Capturing Real-Time Traffic Congestion with Dash Cam Video Feed

Eloy Frias, Christine Cho, Kevin Stutenberg

MScA 32019 – Real Time Intelligent Systems

Presentation Overview

- Problem Statement and Project Goals
- Data Properties, Cleaning, and Transformations
- Proposed Model Approaches
- Model Results and Proposed Solution
- Future Work

Problem Statement and Project Goals

Traffic congestion in US Cities has grown substantially in 20 years- costing each american 97 hours, $1,348 A Year [1]

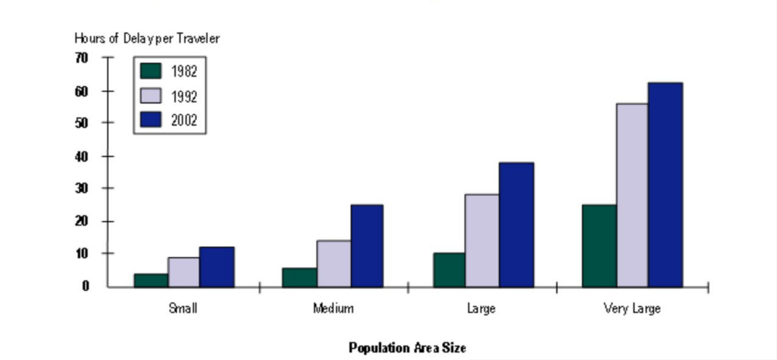


Figure ES.1 Congestion Has Grown Substantially in U.S. Cities Over the Past 20 Years

Source: The 2005 Urban Mobility Report, http://mobility.tamu.edu.

Advances in Mobility provide a new, unique data source which may be aid in improving congestions

Project Goal

Development of a low cost method for determining traffic congestion through video which could then be broadcast to a surrounding traffic infrastructure.

[1]- http://inrix.com/press-releases/scorecard-2018-us/

Proposed Process

Utilize an open source “dash cam” system and real-time video processing to capture surrounding vehicle traffic density.

Process:

- Replay / capture on-road data for use in the system development
- Open source enables real-time execution and control on device
- Preliminary focus areas
  - Congested freeway
  - Image / video processing
  - Vehicle count vs classification
  - Traffic in specific lane

Comma.ai Eon w/ ChffrPlus

https://comma.ai/

Data Properties and Cleaning

- Data captured from direct driving- “Available” datasets did not pan out

Data capture→ Cloud storage→ Segment Data→ Download & Convert

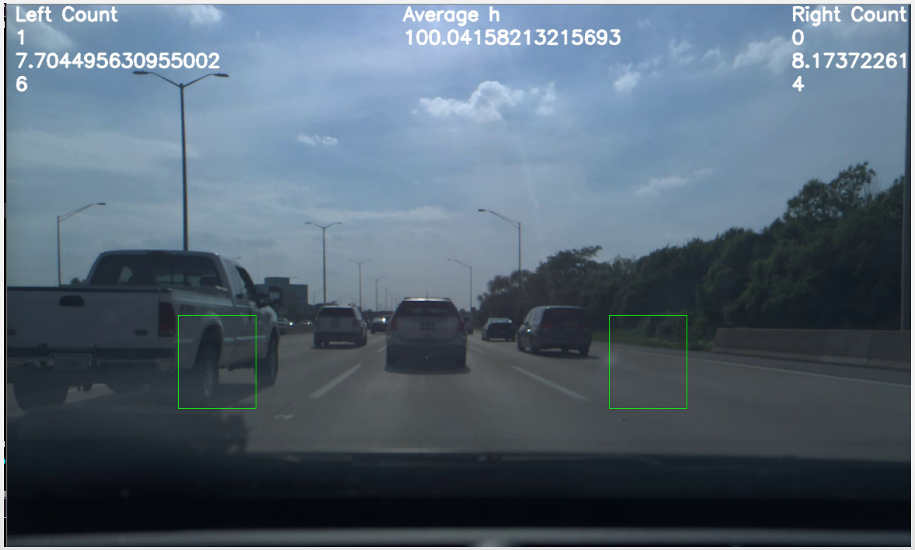
- Direct video upload to Comma.AI servers adds
- Video Conversion: HVEC to MP4
  - OPENCV +HVEC(h.265) = ☹
  - Solution- VLC conversion to MP4

Now that we have the video…

Lets Play!

Method 1-

Lane counting varying colorspace threshold limits within a cropped area

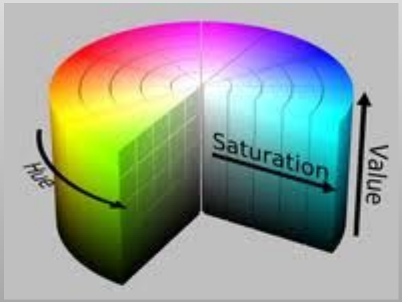


Process:

- OpenCV (Python)
- Mask a sample area per lane
- Convert to HSV
- Mask to V metric (value)
- Threshold > Mean(V)
    - Indicates vehicle passing
- Increment counter on variable change
- Overlay key info on video feed

Key drawback -

Robustness.. does not transfer well



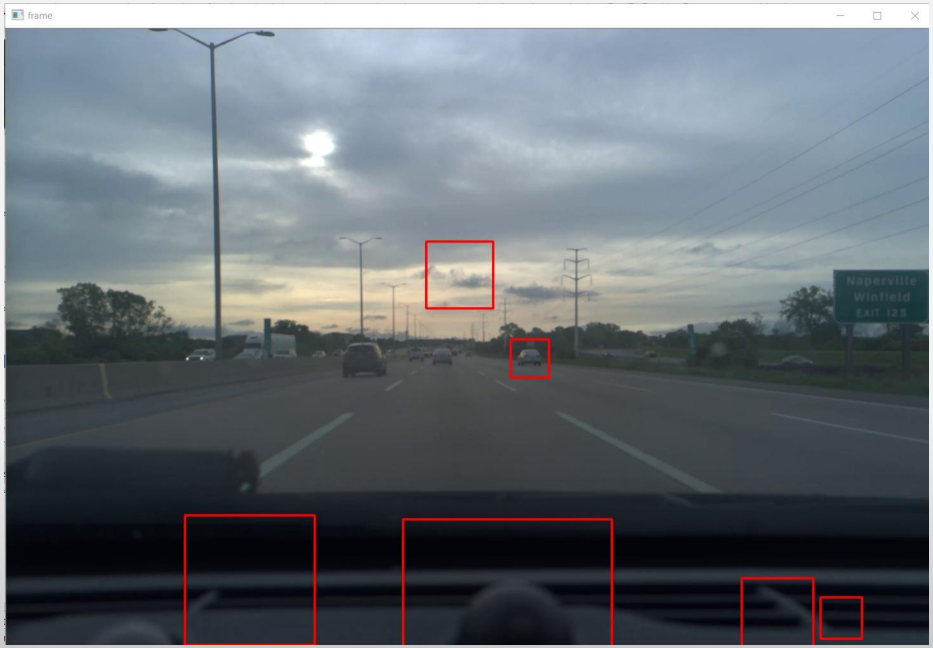
HSV Space

Method 2-

Cascade Classifier

Process:

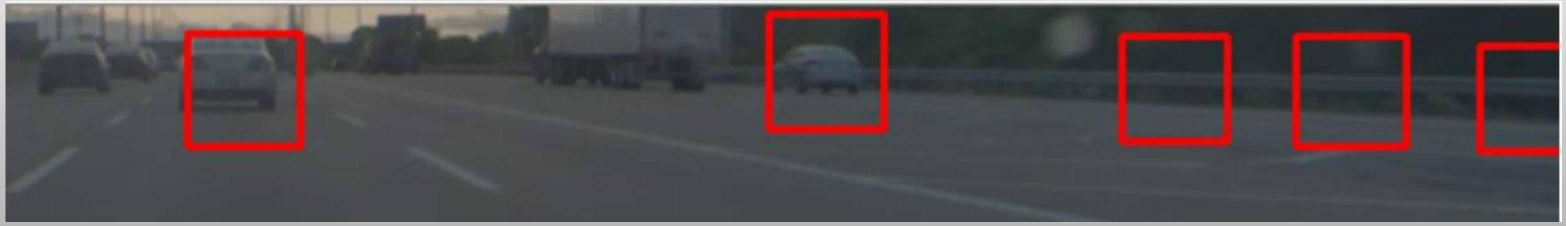
- Trained on images
- Positive/Negative examples
- 1 for match, 0 no match
- False positives in sky
- Stationary camera vs Dashcam



https://docs.opencv.org/2.4/modules/objdetect/doc/cascade_classification.html

Method 2-

- Eliminate the areas that cause false positives
- Crop out the areas that don’t have vehicles
- Guard rail posts identified as vehicles
- Autonomous vehicle crashed into cement posts two weeks before Intelligent Transportation Systems conference

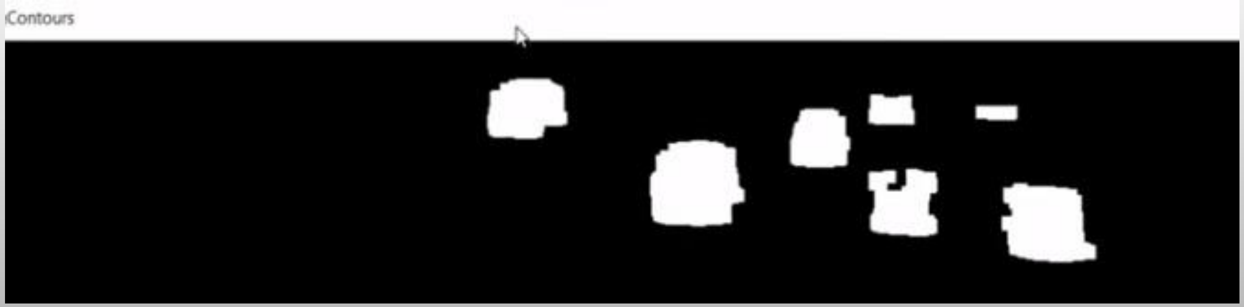


Method 3-

Contour Detection

Process:

- Convert to grayscale, cvtColor
- Get a threshold with adaptiveThreshold
- cv2.findcontours returns a list of contours, each contour is list of points of object parameter
- cv2.drawContours on the list of contours



https://docs.opencv.org/3.4/d4/d73/tutorial_py_contours_begin.html

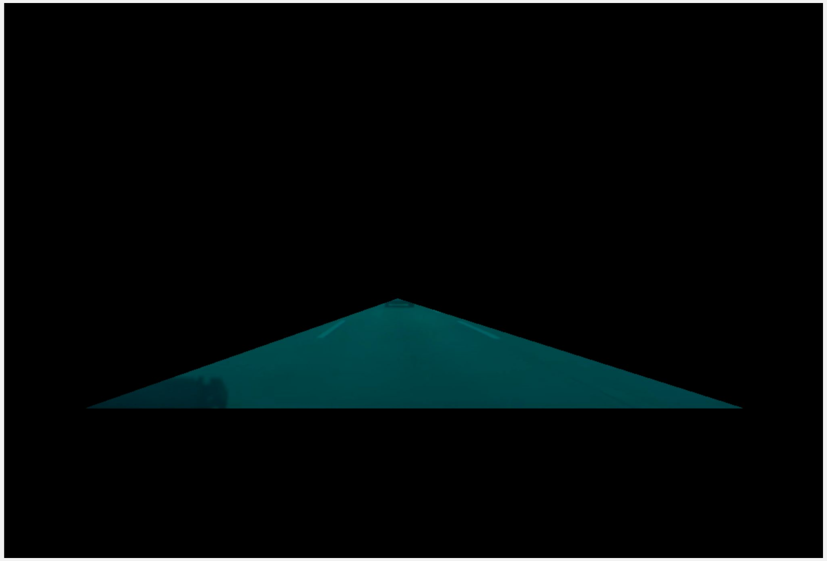
Bonus-

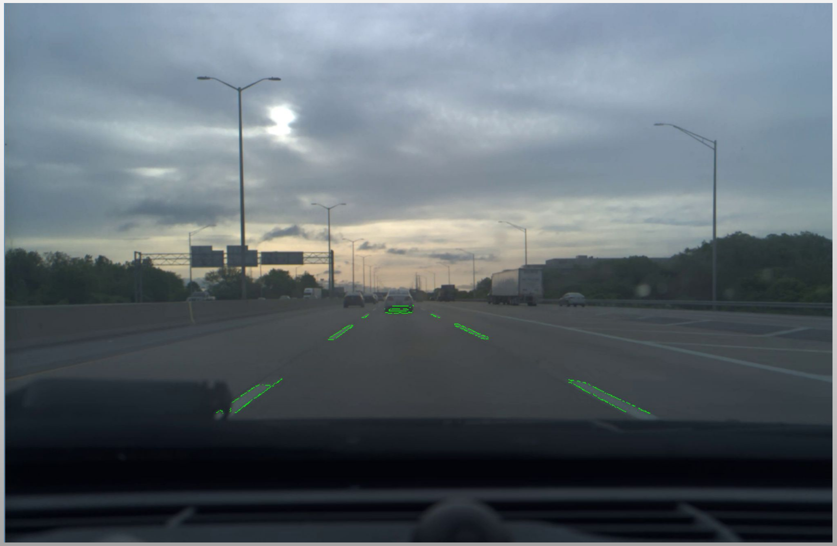
Lane Detection

Process:

- grayscale
- HSV, HSL scale tried
- blur
- Canny edge detection
- region of interest
- Hough Transform to detect lines
- plot Hough detections onto original image

https://arxiv.org/abs/1807.01726





Conclusion

- **Successful development of platform for real-time traffic congestion monitoring**
- Object detection a major benefit, but challenging to accurately accomplish

- Improvements needed for consistent results
  - Robustness for different brightness or camera angle
  - Varying roads- (Rural / Curvy)
  - Current set up only processes information, rather than recommends action for the user

Future Work

- Expand to rural/urban environments, drives with varying speed

- Gain robustness through ML/DL (LaneNet) rather than hard-coding thresholds
- Feedback to the driver for steering, lane change, etc.

- Direct hardware implementation
- Fusion of vehicle sensor data (radar, etc.)
- V2X communication of congestion at specific GPS coordinates

Questions?

References

- Urban mobility data- The 2005 Urban Mobility Report,
  - http://mobility.tamu.edu
- Hardware images and data collection- Comma AI:
  - https://comma.ai/
- Cascade classifier source:
  - https://github.com/andrewssobral/vehicle_detection_haarcascades
- Vehicle color space selection:
  - https://towardsdatascience.com/teaching-cars-to-see-vehicle-detection-using-machine-learning-and-computer-vision-54628888079a