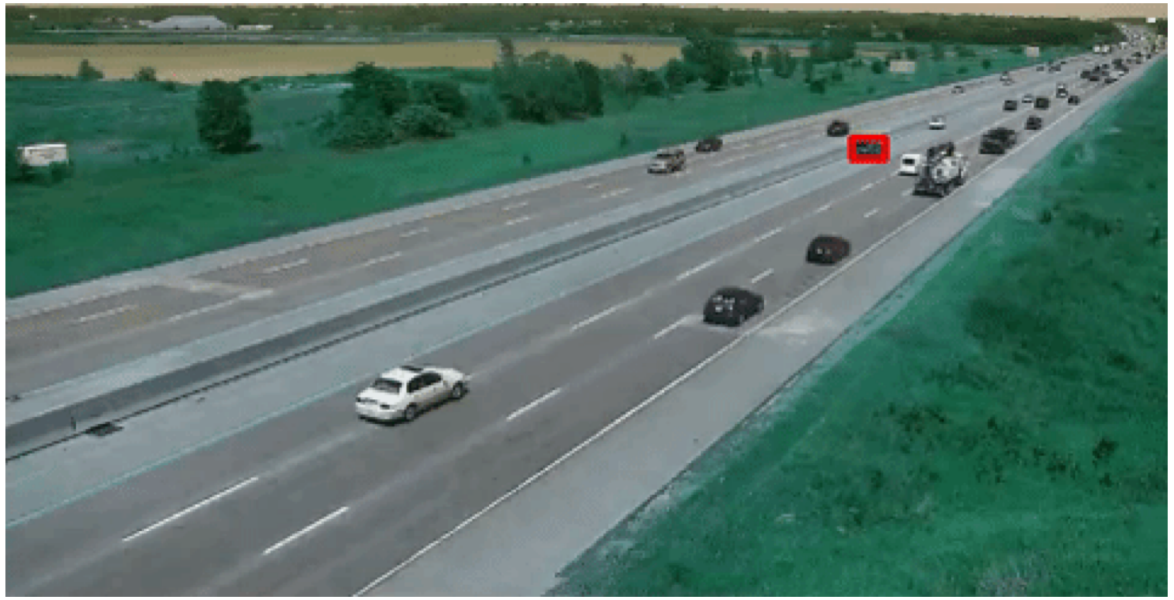


False Positives

F ((avg_v - v[t]) > C)



Time of Detection



Anomaly Detected: 10:47

Actual Start Time: 9:39

Time of Detection

Anomaly Detected: 5:07

Actual Start Time: 4:50

Time of Detection

Anomaly Detected: 6:56

Actual Start Time: 6:55

Overall Results

- Small Sample Size:
  - 20 videos
  - 8 Anomalies
  - 5 Anomalies Detected
- Precision: 13%
- Recall: 63%
- RMSE Time of Detection: 31.7

Conclusion

Accomplishments and Areas of Improvement

- Neural Network for Object Detection
- Lane Detection with Computer Vision
- Improvements for better Anomaly Detection
  - Fine Tune Object Detection and Labeling
  - Improving techniques for automatically finding regions of interest

Future Work

- More STL rules
- Build a Neural Network to Detect Anomalies and use STL to evaluate
- Build a model to work in real time

Q&A

https://github.com/nicaless/cs599_project