



DIVE INTO CODE

Traffic sign detection on real Vietnamese streets



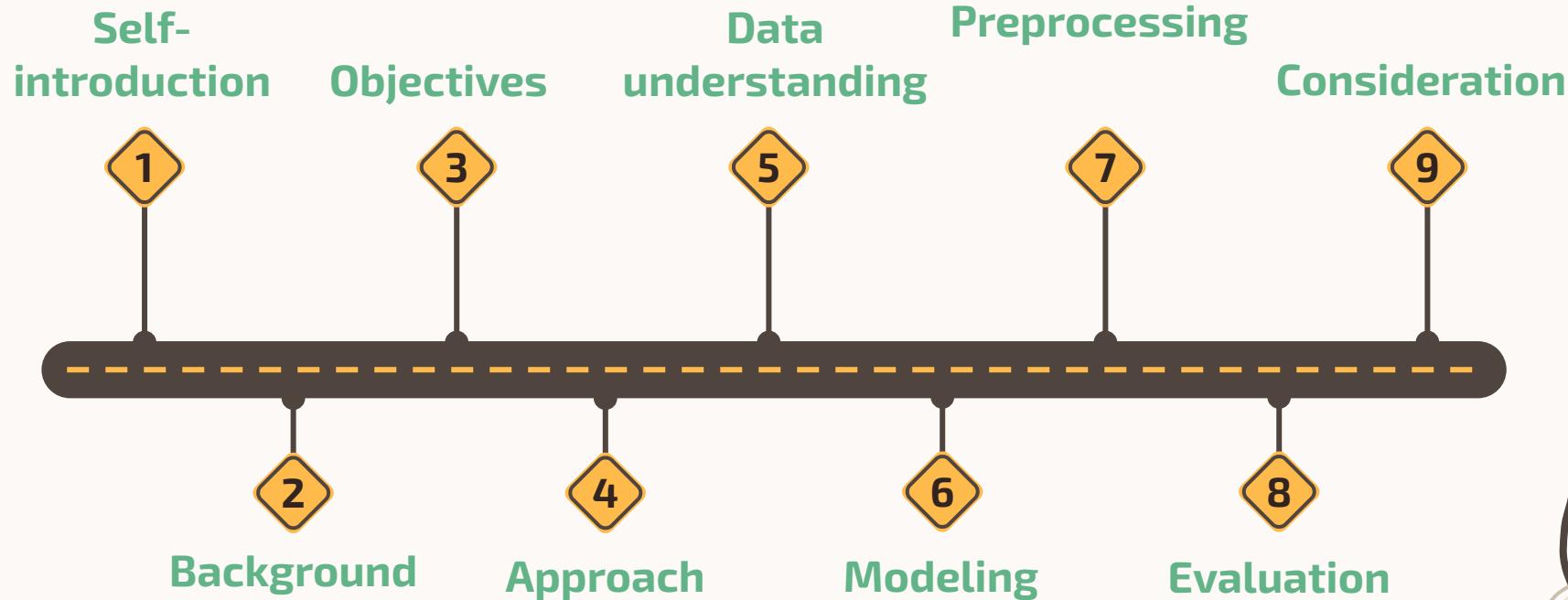
mef2010 - Ho Hoang Thien Long





DIVE INTO CODE

TABLE OF CONTENTS





01 Self-introduction

Hometown Dalat City, Vietnam.

Academic Final year student in Computer Engineering.

Hobby Engineering, gadget, DIY, have fun with friends.



**HO HOANG
THIEN LONG**



DIVE INTO CODE

02

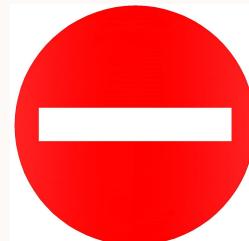
Background



DIVE INTO CODE

Traffic sign exists to:

- guide and protect drivers.
- Control traffic flow.





DIVE INTO CODE

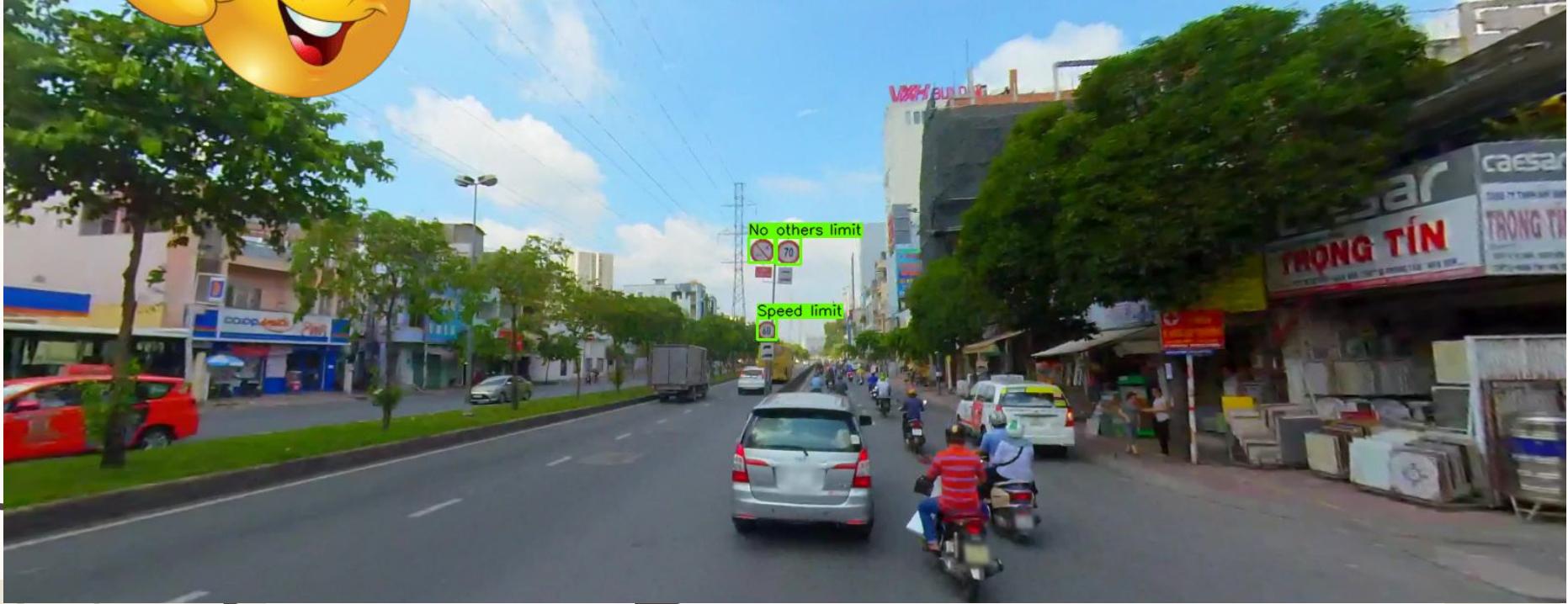




DIVE INTO CODE



Propose a solution!





DIVE INTO CODE

03

Objectives



DIVE INTO CODE



Traffic sign detection



Real-time detection

≥ 24 FPS



$\geq 70\%$ mAP@0.5
score on valid set



DIVE INTO CODE

04

Approach



DIVE INTO CODE

Dataset?

- 1 Use existed dataset from **Zalo AI challenge 2020**.

Model?

- 2 Choose **YOLOv4 - darknet**, a highly accurate and fast image detection model





DIVE INTO CODE

05

Data understanding

(data_analysis.ipynb)



File structure

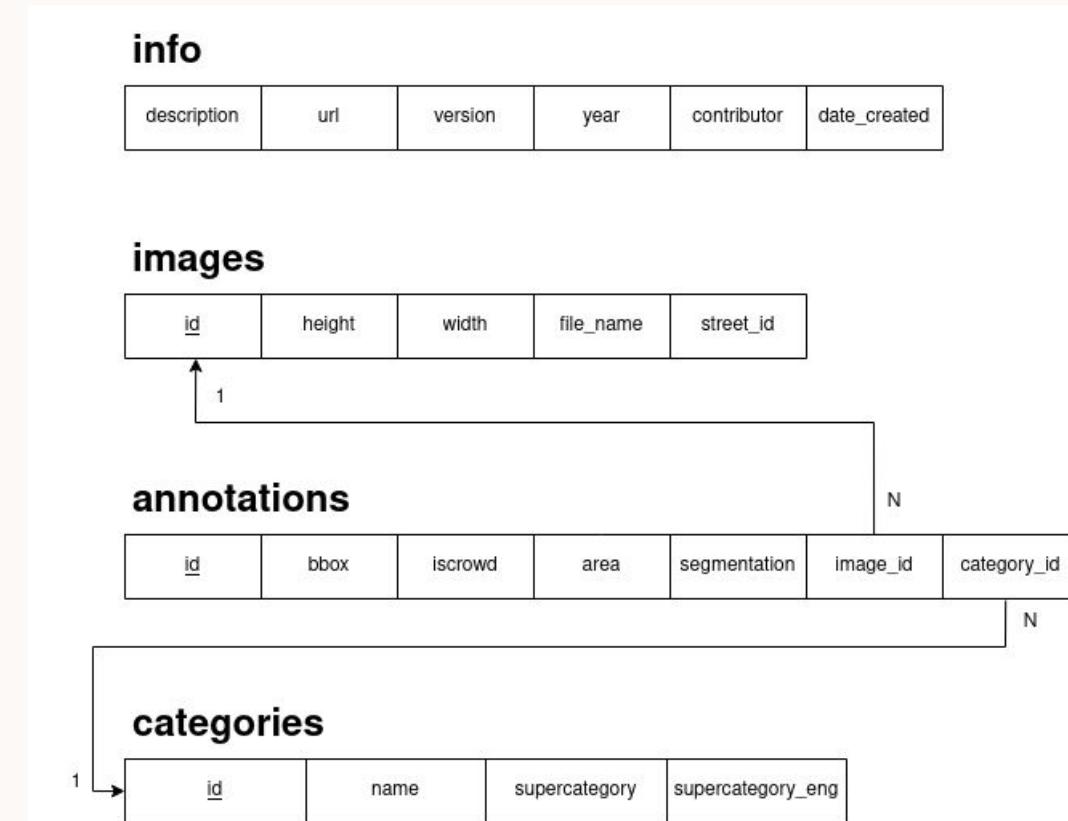




train_annotations.json

COCO format

No NULL values





categories

1 - No entry



2 - No stopping and parking



3 - No turning



4 - Speed limit



5 - No others



HẠN CHẾ TƯỞNG LƯỢNG XE
HẠN CHẾ TƯỞNG TRÊN TRỤC XE
HẠN CHẾ CHIỀU CAO

6 - Danger



GIAO NHẬU VỚI ĐƯỜNG VÙI TÚI
GIAO NHẬU CÙ TÌNH HỆ KHÍ
GIAO NHẬU SƠ CÙ KHOA CHÂN

7 - Mandatory



301a

CÁC XE CHỈ ĐƯỢC DI THẮNG



301b

CÁC XE CHỈ ĐƯỢC RẼ PHẢI



301c

CÁC XE CHỈ ĐƯỢC RẼ TRÁI



302b

LƯỜNG ĐI VÒNG CHỐNG NGƯỜI VẮT SANG TRÁI



303

NƠI GIAO NHẬU CHẠY THEO VÒNG XUYÊN



304

ĐƯỜNG ĐÁNH XE THỔ SỐ



images

There are **4500** training images taken across Vietnamese streets.

1622 px





annotations

There are total **11000** bounding boxes (bboxes).





DIVE INTO CODE

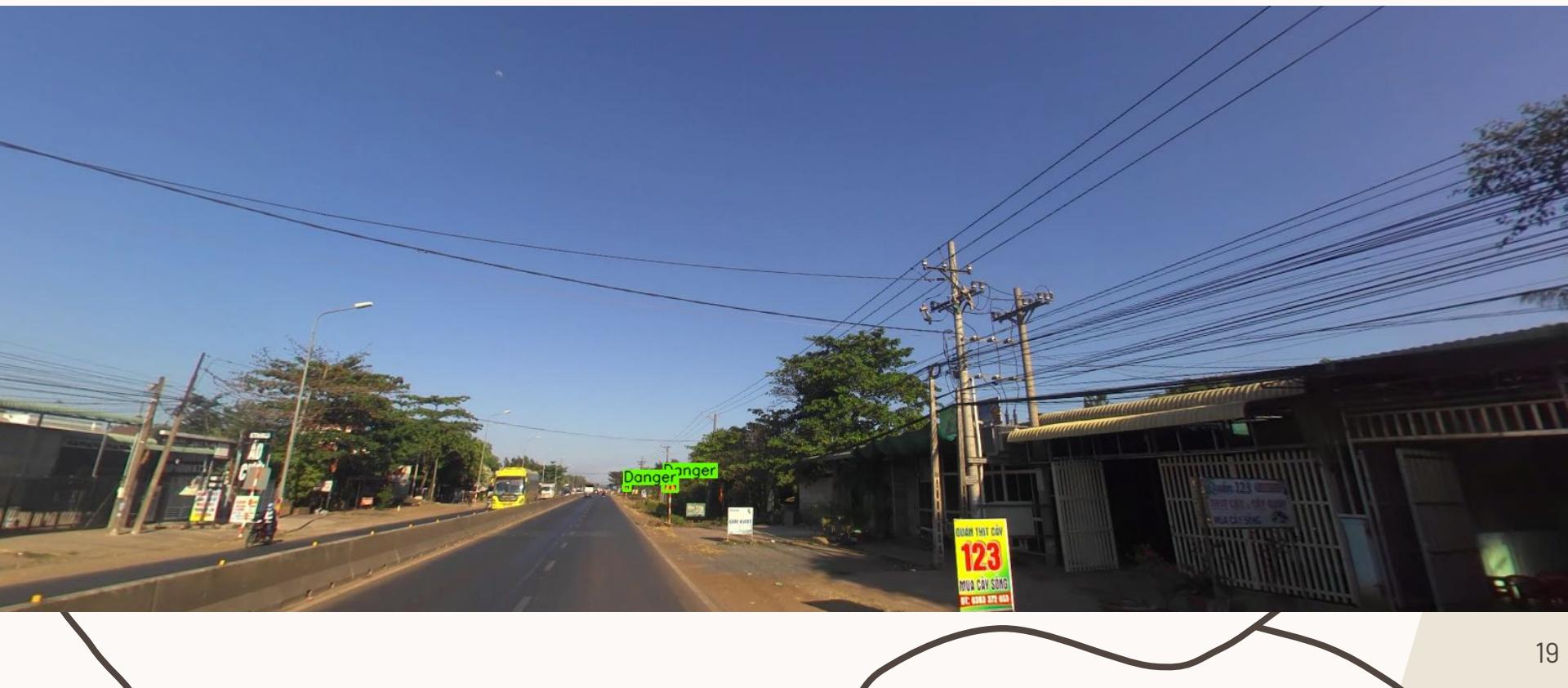
annotations





DIVE INTO CODE

annotations



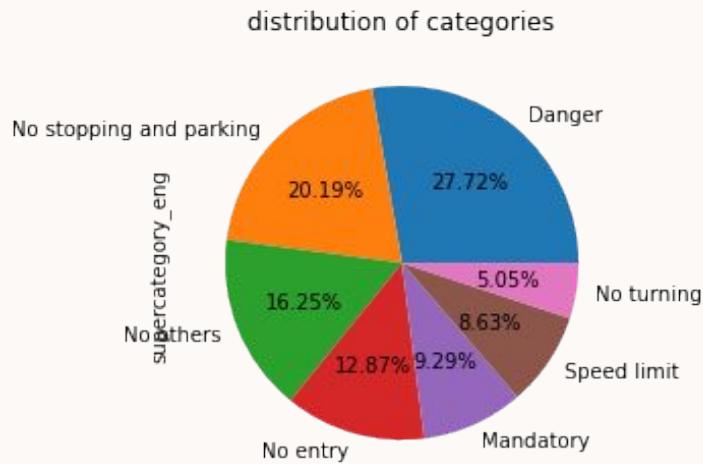


annotations



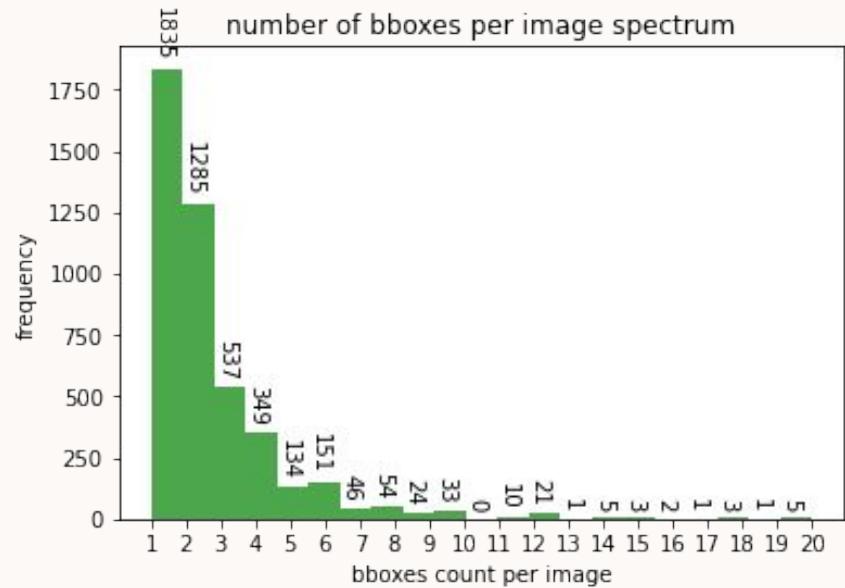


annotations



subcategory_eng

Danger account for the most.



average **2 - 3** bboxes/image.



DIVE INTO CODE

06

Modeling

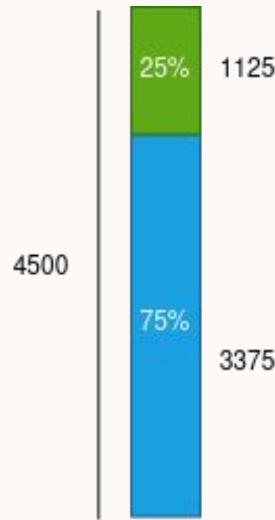
(`prepare_data.ipynb` &
`prepare_model.ipynb` &
`train.ipynb`)



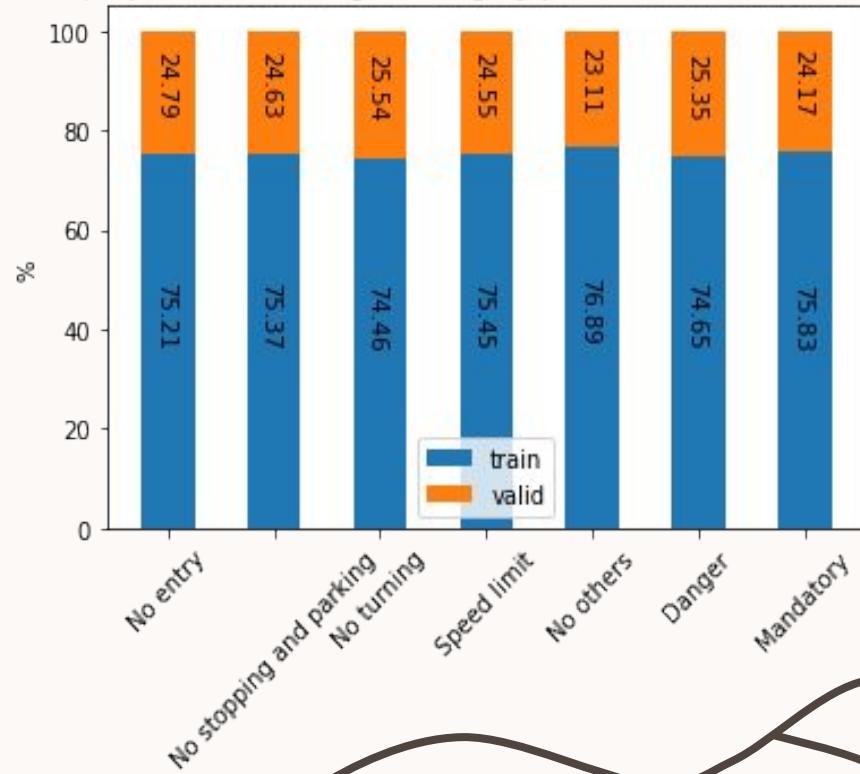
Train test split

test_size = 0.25

Images proportions



bboxes proportion according to category presented as a stacked bar chart





DIVE INTO CODE

Augmentate training images

ShiftScaleRotate

6014-augmented-0.png



Resize

1382-augmented-0.png



MotionBlur, MultiplicativeNoise, RandomBrightnessContrast

5634-augmented-0.png



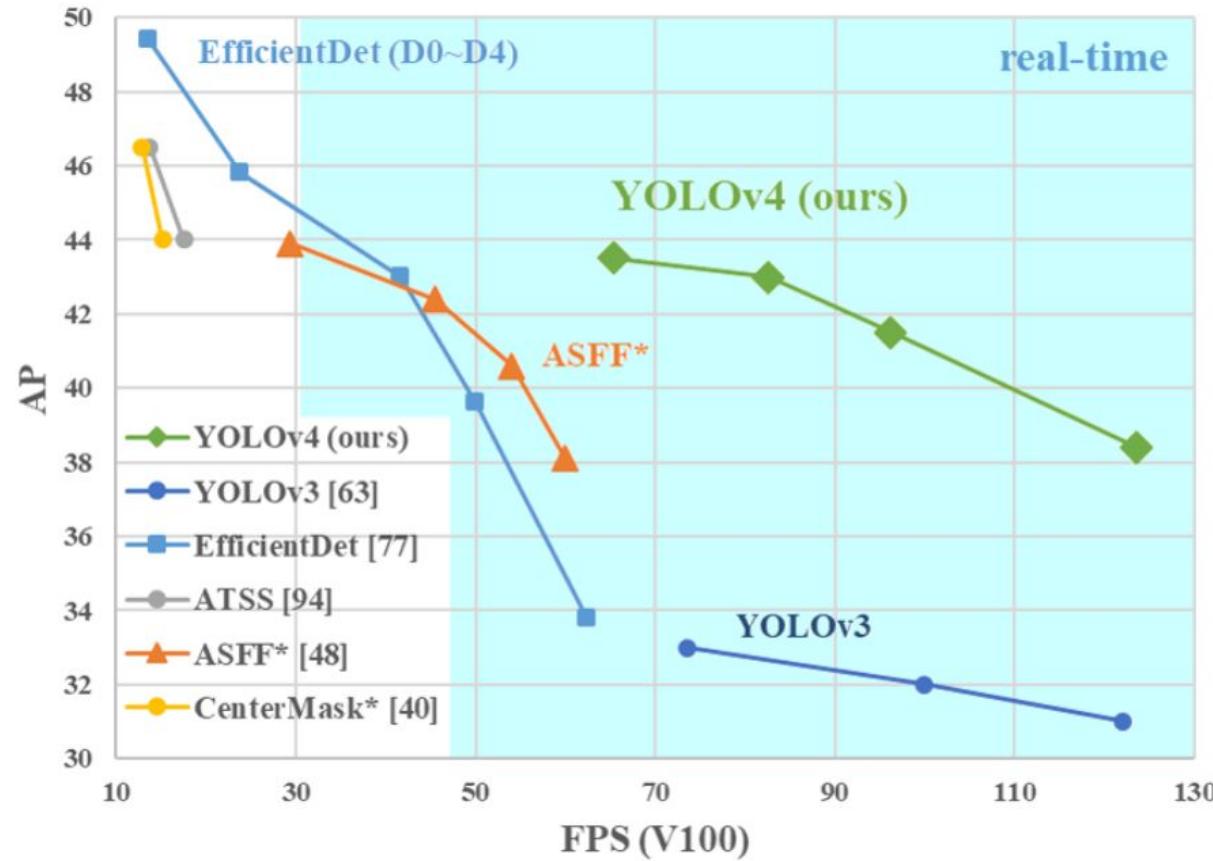


DIVE INTO CODE

YOLOv4

<https://arxiv.org/pdf/2004.10934v1.pdf>

MS COCO Object Detection

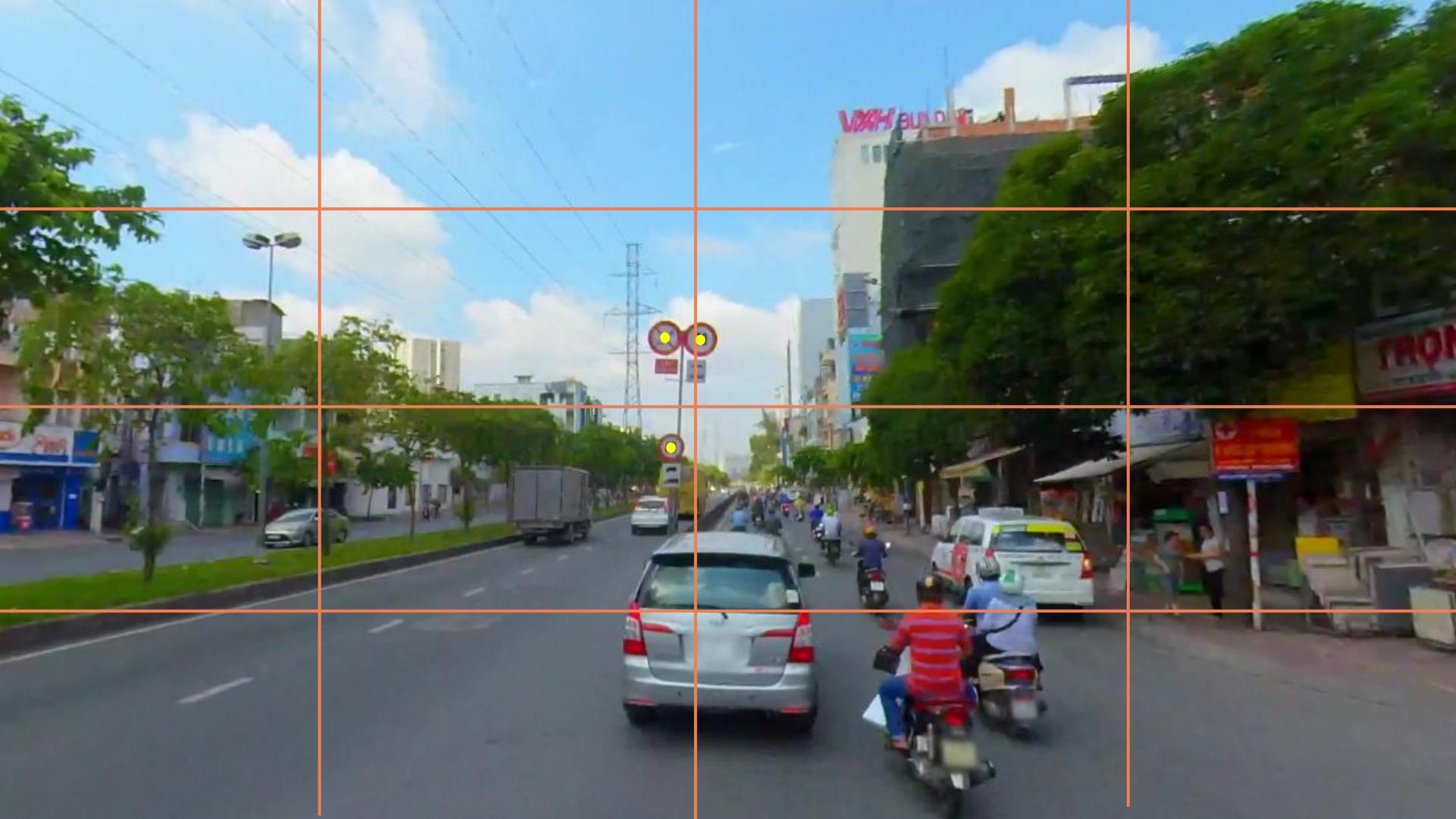


No others limit



Speed limit





YOLOv4

Added more state-of-the-art features to YOLOv3

- Grid Weighted-Residual-Connection.
- Mish-activation.
- Mosaic data augmentation.
- Apply both one-stage and two-stage detection.
- ...



Loss function:

$$L_{total} = L_{classification} + L_{localization} + L_{confidence}$$



DIVE INTO CODE

Darknet

Open source neural network in C
<https://pjreddie.com/darknet/>

- GPU computing.
- Parallel computing backend.





Training

- $\text{Max_batches} = (7[\text{classes}] * 2000) = 14000$
- $\text{batch}=64$

Using GPU Tesla T4 of Google Colab

- 10 sec/batch

=> Total training time = $10 * 14000 = 140000 \text{ sec} = 39 \text{ hours}$

Using YOLOv4 config files

- yolo4.conv.137
- yolov4-custom.cfg



Convert to YOLO annotations

123.txt (<class idx> <x_center> <y_center> <w> <h>)

```
0 0.49642371469312846 0.5248235045826511 0.004581956116896904 0.009782892547647248  
6 0.49710206379130606 0.533893360079413 0.004775770144947688 0.012374279832436342
```

Data

- images
 - 123.png
 - 123.txt
- train.txt
- valid.txt
- classes.names
- classes.data

Train.txt
(path to images)

```
data/images/1558.png  
data/images/1824.png  
data/images/4914.png  
data/images/6087.png  
data/images/4842.png
```

valid.txt

```
data/images/11235.png  
data/images/5001.png  
data/images/5101.png  
data/images/11918.png  
data/images/11508.png
```

classes.names

```
No_entry  
No_stopping_and_parking  
No_turning  
Speed_limit  
No_others  
Danger  
Mandatory
```

classes.data

```
classes = 7  
train = data/train.txt  
valid = data/valid.txt  
names = data/classes.names  
backup = backup
```



DIVE INTO CODE

08

Evaluation (test.ipynb)



DIVE INTO CODE

Before evaluate...

- Change input image bigger: $[608 \times 608] \rightarrow [1024 \times 1024]$
=> less feature losses -> better score.
- Get the best weights at 8100 iteration.



DIVE INTO CODE



ground truth



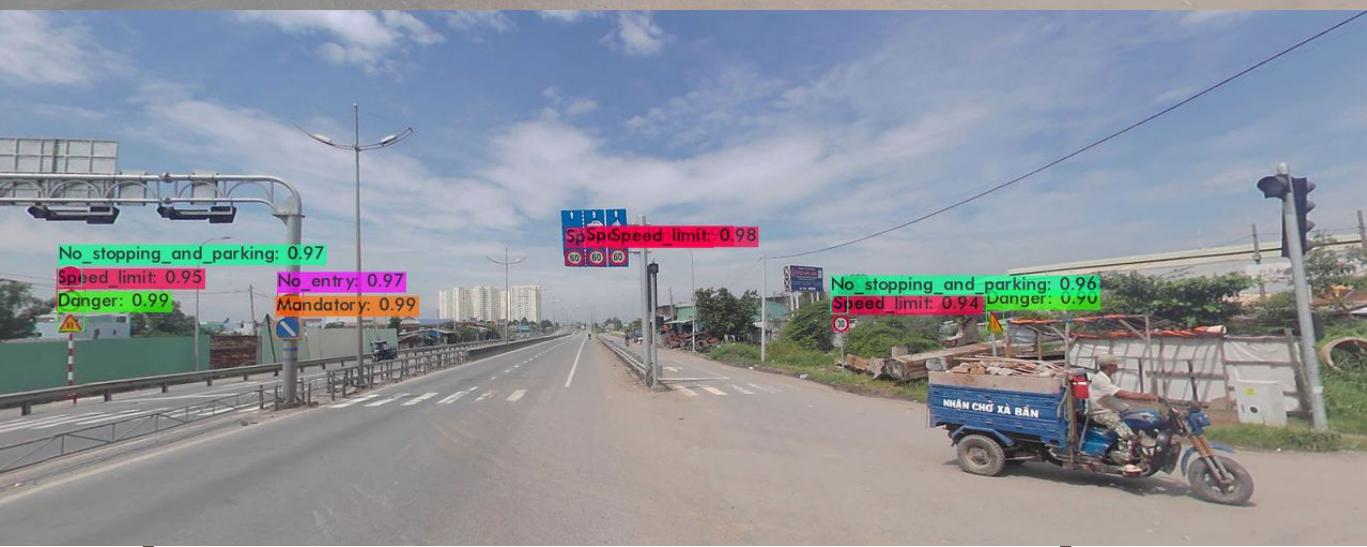
prediction



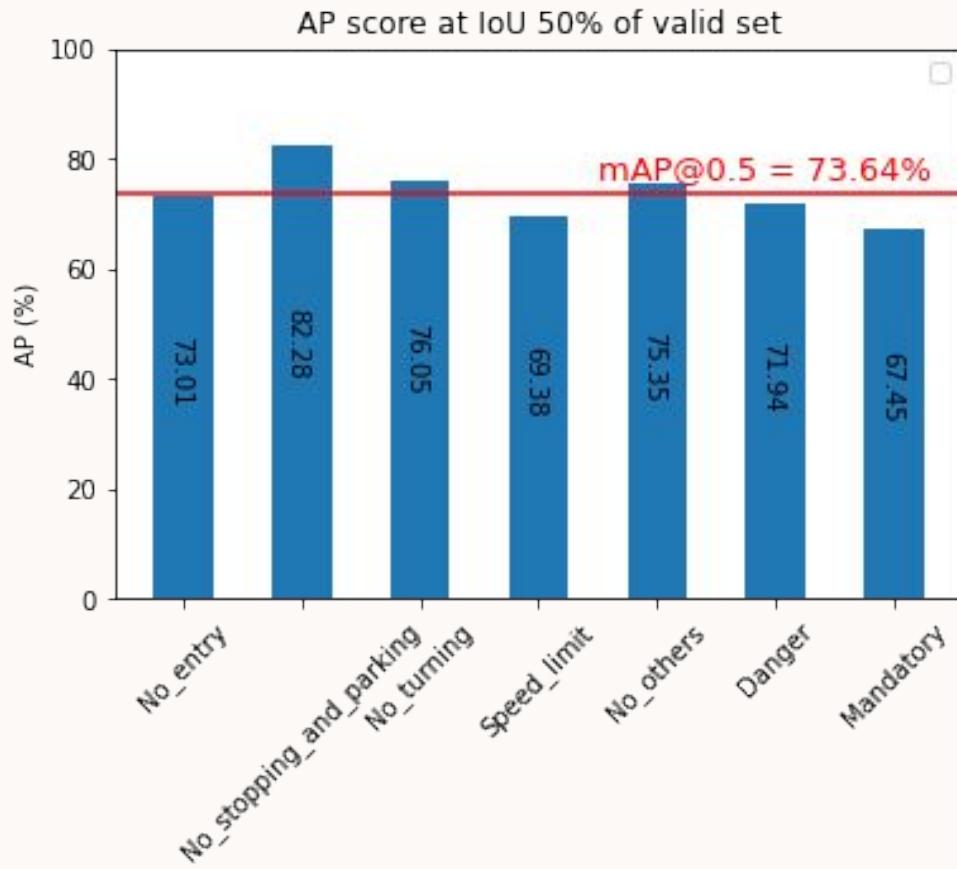
DIVE INTO CODE



ground truth



prediction



mAP score



Summary

	608 x 608	832 x 832	1024 x 1024	1120 x 1120
mAP@50 (%)	68.78	72.29	73.64	72.97
Processing time (ms/image)	50	51	76	86
FPS	20	19	13	12

=> More than [1024 x 1024]: lower score, higher processing time.



DIVE INTO CODE



original

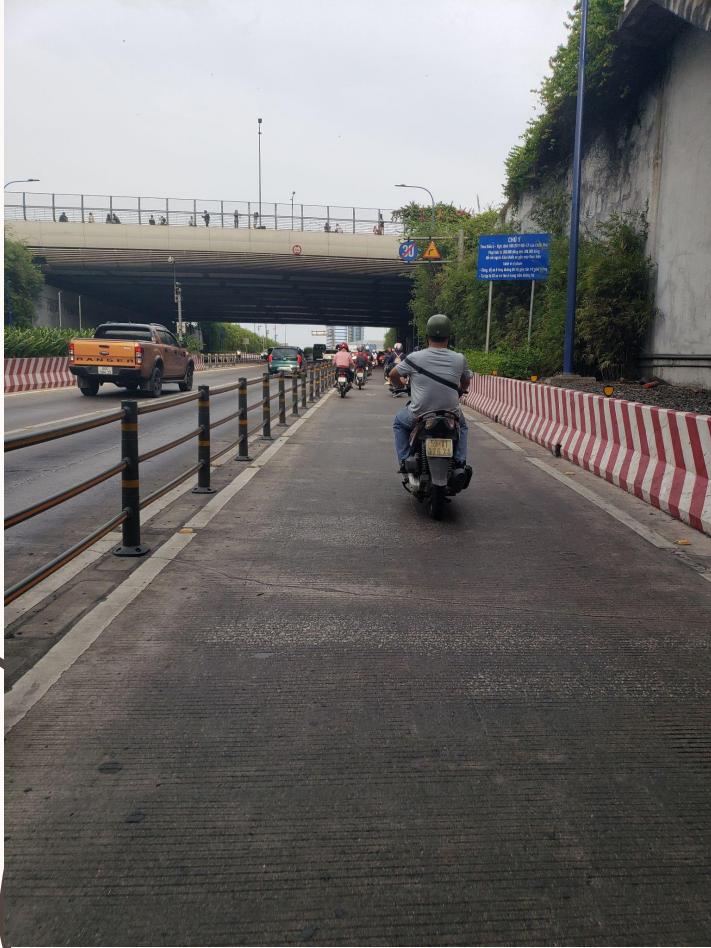


Real world
testing!

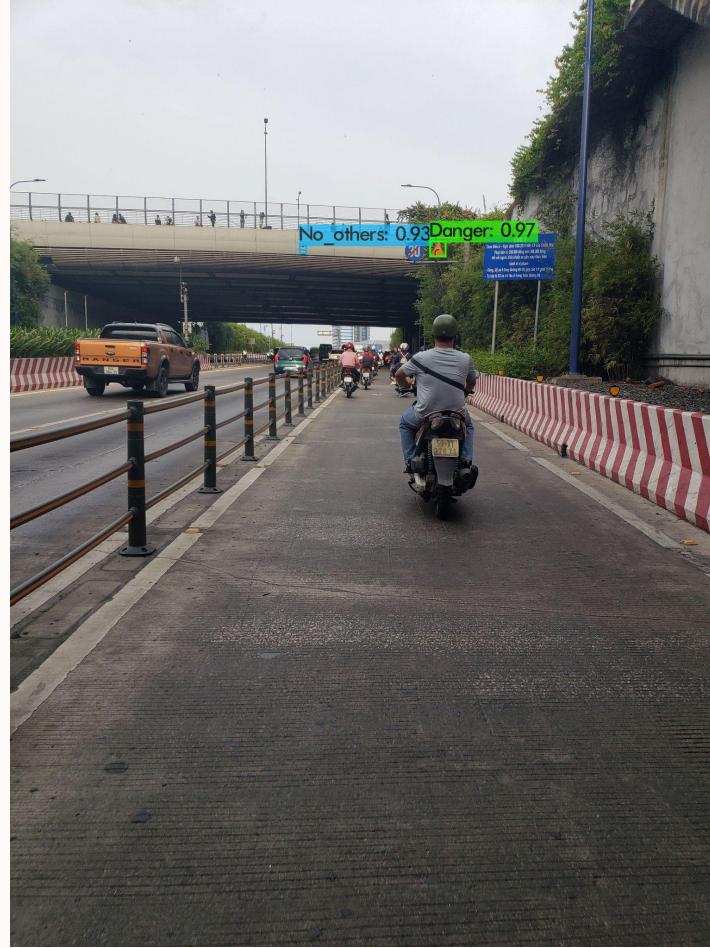
prediction



DIVE INTO CODE



original



prediction



DIVE INTO CODE



original



prediction



DIVE INTO CODE



original



prediction



DIVE INTO CODE



original



prediction



DIVE INTO CODE

09

Consideration



Consideration

- YOLOv4 meets 2/3 of my proposed objectives.
 - ✓ Can detect traffic sign in Vietnamese street
 - ✓ > 70% mAP@0.5
 - ✗ 24 FPS
- In the near future...
 - Try with video, webcam for real-time detection.
 - Try more methods of augmentation to detect more cases.



DIVE INTO CODE

Q&A!

Thank you for listening!