Pavement Marking Fadeness Detection

Columbia Data Science Capstone project

By Moya Zhu, Zihao Zhang, Megan Zhou, Jingfei fang, Ran Pan
(Advised by Adam K.)

Introduction

Project goal

Detect the **fading** of the **pavement markings**

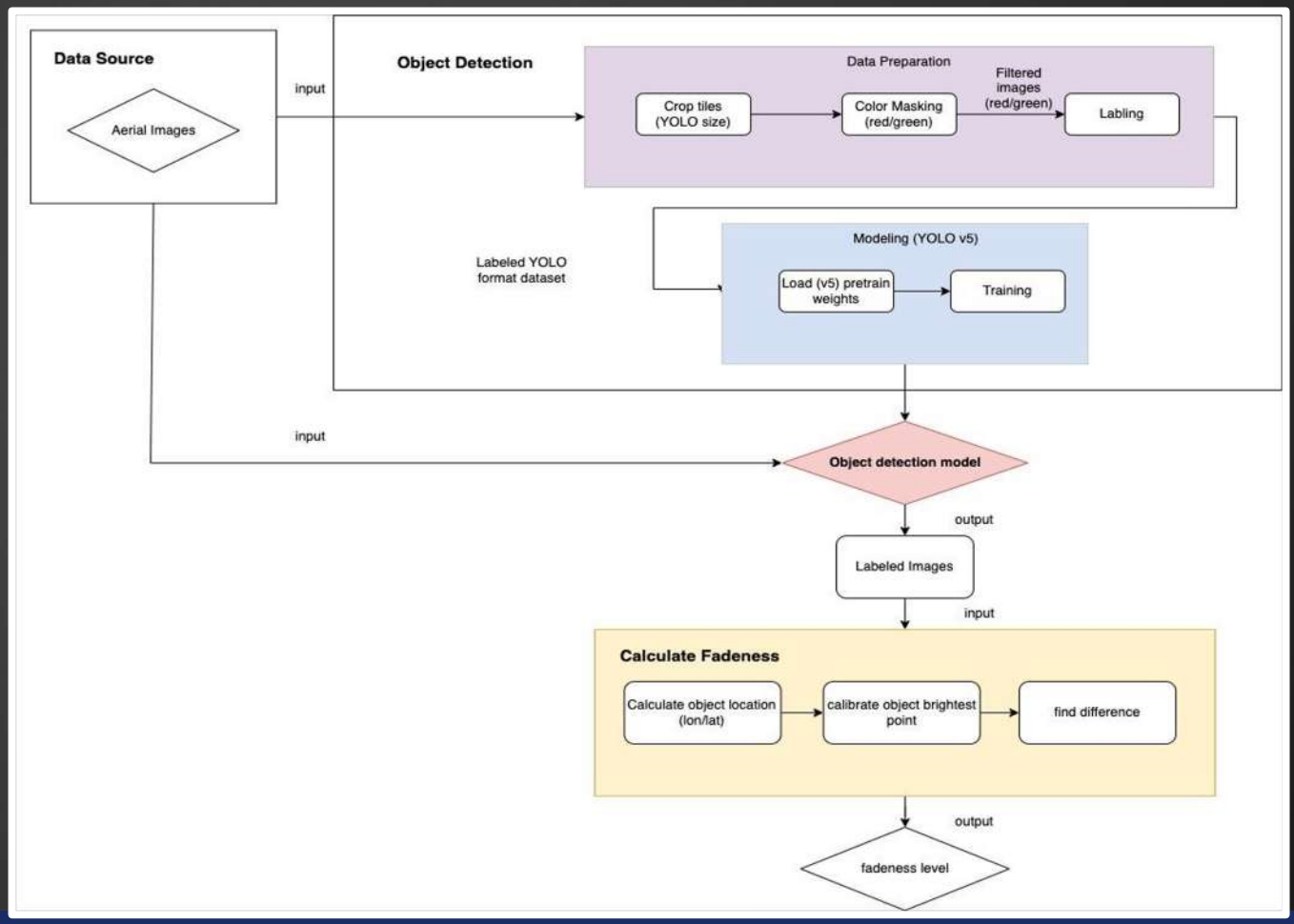
- Bike lane

- Bus lane

- Crosswalks

Crosswalk fadeness example:

Workflow


Data Source
Aerial Images
input
Object Detection
Data Preparation
Crop tiles (YOLO size)
Color Masking (red/green)
Filtered images (red/green)
Labling
Modeling (YOLO v5)
Load (v5) pretrain weights
Training
Labeled YOLO format dataset
input
Object detection model
output
Labeled Images
input
Calculate Fadeness
Calculate object location (lon/lat)
calibrate object brightest point
find difference
output
fadeness level

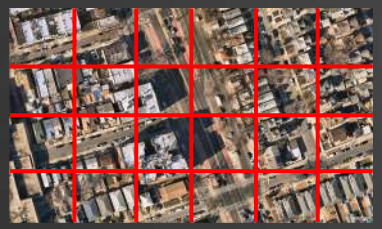

Object Detection

Steps

1. Image Crop
2. Image Labeling
3. Model Selection
4. Model Training

Crop & Labeling

Example from one image



(size 2137*3637)

crop

24 640*640 tiles

label

(output in YOLO form)

YOLO

You Only Look Once (YOLO)

- Train faster with same accuracy
- Support Real Time detection
- Small network - can be deployed easily

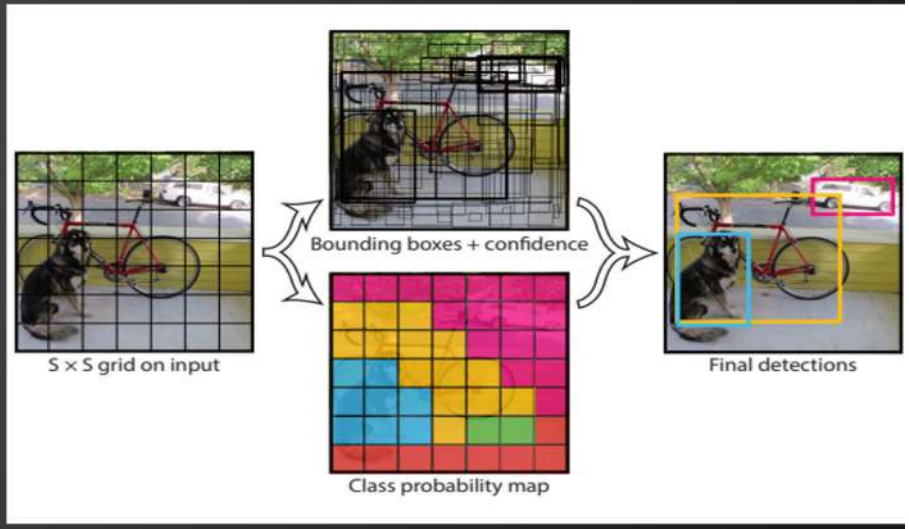


Image Source: https://doi.org/10.48550/arXiv.1506.02640

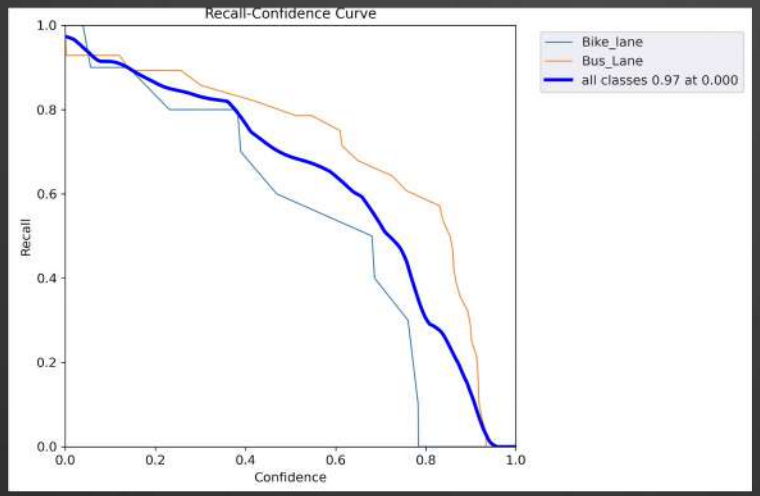
Training (bus & bike lane)

- Data: 175 Labeled bus/bike lane cropped images
- 8:2 train validation data split
- YOLO v5 default pretrained weight
- 100 epochs with batch size = 10

Training (bus & bike lane)

Accuracy:

- Bike Lane: 0.85 precision, 0.8 recall
- Bus Lane: 0.92 precision 0.84 recall

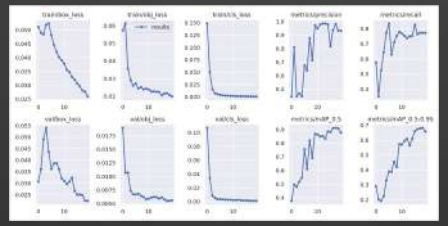


Training (crosswalk)

1. Data Source: Korean dataset https://open.selectstar.ai/data-set/wesee (100+GB, but only used a small portion)
2. Data Augmentation (with Roboflow) (for both training and validation)

Bounding Box: Rotation: Between -24° and +24°

1. Result:



Coordinate

Original image
991966.619472 , 185356.731421
NYC plain coordinate

crop

output in YOLO
for each bounding box:
abscissa, ordinate,
box with, box height
(in this cropped image)

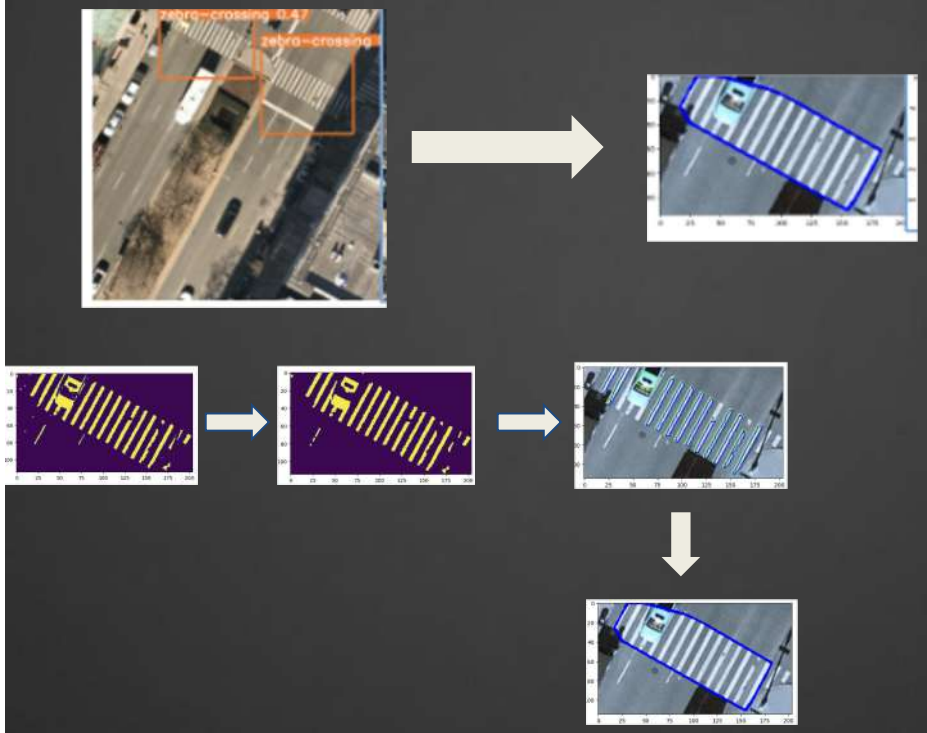
for each bounding box:
output abscissa, ordinate
(in NYC plain coordinate)

transform

coordinate.py
input: aerial image, related .jgw file
output: a file folder, for all crosswalk and corresponding coordinates


Segmentation Approach

Convex hull around crosswalk



Procedures:

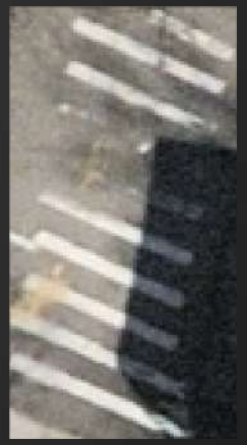
- Denoise using morph_open in opencv
- Contour and minimum_area_rectangle
- Filter using aspect ratio of rectangles
- Draw convex hull

Fadeness Calculation

Pixel Value Based Fadedness Score

- Based on Convex hull segmentation, locate the pixel that belong to the crosswalks
- Perform pixel values calibration
- Calculate the fadedness score based on the mean of pixel values
- Calculate the percentage fadedness score based on the threshold value

98.7%



87.5%

Future Work

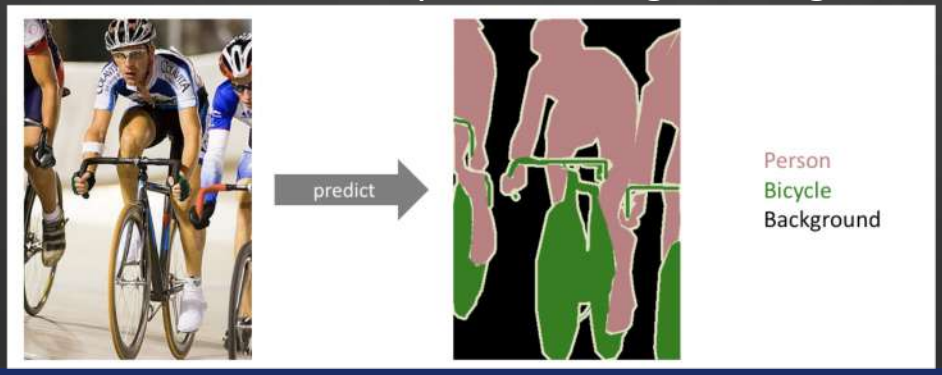
Future Work

Improve current:

- Add more training data, fine tune model
- Calculate fadeness with two image in different time (superimpose)

Alternative:

- Train semantic segmentation model instead of object recognition
  - More complex masking labeling and modeling



Question?