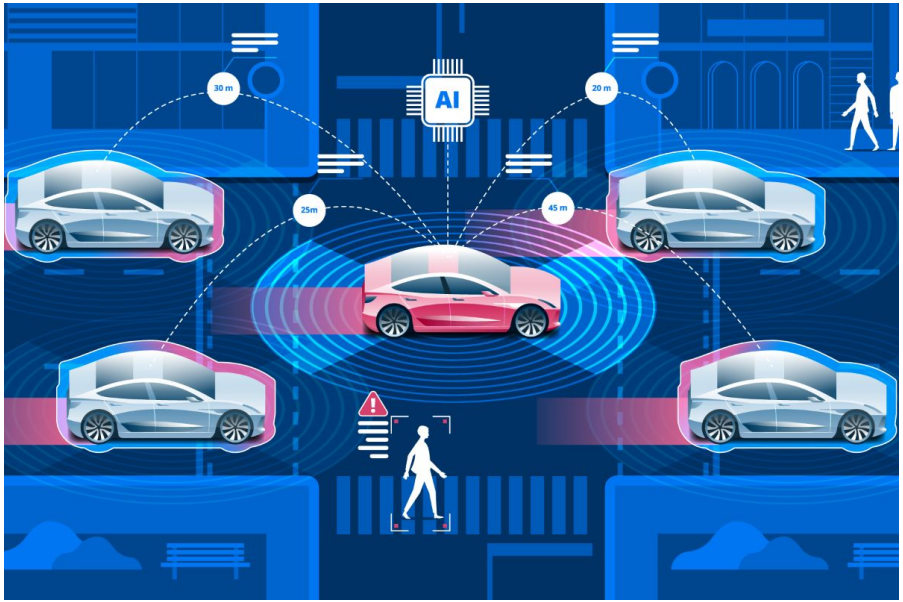
Driving Images Semantic Segmentation

W207 - Group 4
Fidelia Nawar, Tiffany Shih, Karina Tsang

Problem Statement & Objective

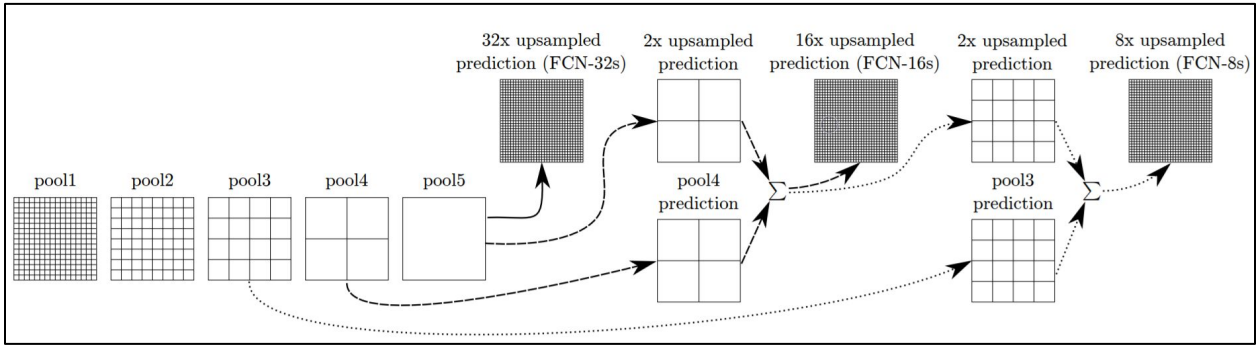


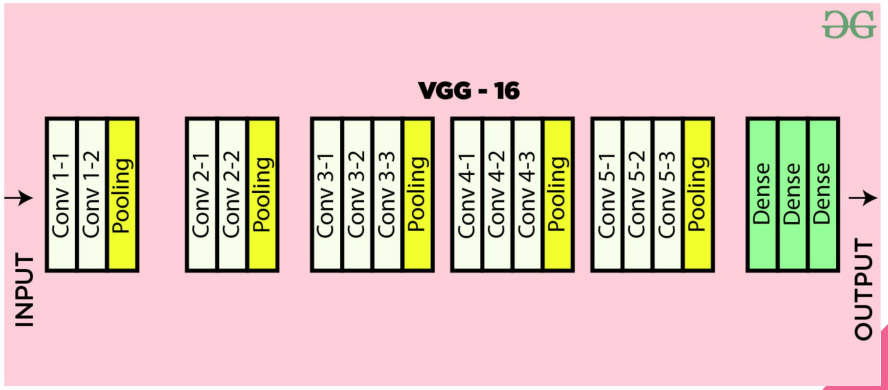
Successful automation requires a series of detection of the vehicle's surroundings:

- General segmentation of the environment
- Detection of nearby objects
- Detection of lane markings
- Speed of surrounding vehicles

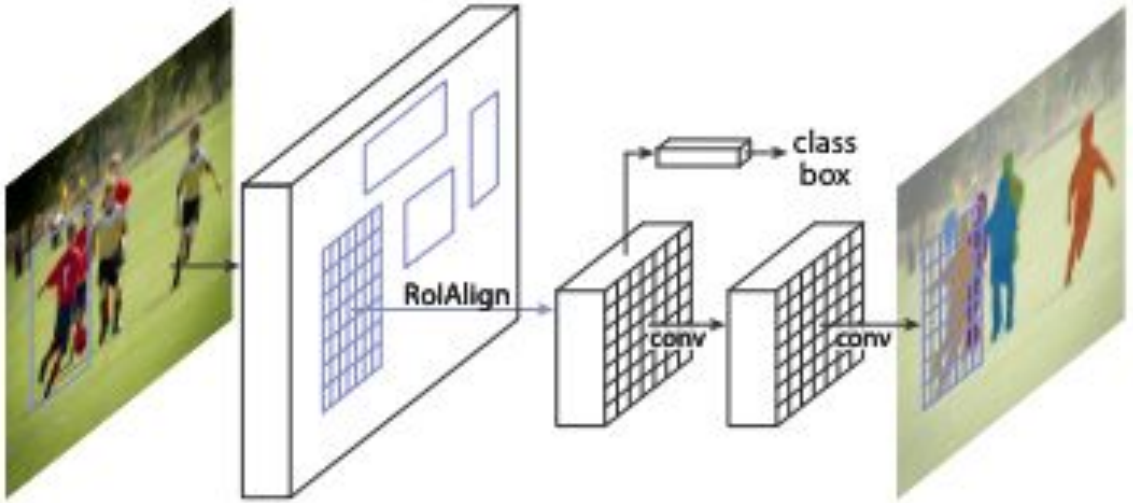
Image source: https://www.smartcitiesworld.net/opinions/opinions/driving-autonomous-vehicles-forward-with-intelligent-infrastructure

Approach - FCN





Approach - Mask R-CNN



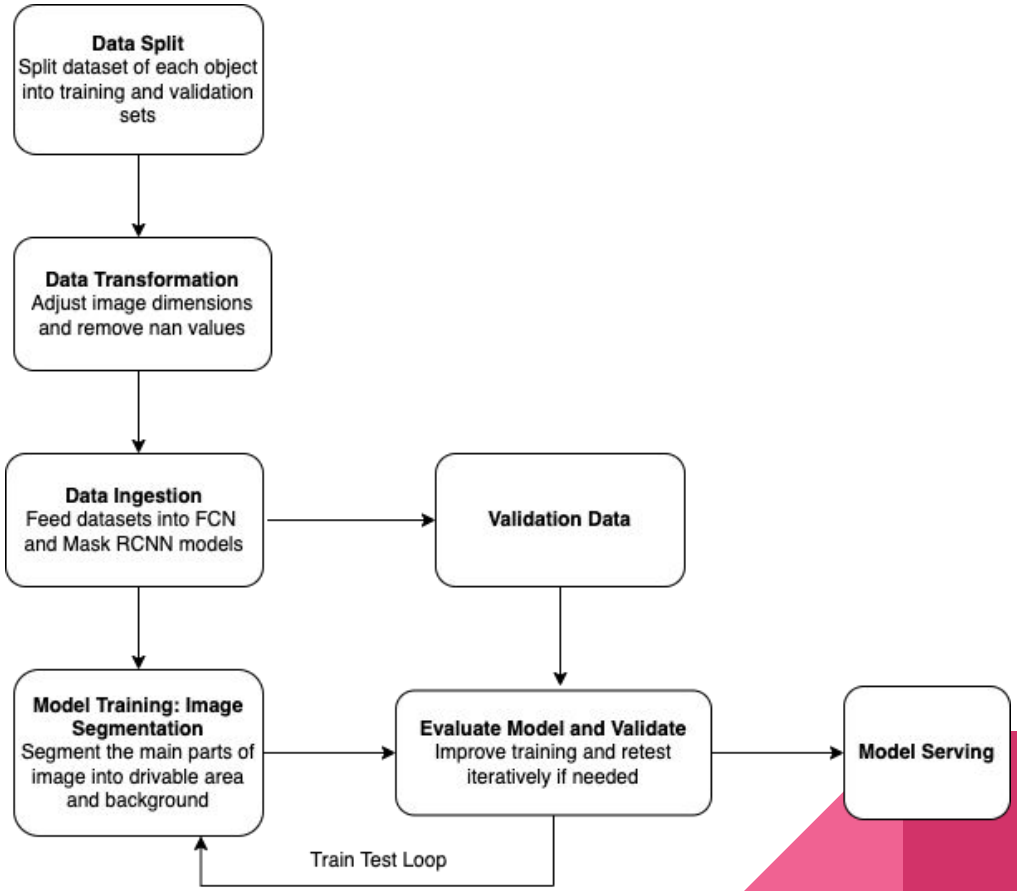
Model Attributes

- **Pixel values:** Used as an attribute to determine contrast between an object and background. Threshold value may be set to differentiate between the drivable area and background. Used for region-based segmentation.

- **Filtered values:** Kernel filters applied on pixel values can allow for edge detection.

- **Neighboring clusters:** Relationship between pixels and their neighboring pixel values.

Block Diagram



Dataset and Variables

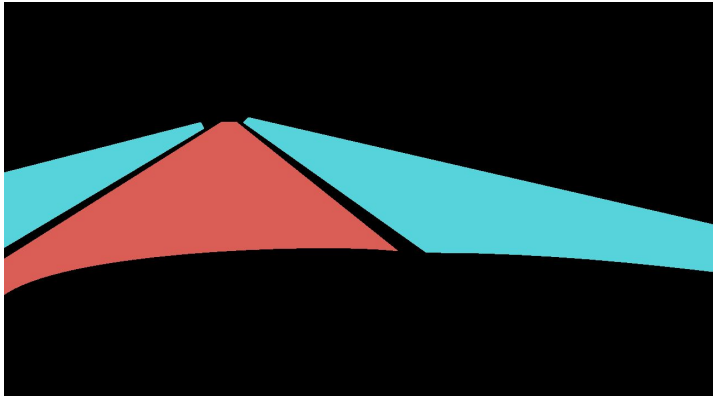
https://www.bdd100k.com/

Semantic Segmentation

Evaluation parameters

Image Segmentation

- Identify the drivable area and background
- Pixel segmentation scoring => accuracy score



Parameters

- Accuracy
- Precision
- Recall
- F1-Score

Success Criteria

- Accuracy: 50%

Experiments - FCN

Data ingestion

- Import as numpy arrays using the cv2 package
- Resize the image into the shape of (128,128) for easier FCN model ingestion

Data cleaning

- Imputed any nan values with the average of the other pixel array elements to avoid dimensionality reduction
- Ensured there were no infinite values in the data set
- To combat exploding gradients that resulted in a large update to network weights during training, gradient clipping is performed

Model training and tuning

- Limitations in experimenting different range of epochs and batch size
- Ddopted mini-batch gradient descent
- Tested two sets of parameters: 1) epoch = 5, batch_size = 100 and 2) epoch = 20, batch_size = 4

Evaluation

- Transform predicted and true mask into grayscale to read pixel value in 1 channel rather than 3
- Set threshold to classify into 0 or 1 (background vs drivable areas)
- Compare classification of each pixel between the true and predicted mask to get TP, TN, FP, FN
- Calculate evaluation metrics

Experiments - Mask-RCNN

Environment Set-Up

- Package installation
- Storage
- Memory Utilization

Data cleaning

- Ground truth mask vertices located outside of image size

Model development and tuning

- Mask R-CNN model's open source library
- Pre-trained weights based on COCO dataset (did not include road as a class)
- Hyperparameters:
  - Images per GPU
  - Number of Epochs
  - Steps per Epoch
  - Validation Steps
  - Detection Minimum Confidence

Evaluation

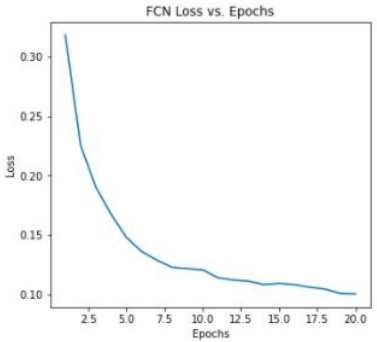
- Predicted mask output for each image was arrays of 1280x720xN shape, where N is number of distinct drivable areas (objects) detected by model
- Performed sum of predicted mask arrays across the N dimension to identify predicted drivable vs. background pixels
- Compared classification of each pixel between the true and predicted mask to get accuracy, precision, recall, and f1-score
- Calculated evaluation metrics (same as FCN)

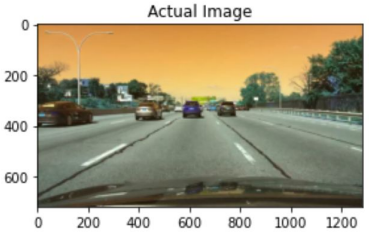
Results & Discussion

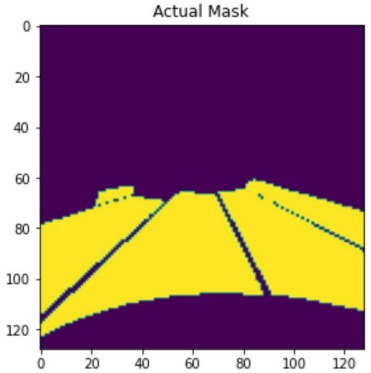
Parameter	FCN	Mask R-CNN
Mean Accuracy	0.3137	0.5174
Precision	0.0980	0.0166
Recall	0.0886	0.0002
F1-Score	0.0829	0.0004

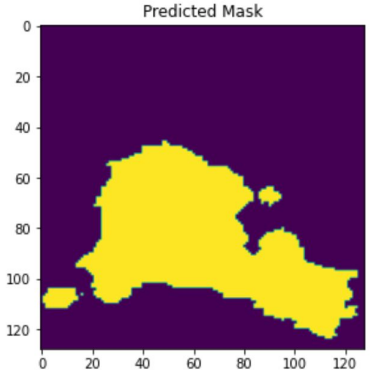
Test Images & Graphs

FCN:

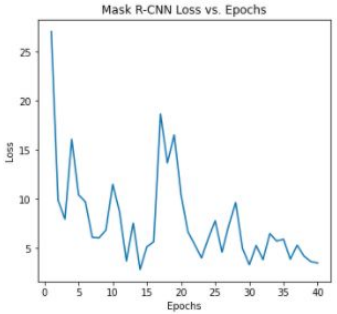


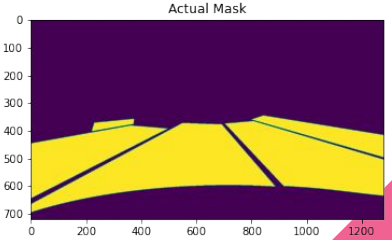


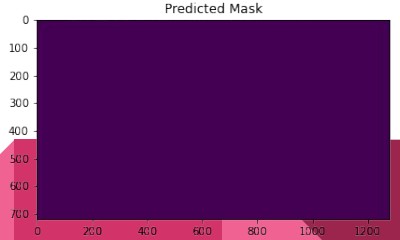




Mask R-CNN:







Standards

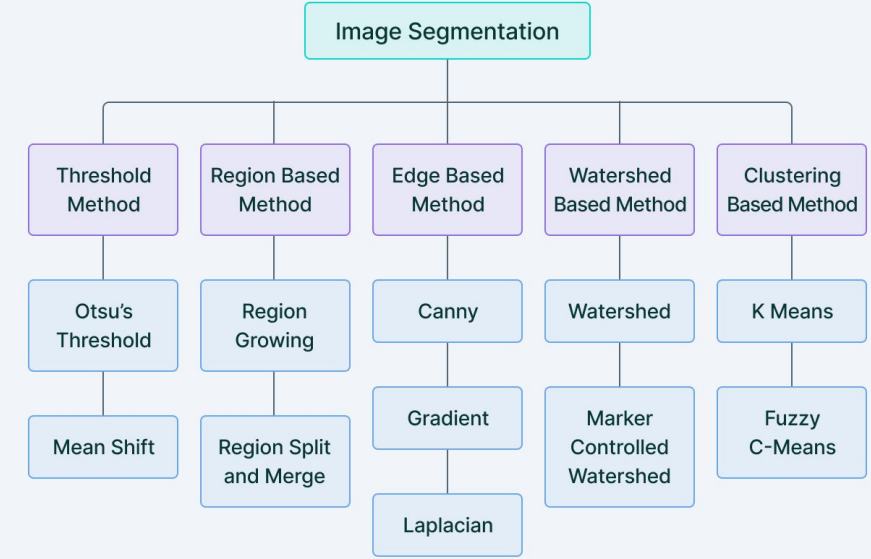
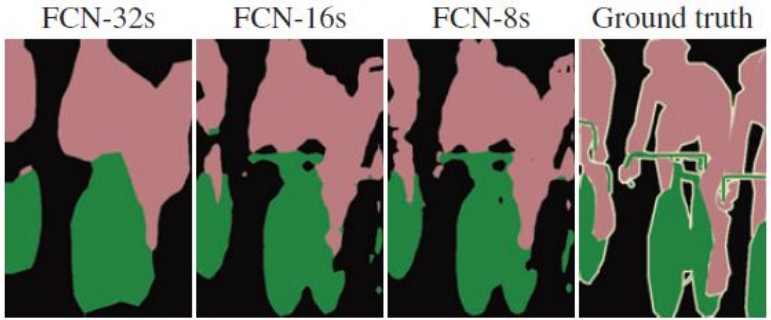


Image Segmentation
Threshold Method
Region Based Method
Edge Based Method
Watershed Based Method
Clustering Based Method
Otsu's Threshold
Region Growing
Canny
Watershed
K Means
Mean Shift
Region Split and Merge
Gradient
Marker Controlled Watershed
Fuzzy C-Means
Laplacian
V7 Labs

Model Comparison

FCN Intended Use Cases:

- Segment image between one or more classes + background



Mask R-CNN Intended Use Cases:

- Classify number of objects in image across two or more classes + background

Constraints

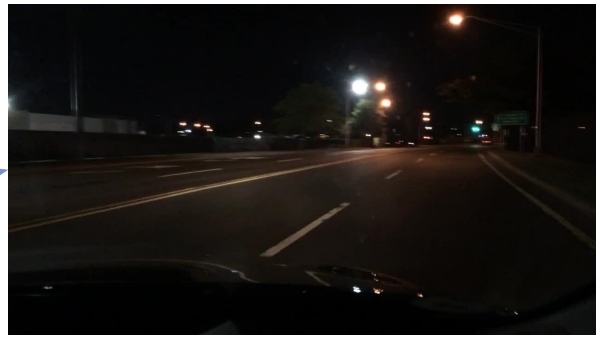
Hardware Constraints for Running Mask-R-CNN Model:

- Processing & memory limitations resulted in us quickly shifting to using Google Cloud Platform & NVIDIA GPUs

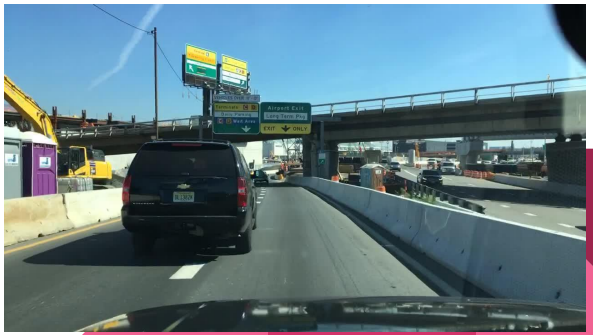
Limitations of Study

Filtered images on sunny, daytime, highway

Model would not be expected to perform well under these conditions



Images collected from SF/NY



Future Work

- General segmentation of the environment
- Detection of nearby objects
- Detection of lane markings
- Speed of surrounding vehicles

Further expansion

1. identify the region that is the opposing/incoming traffic vs. the direction that the car is driving
2. Expand to different weather and road conditions