IMAGE RECOGNITION WITH PYTHON

Computer Education

LECTURER

KETT KONGNAKORN

4th year Student

LECTURER

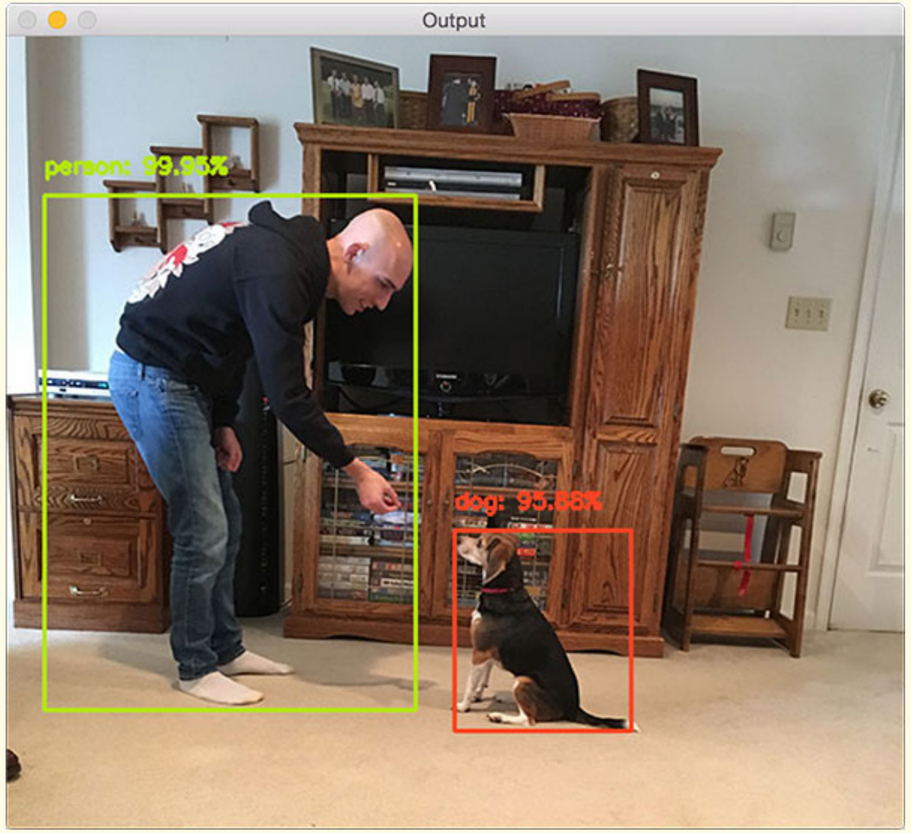
PIYADET KITPHADUNG

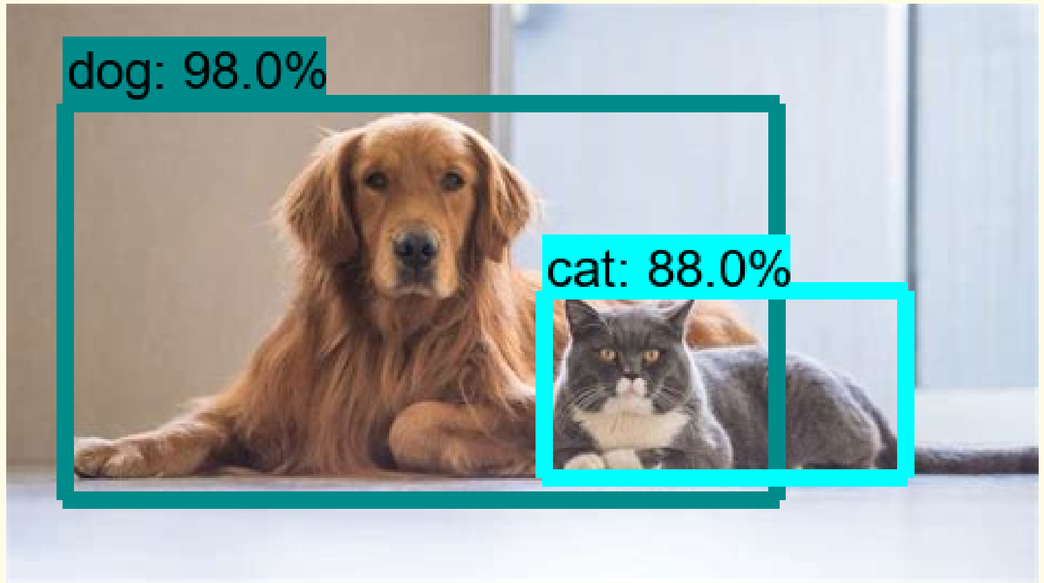
4th year Student

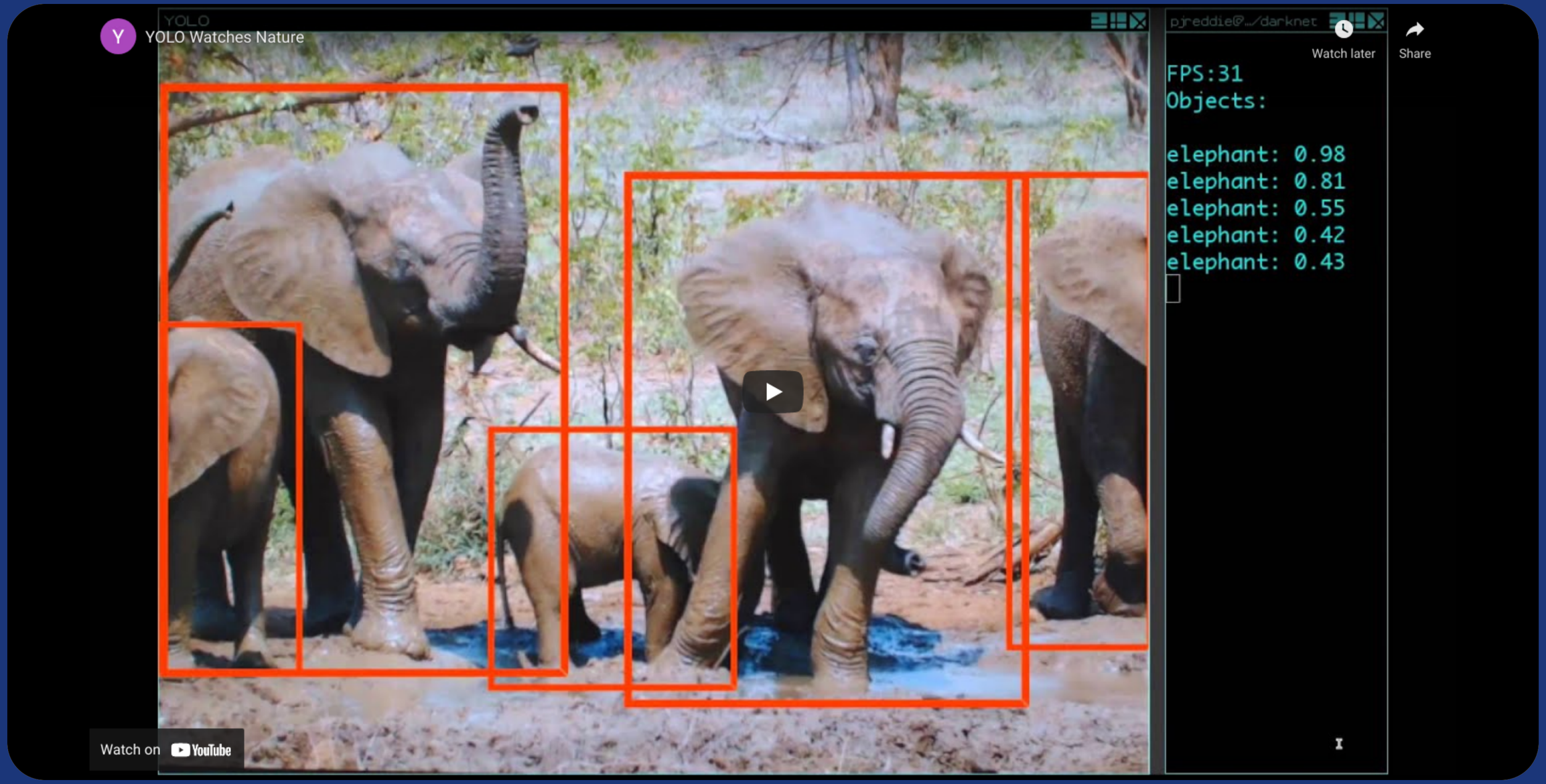
Object Detection

Object Classification VS Object Detection

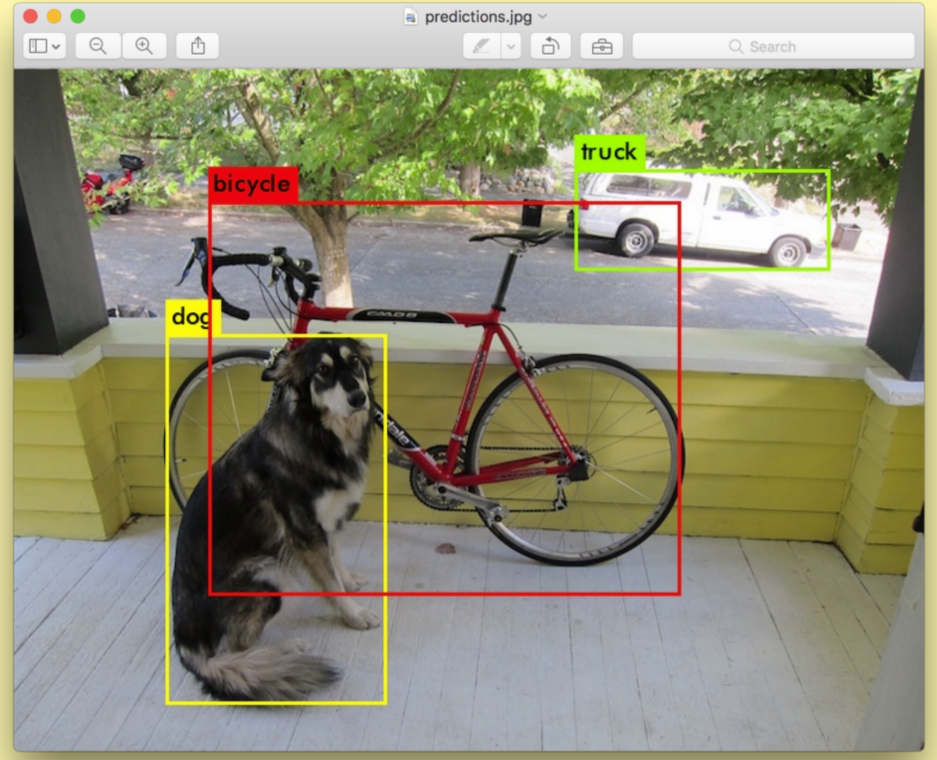
What is YOLO ?


Output
person: 99.95%
dog: 95.88%

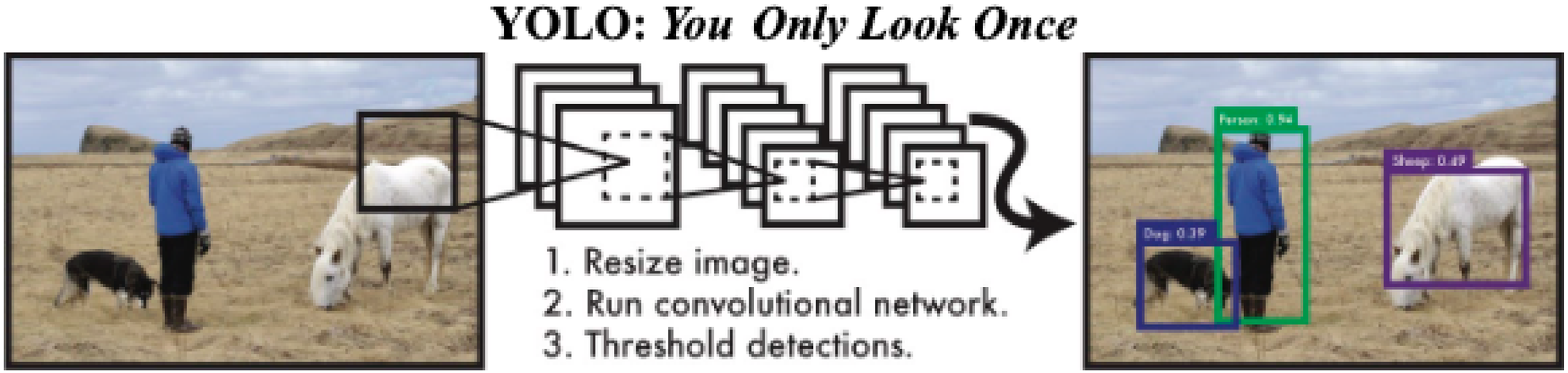

dog: 98.0%
cat: 88.0%


YOLO Watches Nature
Watch later
Share
YOLO
pjreddie@.../darknet
FPS:31
Objects:
elephant: 0.98
elephant: 0.81
elephant: 0.55
elephant: 0.42
elephant: 0.43
Watch on YouTube

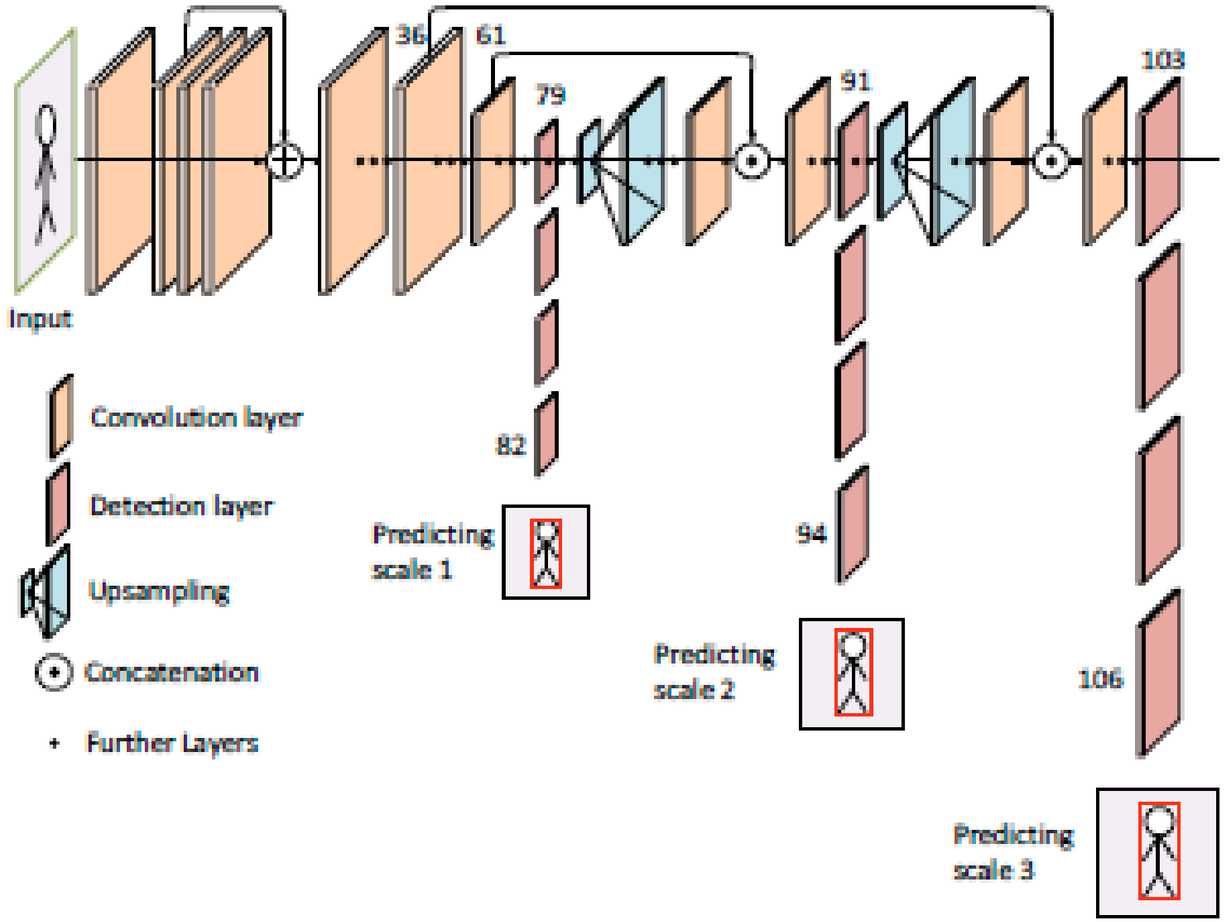
YOLO เป็นการรู้จำวัตถุที่ใช้วิธีการที่เรียกว่า “Fast single-shot detection” ซึ่งจะเป็นวิธีการที่สามารถตรวจจับวัตถุได้จากการส่งผ่านรูปภาพเข้าไปในระบบเพียงครั้งเดียว



With YOLO, you only look once at an image to perform detection

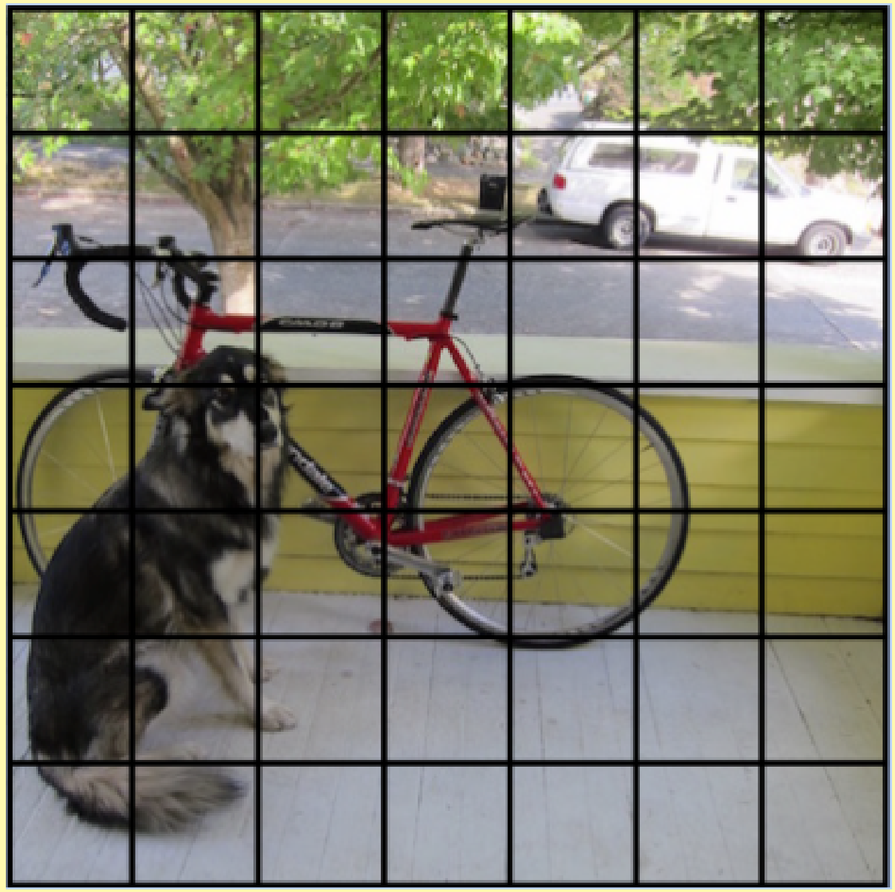


โครงสร้าง Yolov3



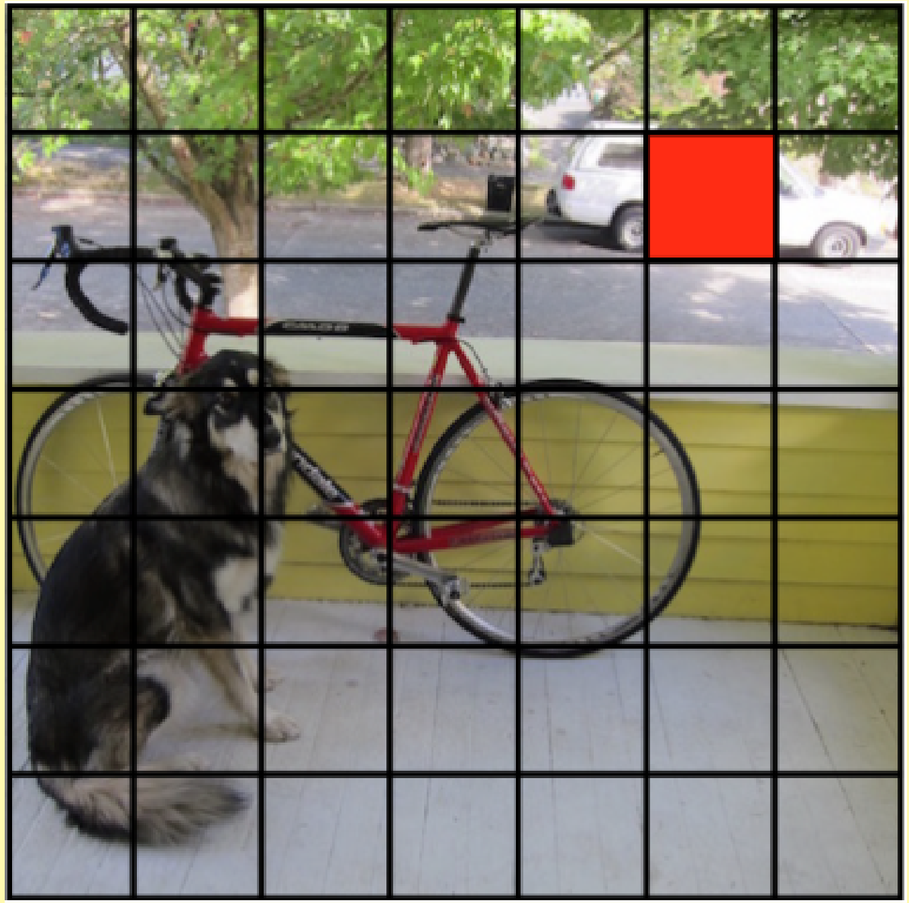
สักษณะการทำงานแบ่งเป็นชั้น ๆ
ชั้นของคอนโวลูชั่น
ชั้นของการเพิ่มอัตราสุ่ม
ชั้นของการทำนายผล

INPUT

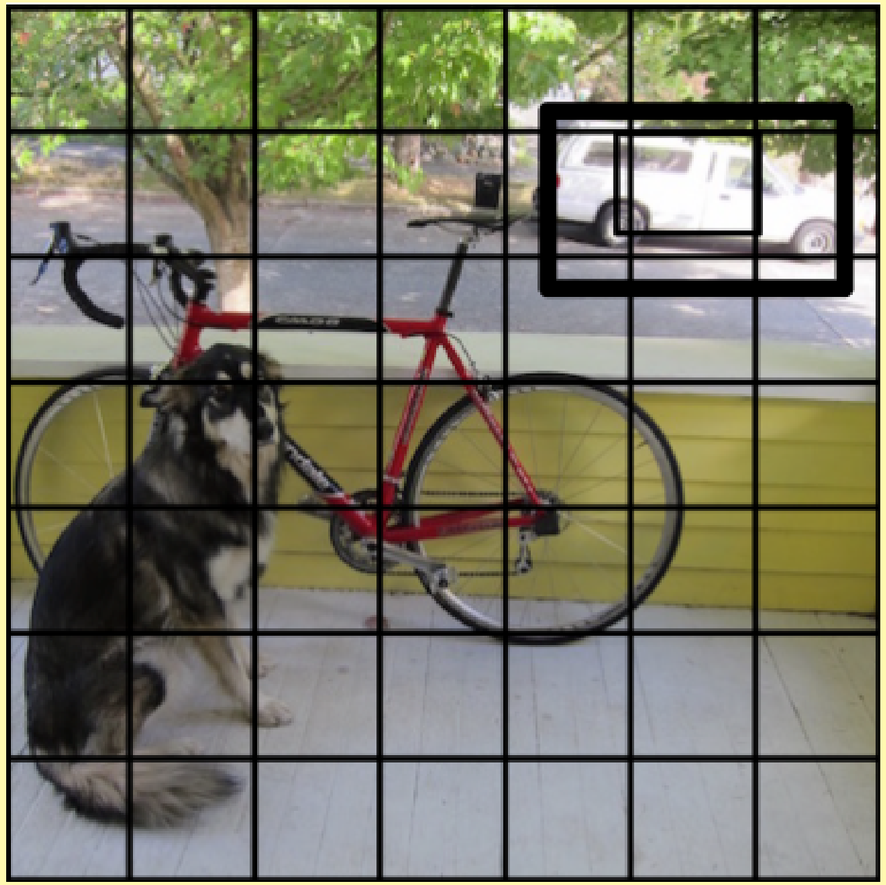


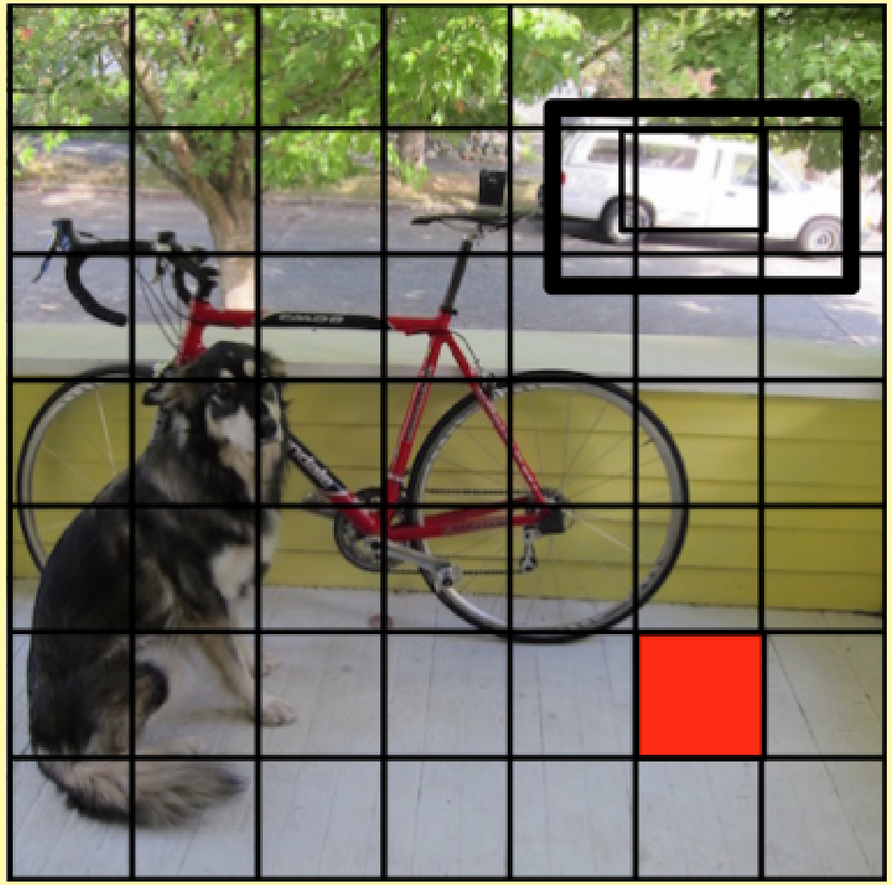
แบ่งภาพเป็นสี่เหลี่ยมตาราง

ในแต่ละตารางที่แบ่งออกมาจะทำการทำนายค่า **boxes** และค่า **confidences : P(Object)**

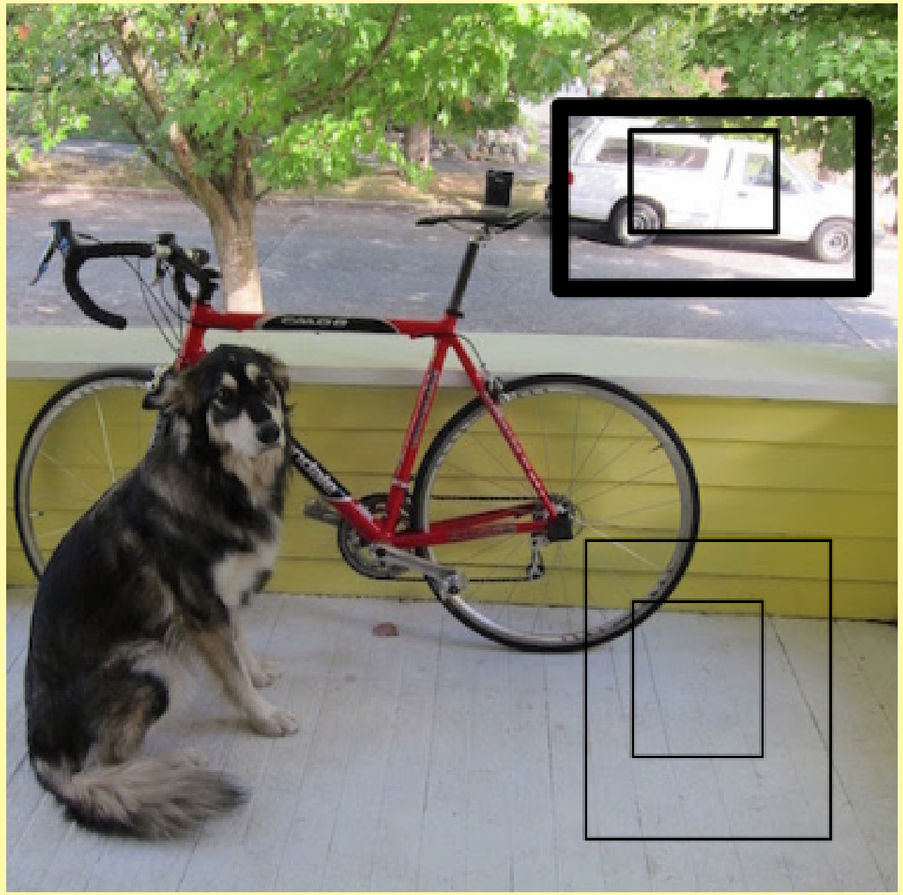


ในแต่ละตารางที่แบ่งออกมาจะทำการทำนายค่า **boxes** และค่า **confidences : P(Object)**

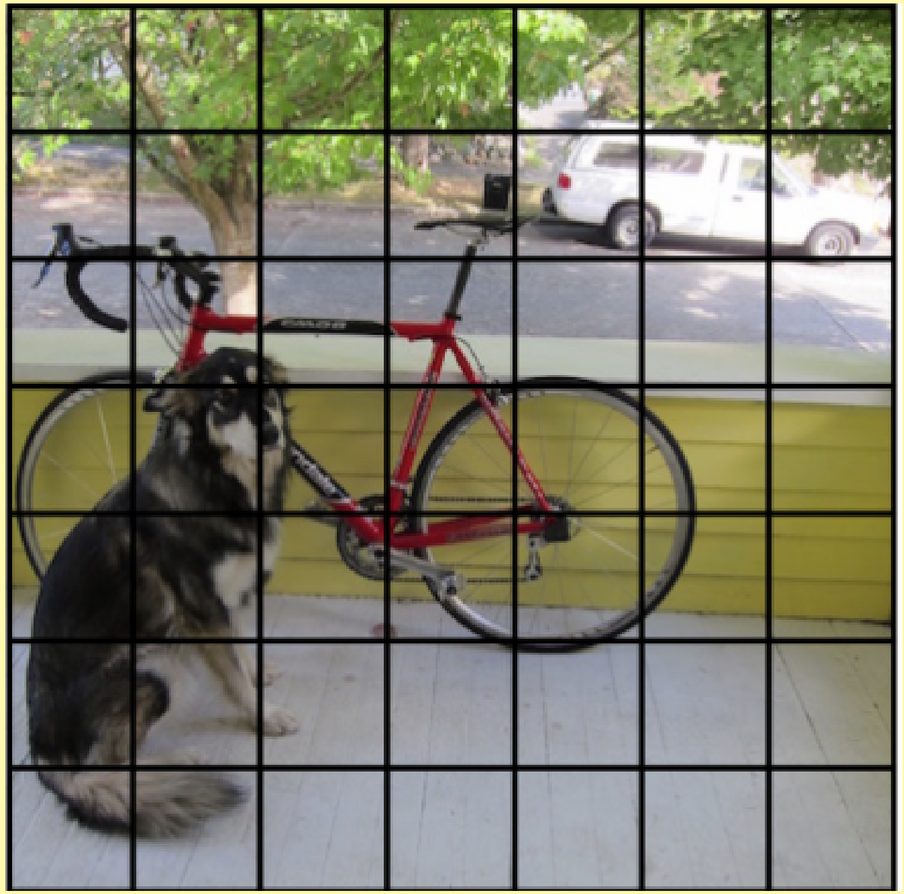


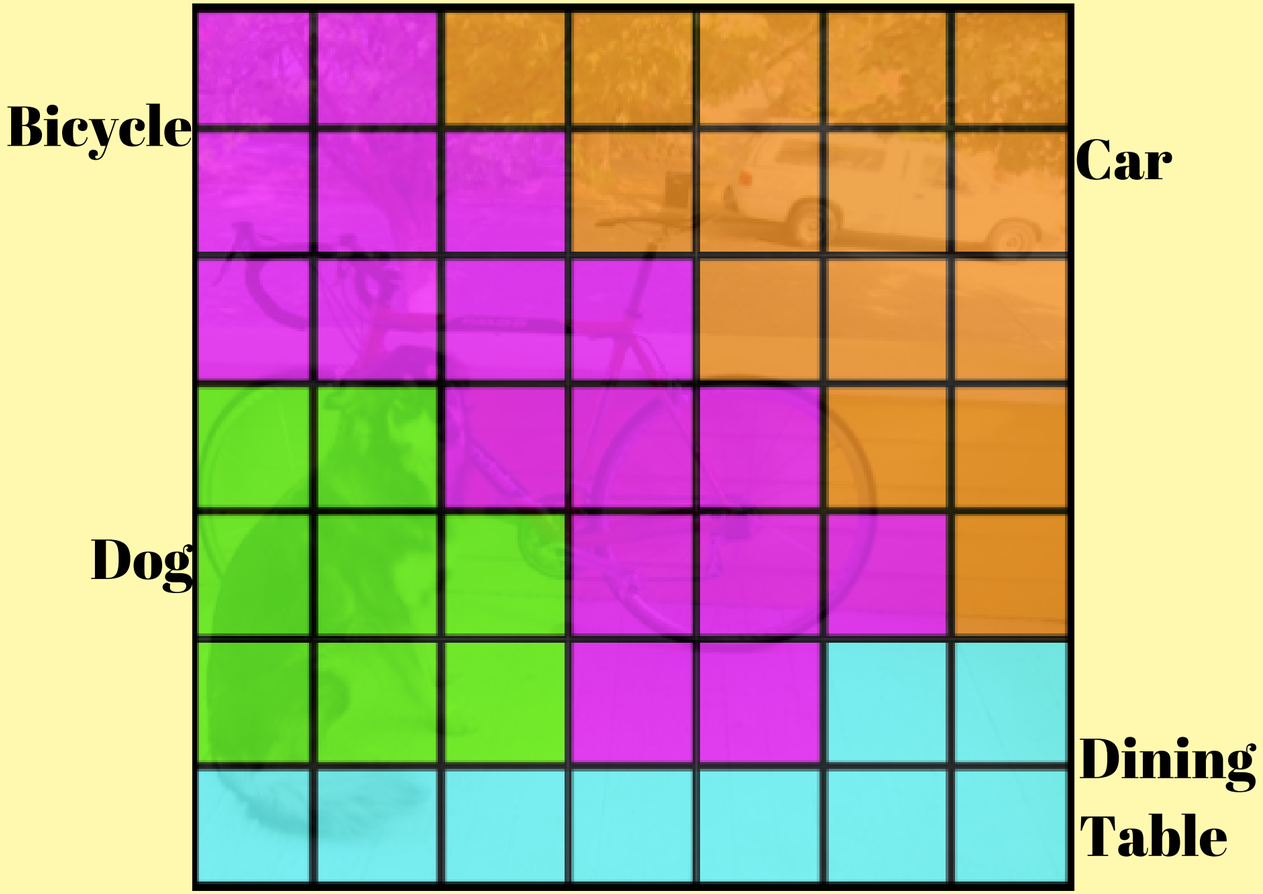


ในแต่ละตารางที่แบ่งออกมาจะทำการทำนายค่า **boxes** และค่า **confidences : P(Object)**



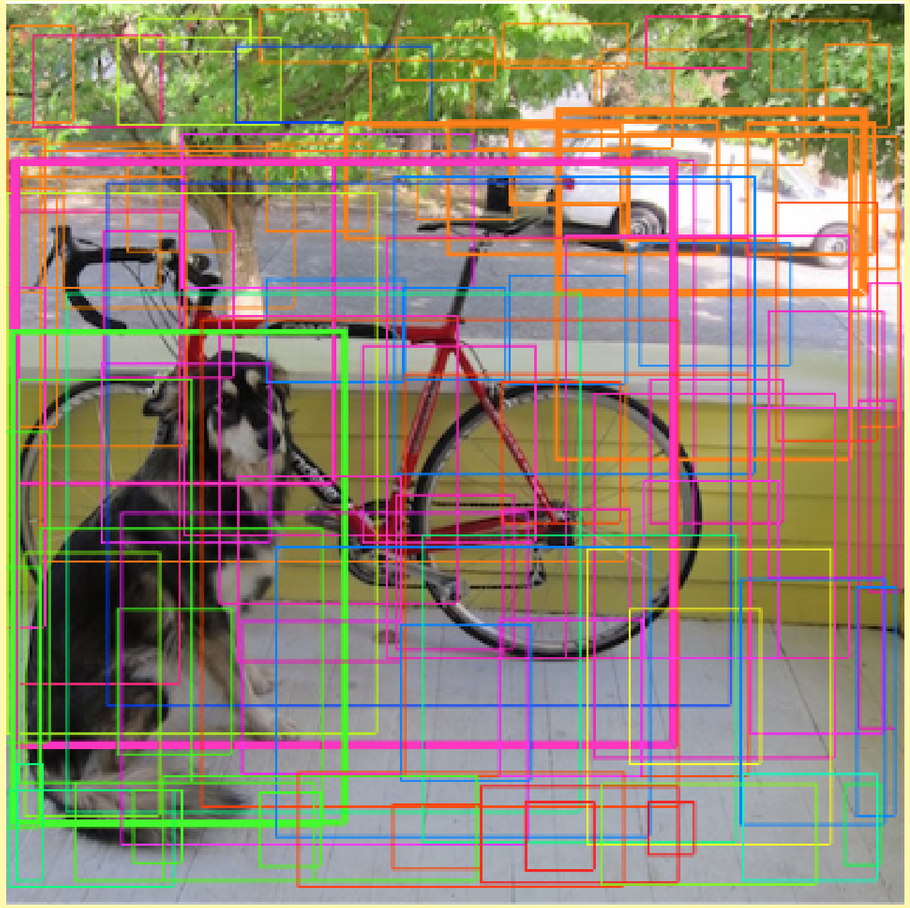
ทำนายความน่าจะเป็นของคลาสจากตารางทั้งหมด



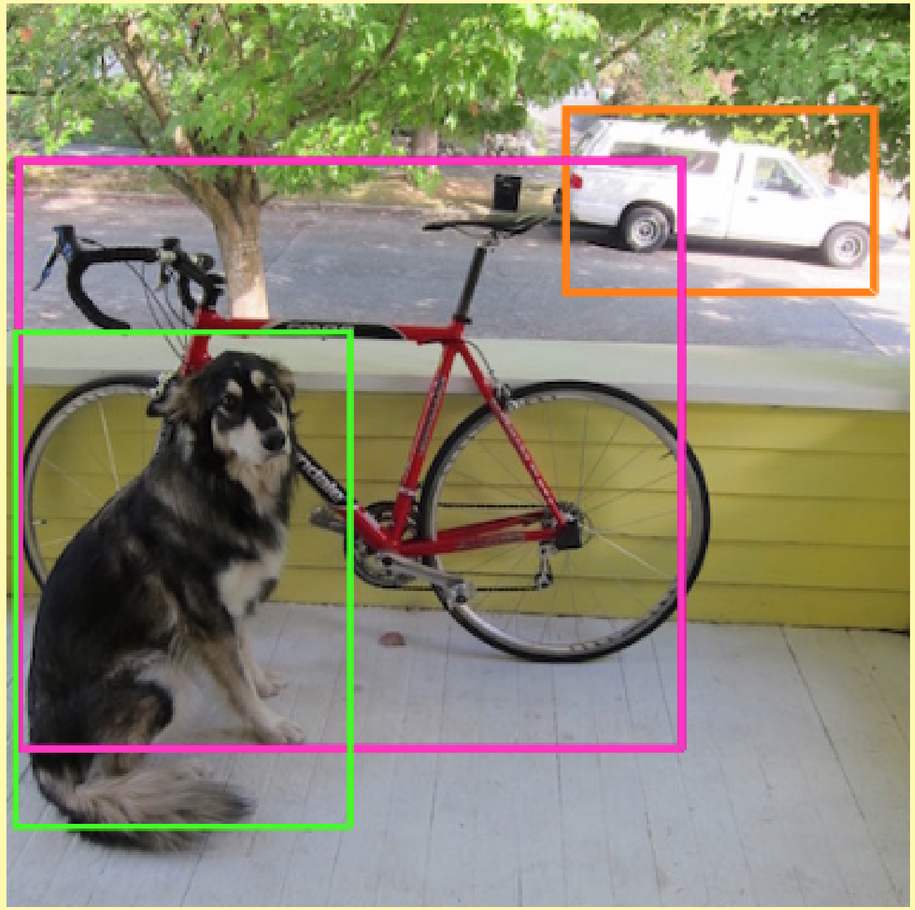


กำหนดเงื่อนไขบนวัตถุ :P(Car | Object)

รวมช่องตารางกับคลาสเข้าด้วยกัน

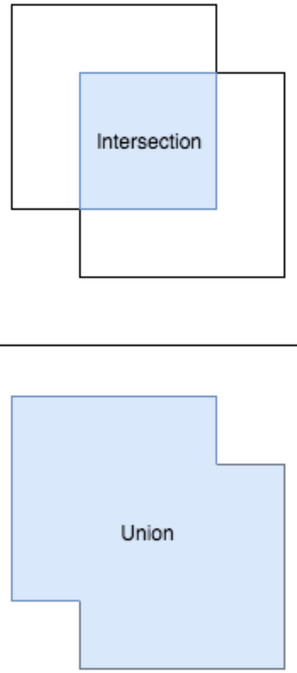


ทำการ **NMS and threshold detections**

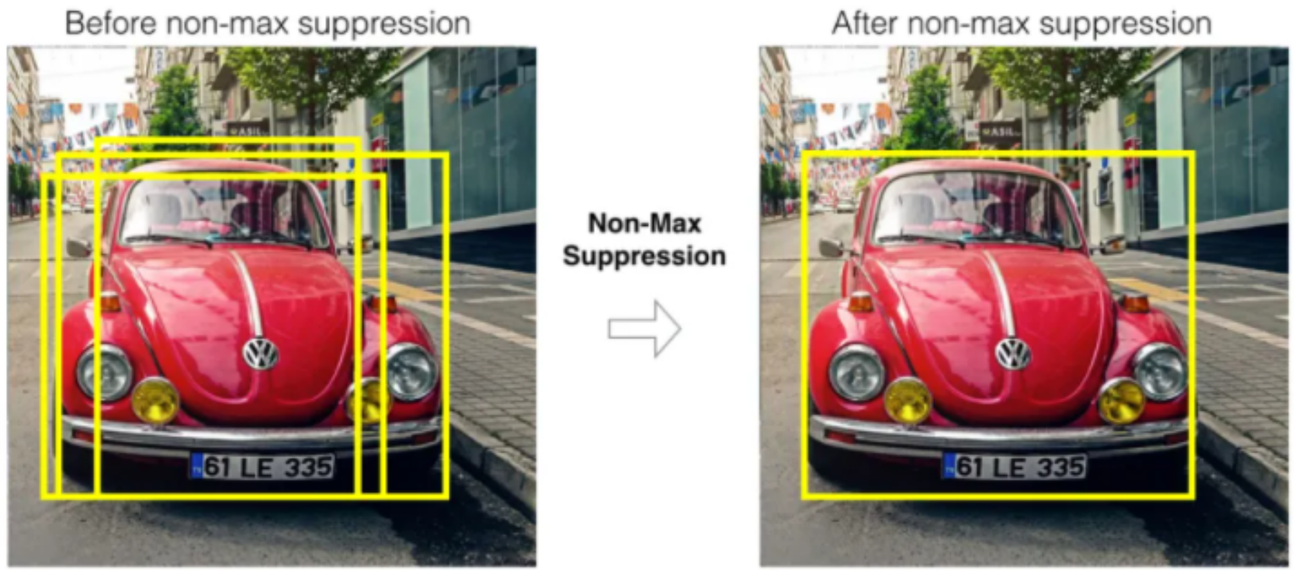


Non-maximum Suppression (NMS)

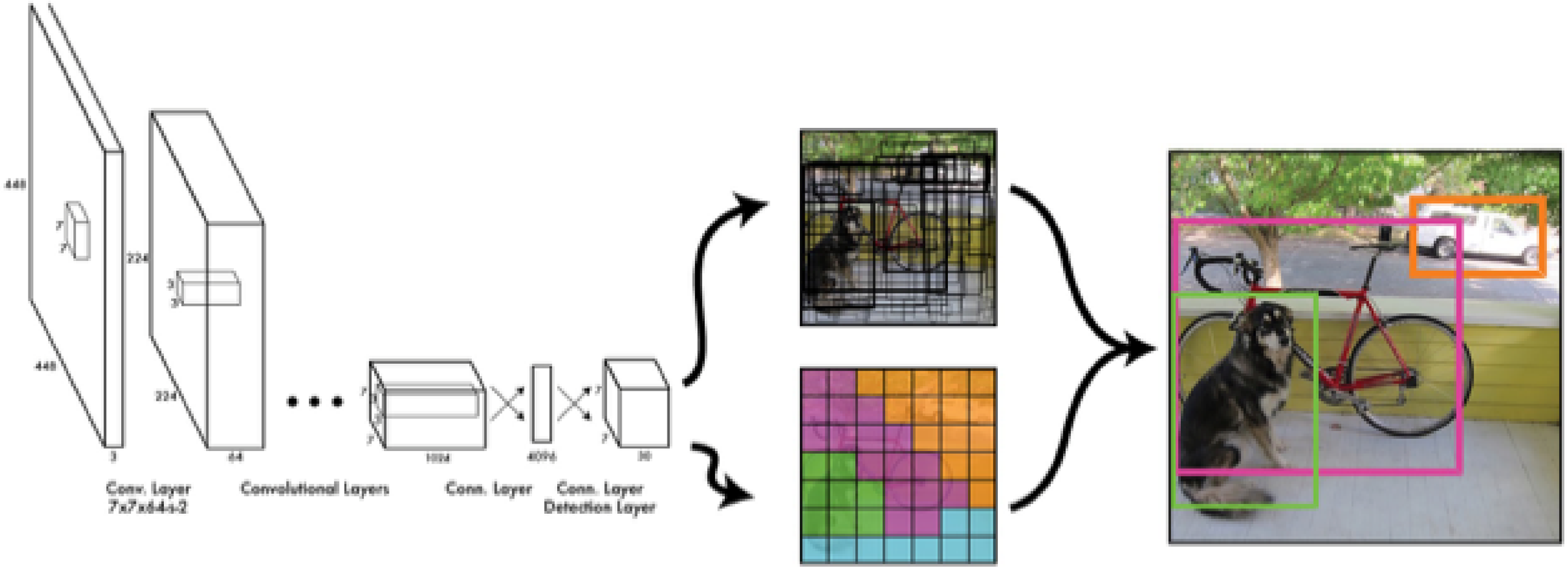
$$\text{IoU} = \frac{\text{Intersection}}{\text{Union}}$$



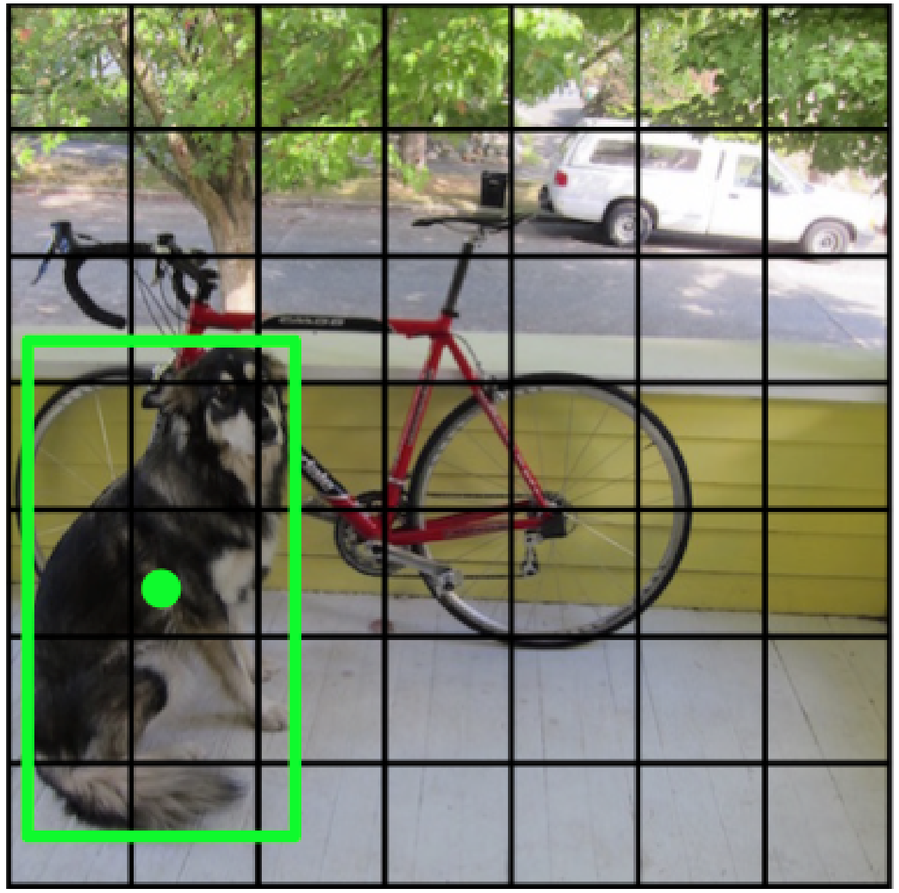
Intersection over Union

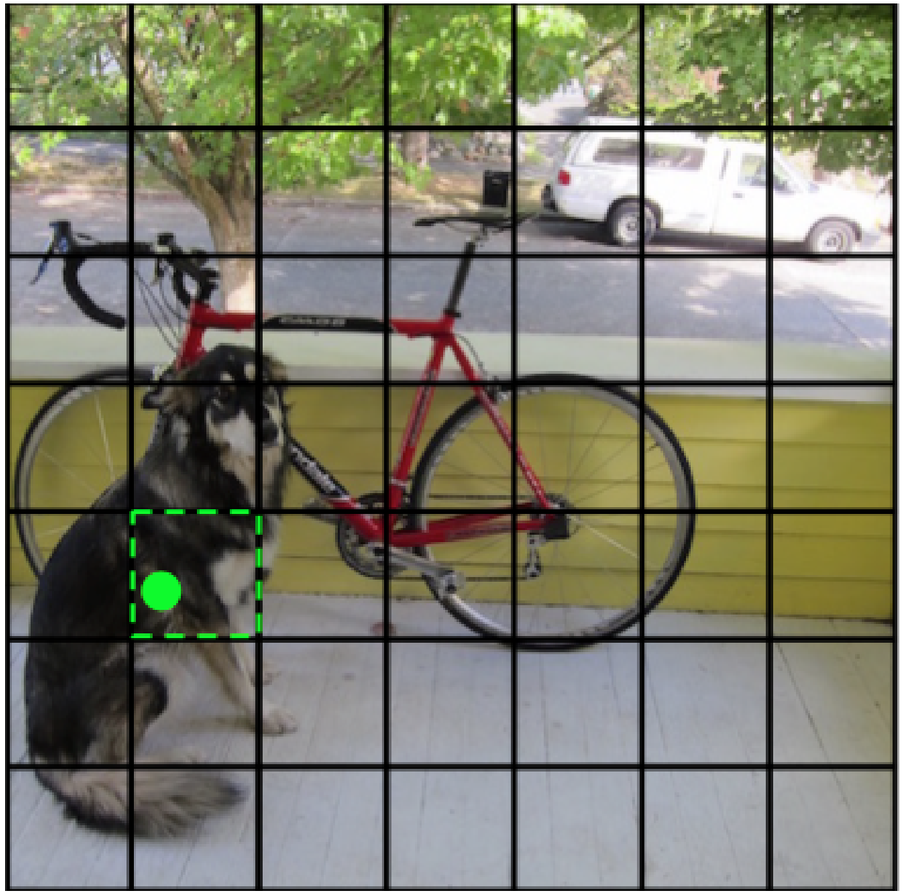


การรวมทั้งสองการตรวจจับวัตถุ



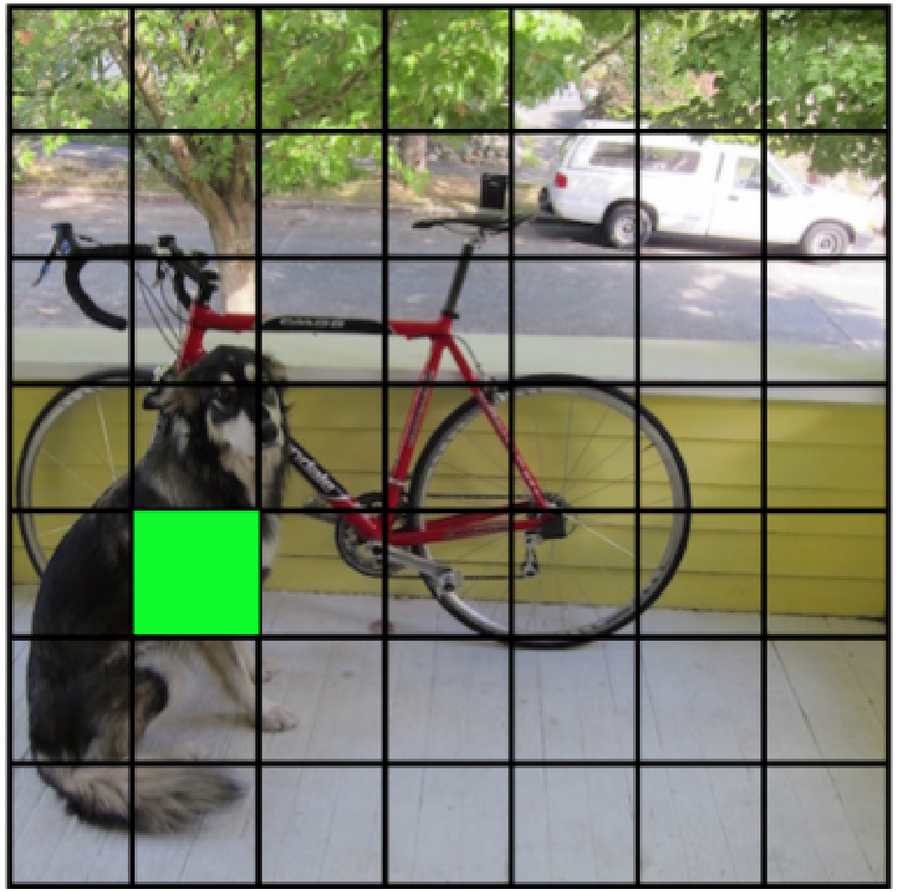
ในระหว่างการเทรนนั้นจะมีการจับคู่

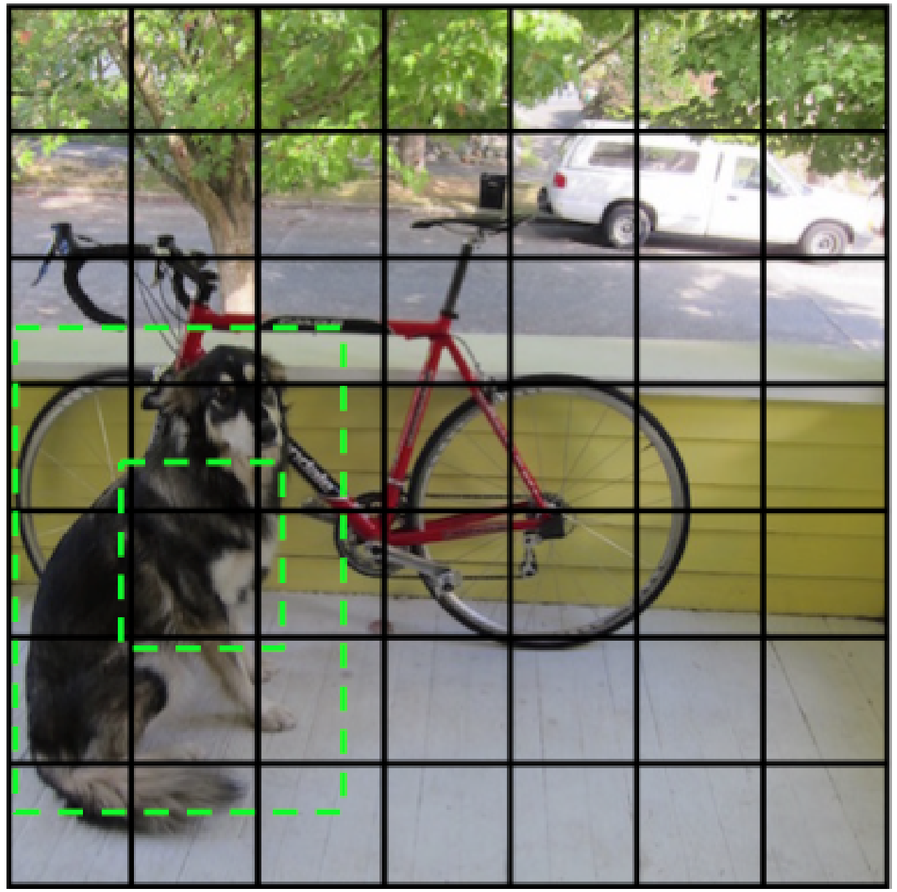




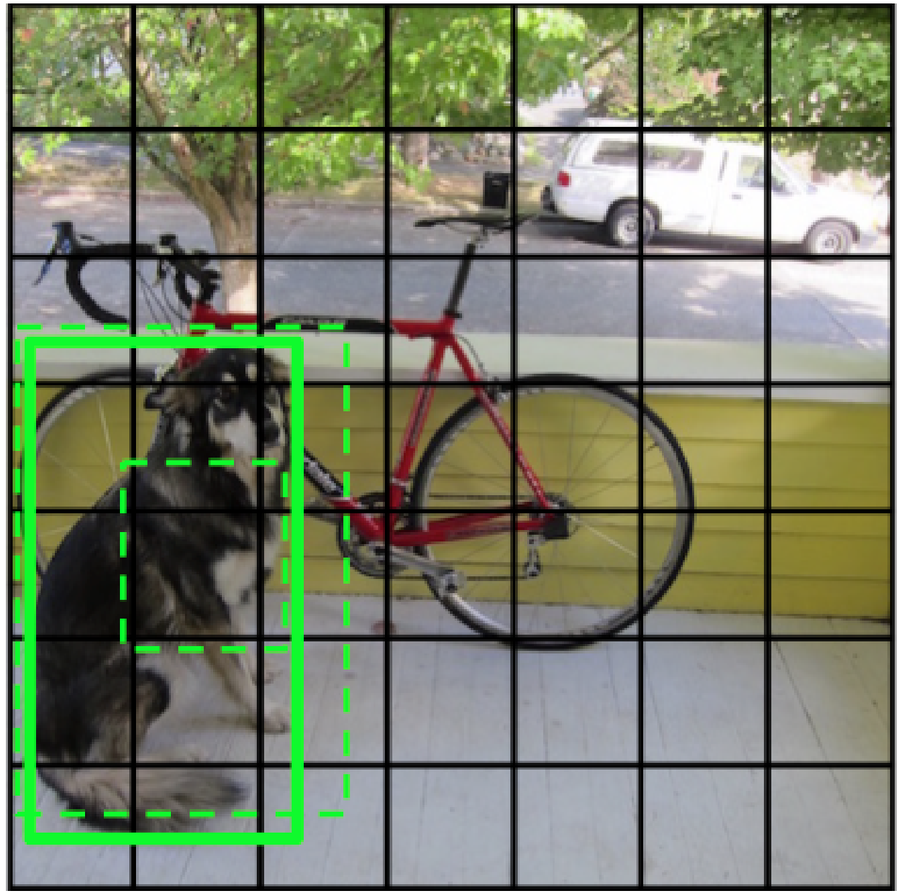
ปรับให้ช่องตารางนั้นเป็นคลาสหนึ่งคลาส

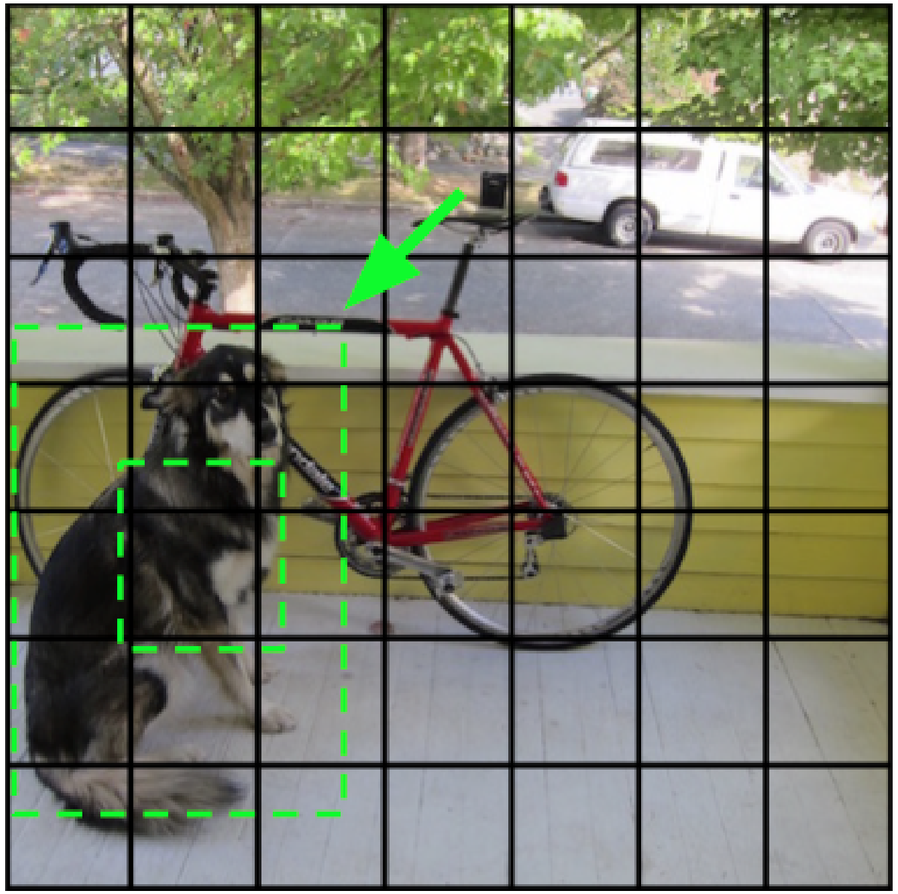
Dog = 1
Cat = 0
Bike = 0
...



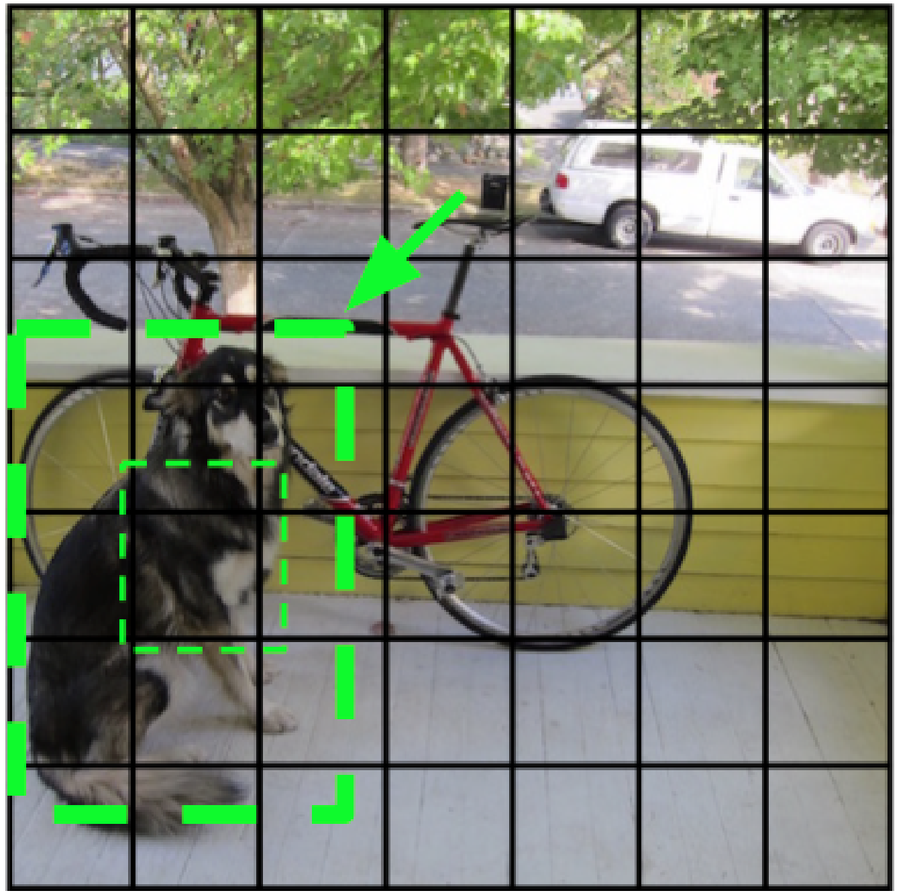


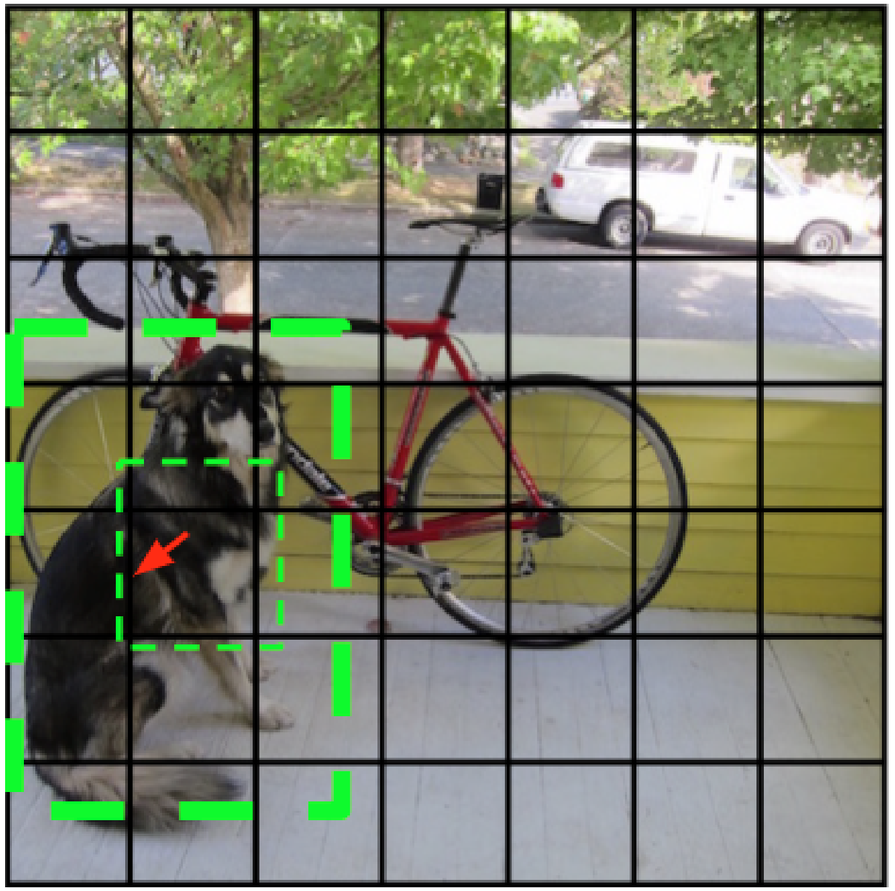
หาช่องตารางที่ดีที่สุด จากนั้นเพิ่มค่าคาดเดา



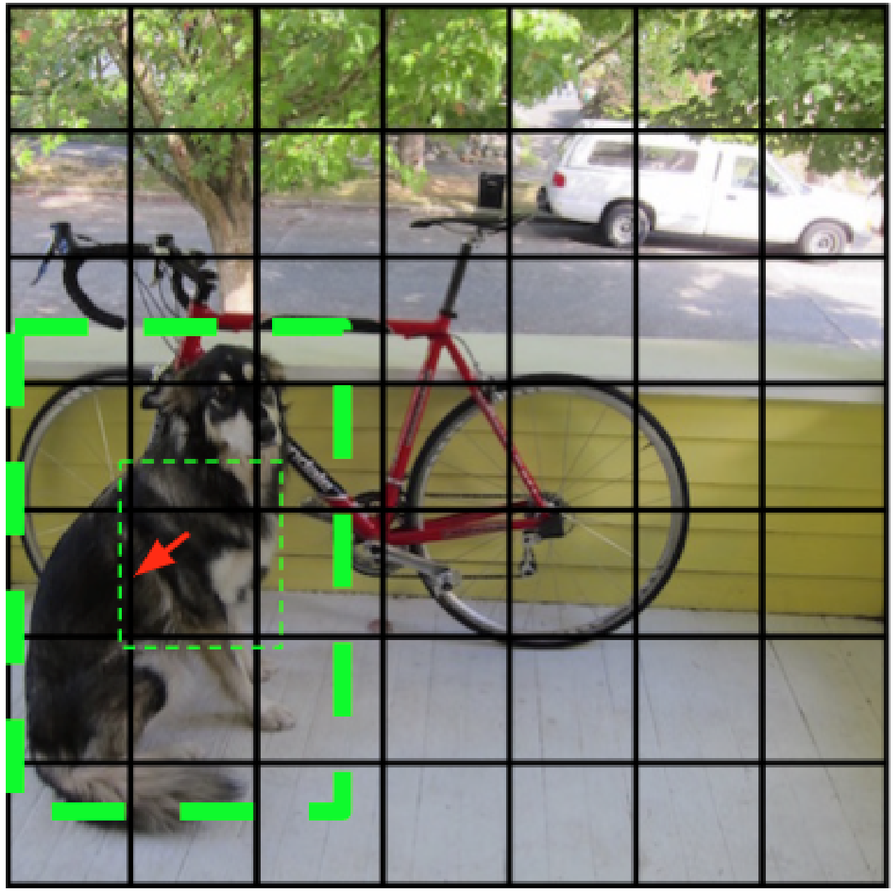


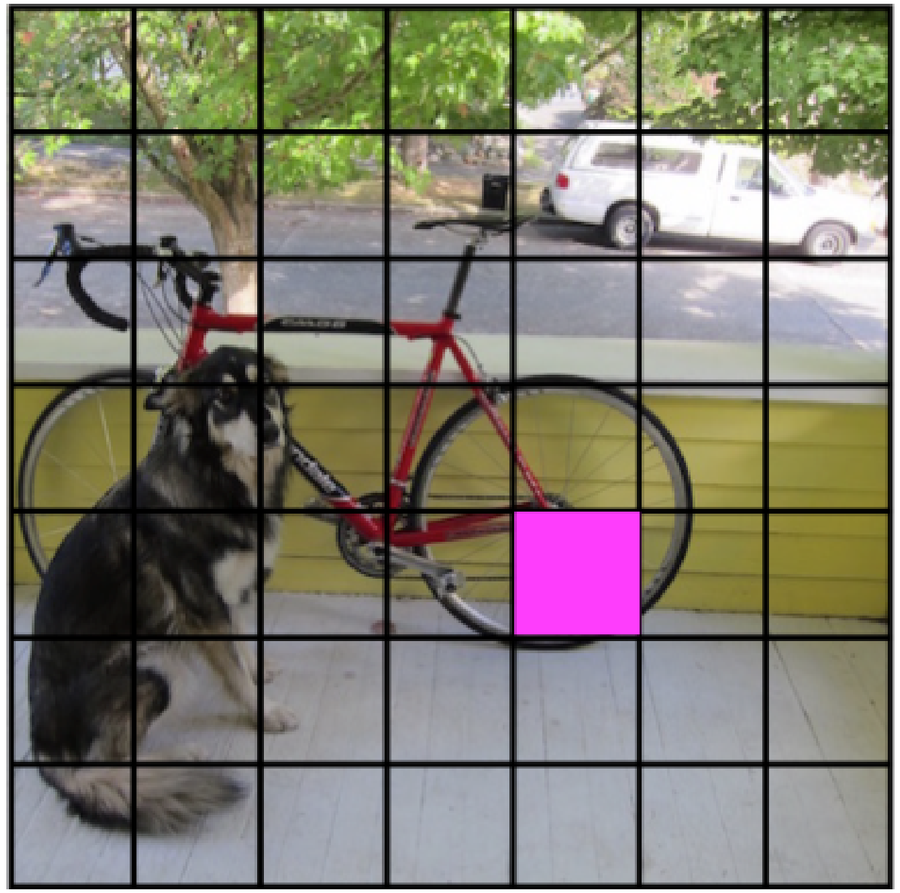
จากนั้นจะลดค่าคาดเดาของตารางอื่น



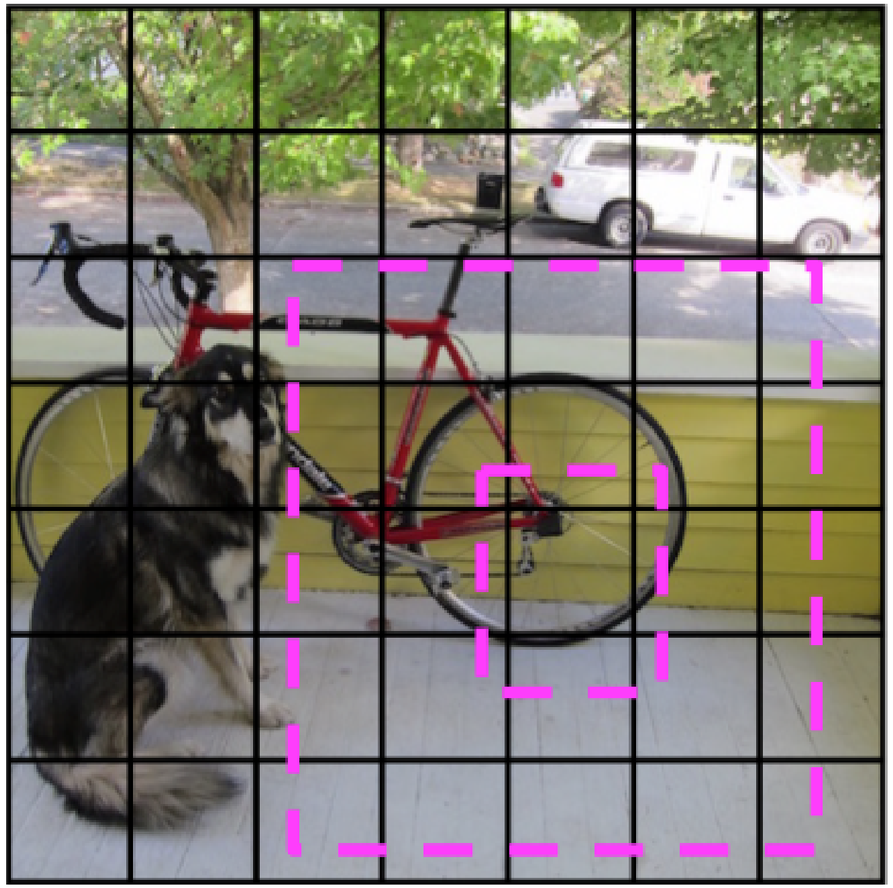


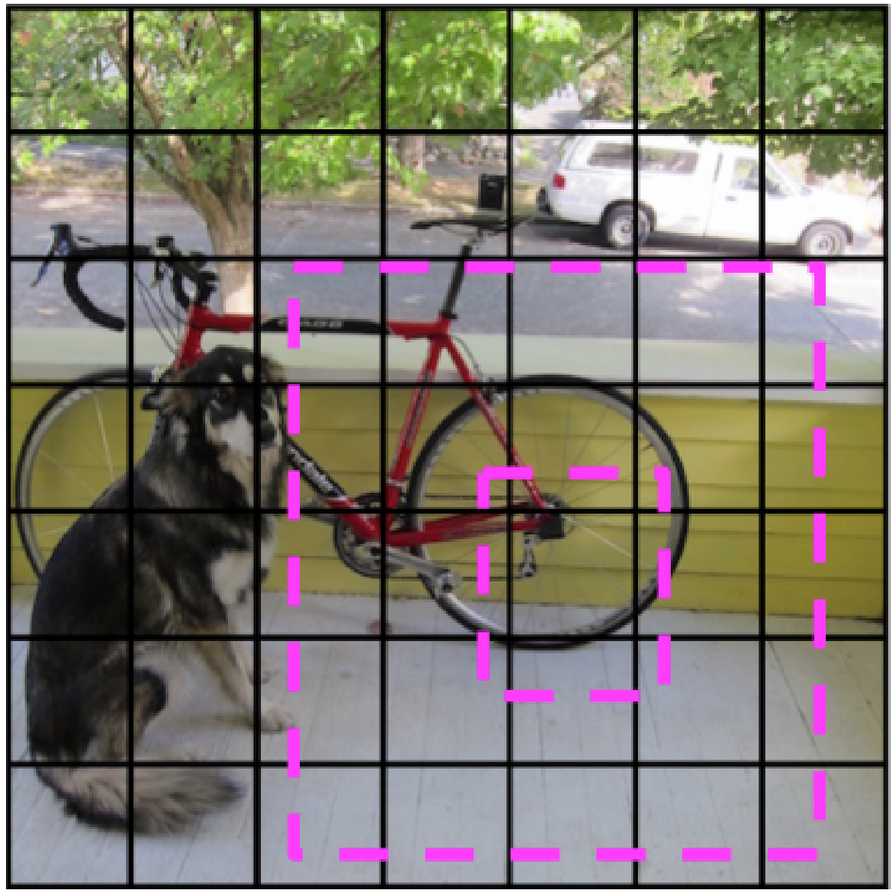
บางจุดไม่มีการอ้างอิงไปยังคลาสใด



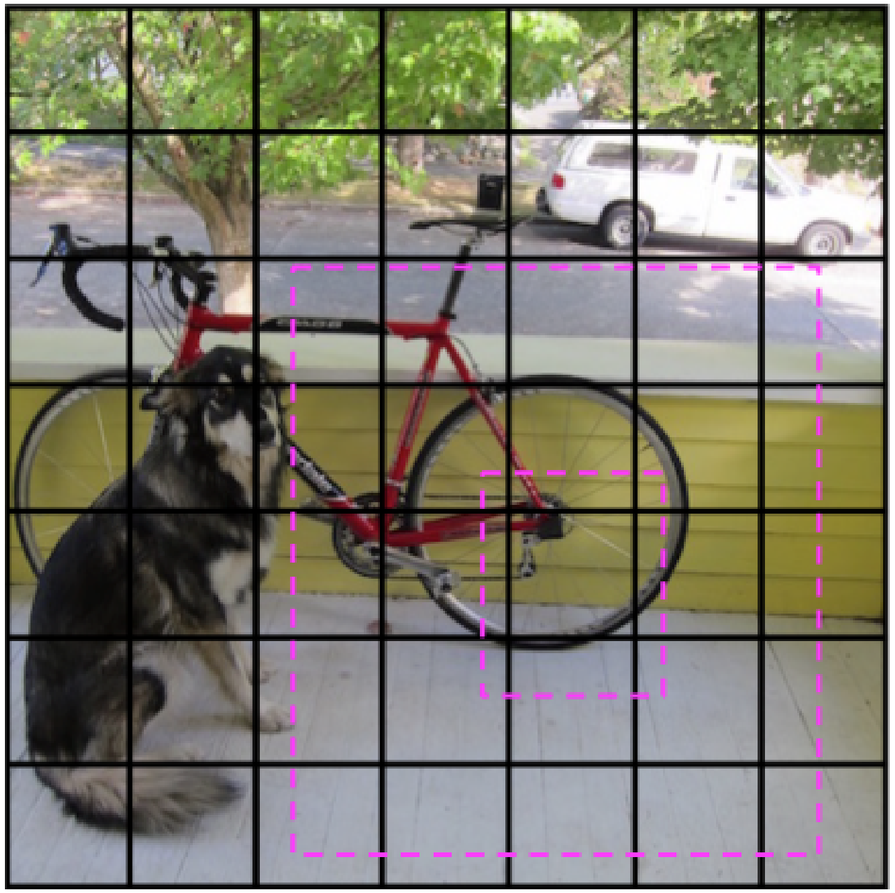


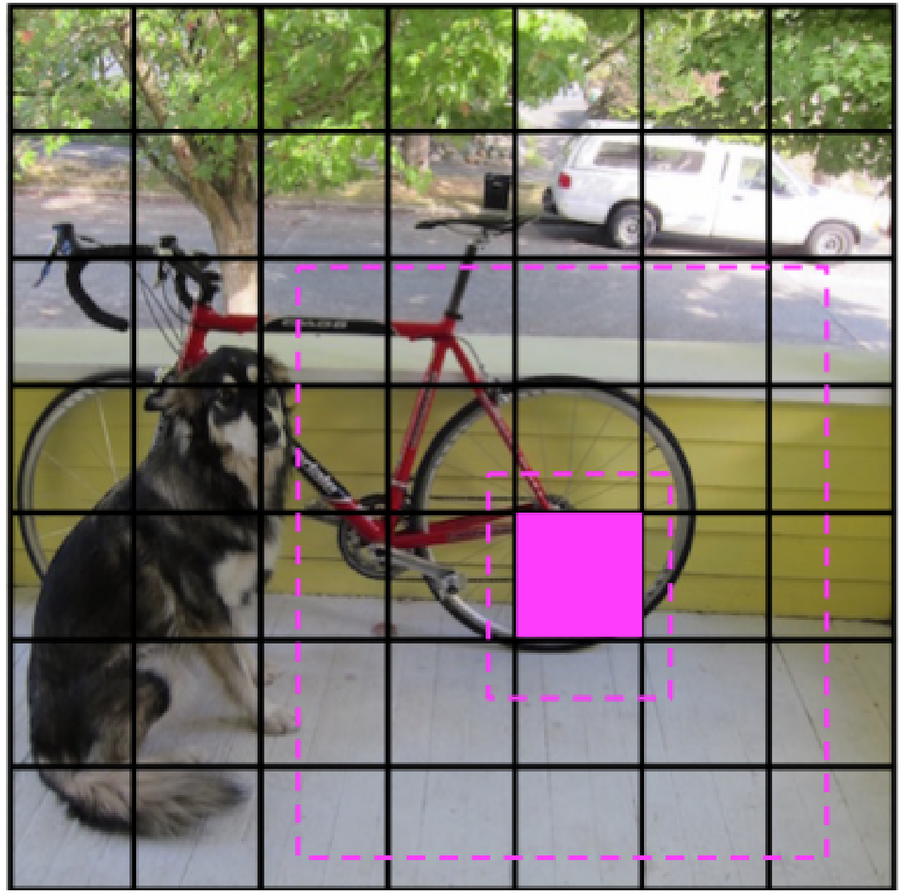
ลดค่า confidence ของกรอบนั้น ๆ



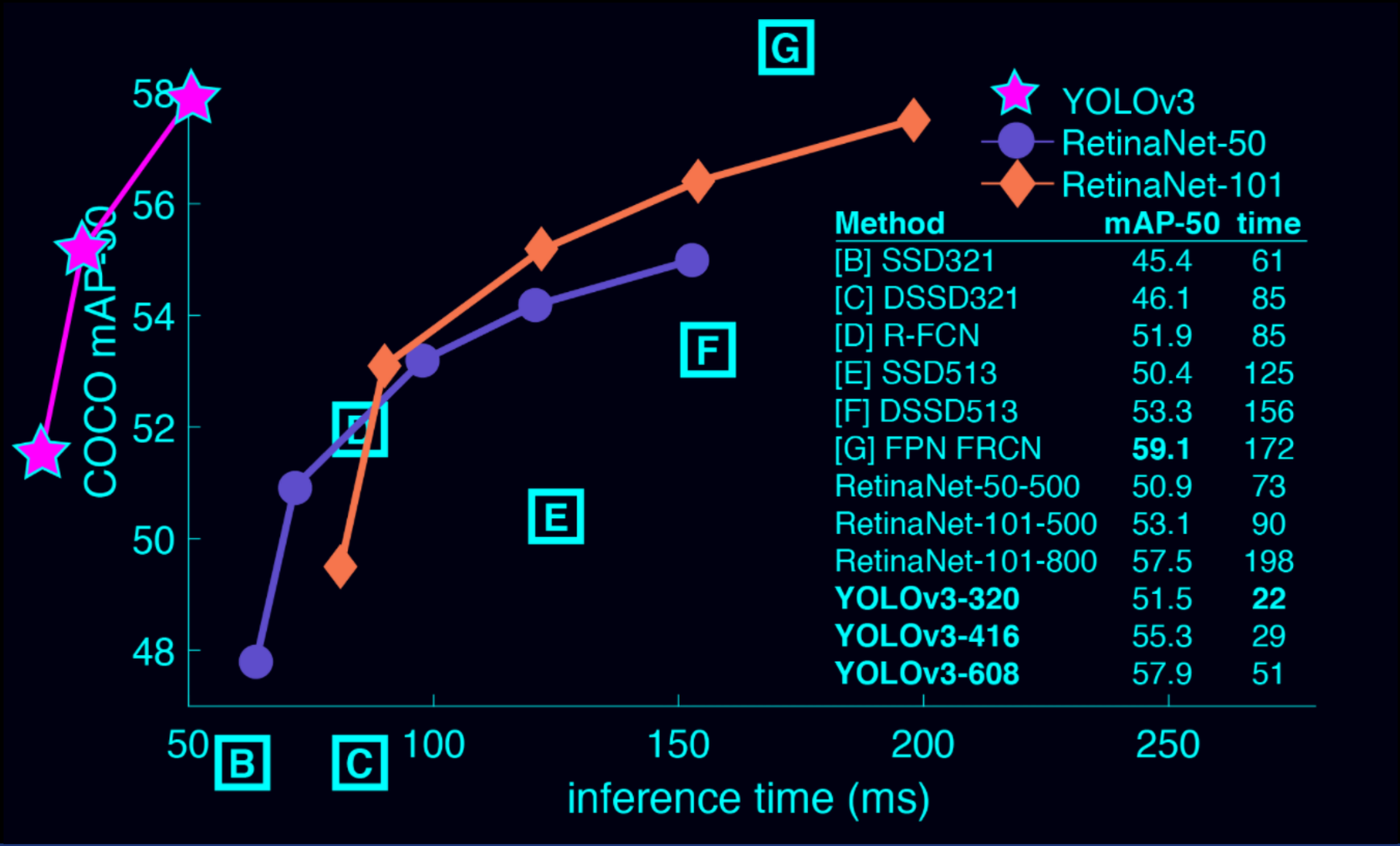


ลดค่า confidence ของกรอบนั้น ๆ





ปัจจุบัน YOLO มีหลายแบบจำลองให้เลือกใช้ (และมีโครงสร้างที่ไม่เหมือนกัน)

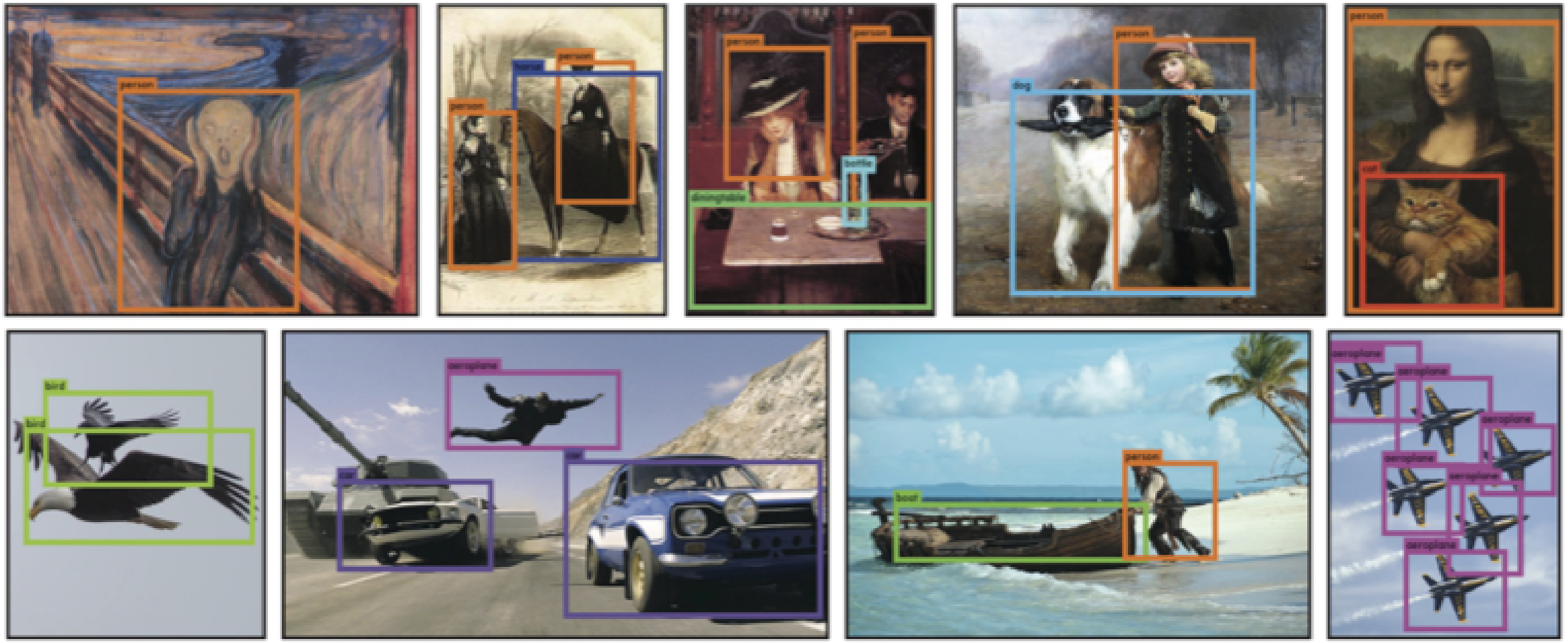


Method	mAP-50	time
[B] SSD321	45.4	61
[C] DSSD321	46.1	85
[D] R-FCN	51.9	85
[E] SSD513	50.4	125
[F] DSSD513	53.3	156
[G] FPN FRCN	**59.1**	172
RetinaNet-50-500	50.9	73
RetinaNet-101-500	53.1	90
RetinaNet-101-800	57.5	198
YOLOv3-320	51.5	**22**
YOLOv3-416	55.3	29
YOLOv3-608	57.9	51

YOLO

Model	Train	Test	mAP	FLOPS	FPS	Cfg	Weights
YOLOv3-320	COCO trainval	test-dev	51.5	38.97 Bn	45	cfg	weights
YOLOv3-416	COCO trainval	test-dev	55.3	65.86 Bn	35	cfg	weights
YOLOv3-608	COCO trainval	test-dev	57.9	140.69 Bn	20	cfg	weights
YOLOv3-tiny	COCO trainval	test-dev	33.1	5.56 Bn	220	cfg	weights
YOLOv3-spp	COCO trainval	test-dev	60.6	141.45 Bn	20	cfg	weights

YOLOv3-tiny คือ Layers น้อยที่สุด (อันบนสุด) จะสังเกตุว่าความเร็วค่อนข้างดีมาก 220 FPS
แต่ถ้าอยากให้แบบจำลองดีขึ้น อาจเลือกใช้ YOLOv3-spp ก็จะถูกต้องขึ้นมาก แต่ความเร็วก็จะตกลง

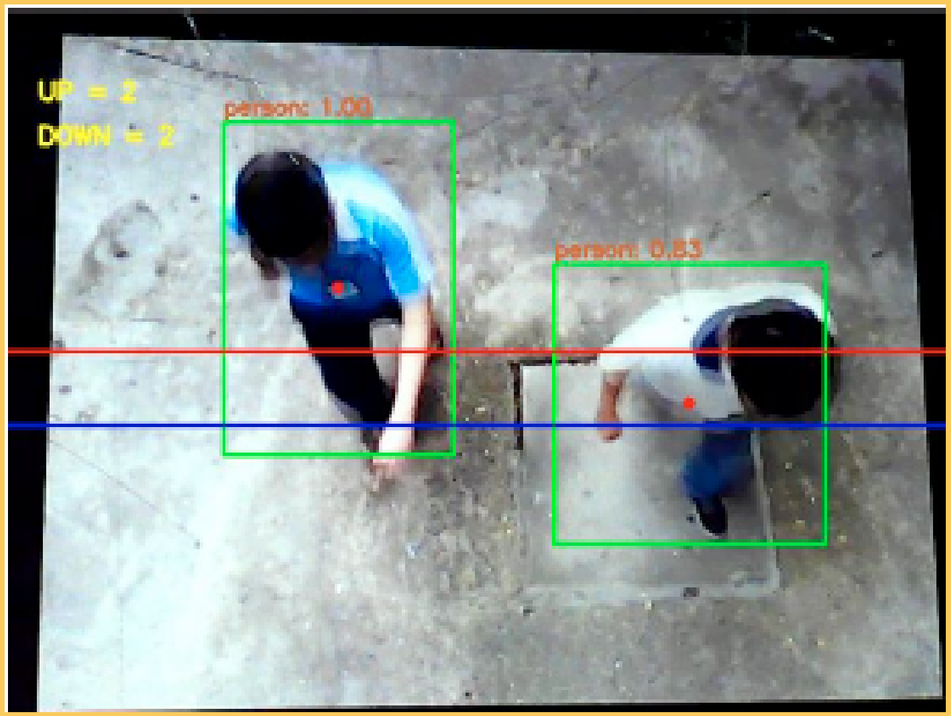

person
person
person
person
person
person
person
person
dog
person
person
cat
bird
bird
aeroplane
car
car
person
boat
aeroplane
aeroplane
aeroplane
aeroplane
aeroplane
aeroplane

ตัวอย่างการนำไปใช้งาน

การตรวจจับหมวกนิรภัยและการใช้อาวุธปืน
เพื่อเตือนภัยเหตุโจรกรรม
จากภาพกล้องวงจรปิดแบบเวลาจริง

จากงานวิจัย
คุณยงยุทธ ละมูลมอญ
และ
คุณธนาสัย สุคนธ์พันธ์

ตัวอย่างการนำไปใช้งาน



การสร้างระบบตรวจนับบุคคลแบบเวลาจริง
ราคาประหยัด
บน Raspberry Pi โดยประยุกต์อัลกอริทึม
Tiny YOLO V3

จากงานวิจัย
คุณปราโมทย์ ปัญญาโต
และ
คุณนลิน สีดาห้าว

Tools Anaconda

ANACONDA®

https://www.anaconda.com/products/individual#Downloads

Anaconda Installers

Windows

Python 3.8

64-Bit Graphical Installer (477 MB)

32-Bit Graphical Installer (409 MB)

Anaconda3-2021.0....exe
66.3/477 MB, 1 min left

MacOS

Python 3.8

64-Bit Graphical Installer (440 MB)

64-Bit Command Line Installer (433 MB)

Linux

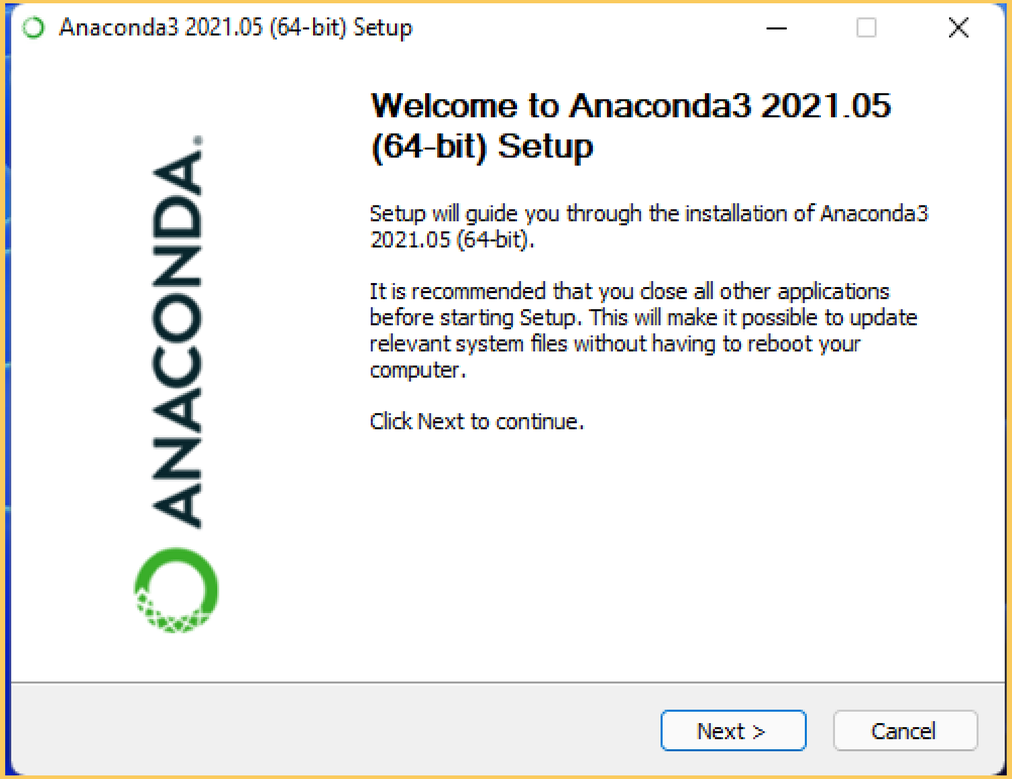
Python 3.8

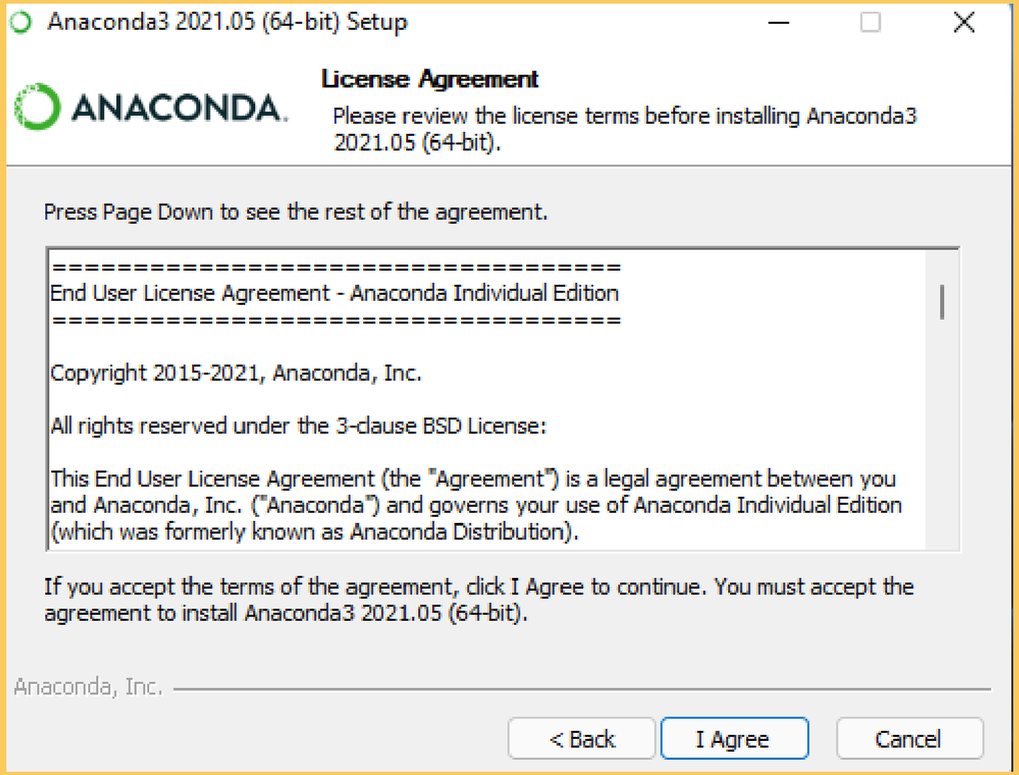
64-Bit (x86) Installer (544 MB)

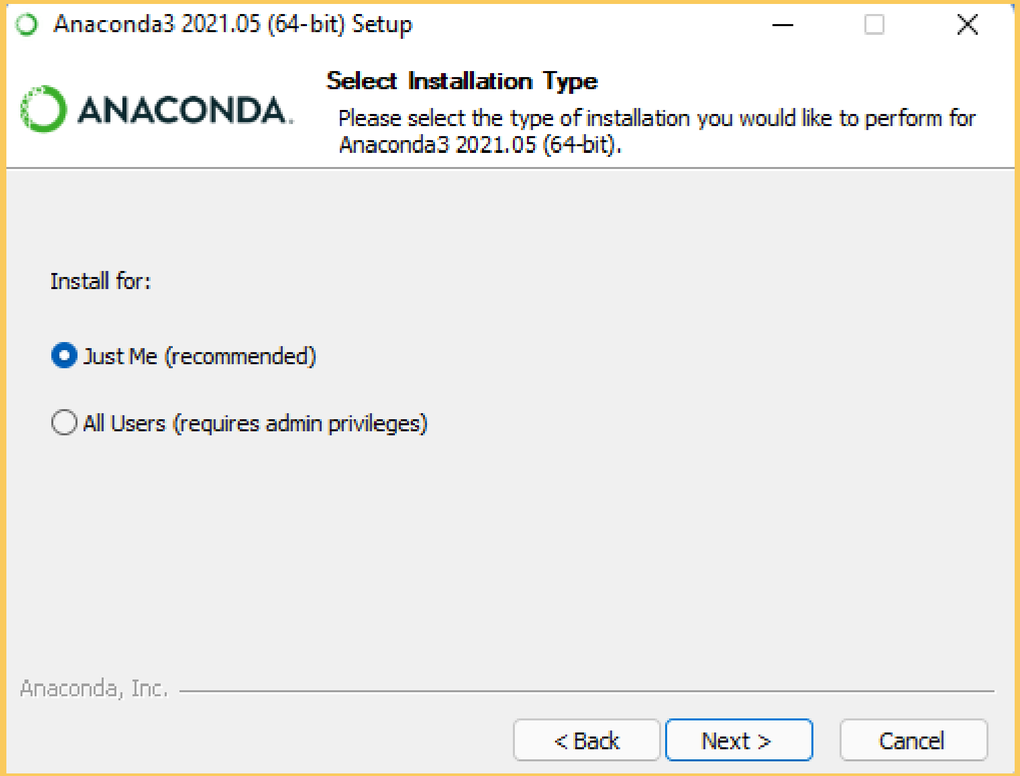
64-Bit (Power8 and Power9) Installer (285 MB)

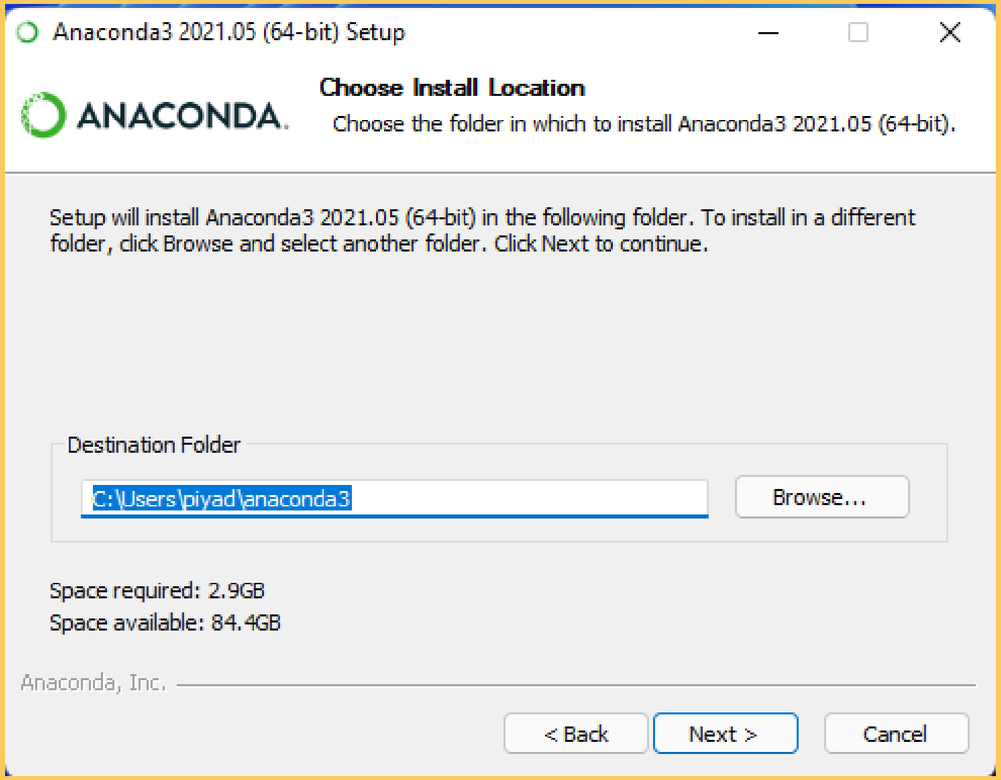
64-Bit (AWS Graviton2 / ARM64) Installer (413 M)

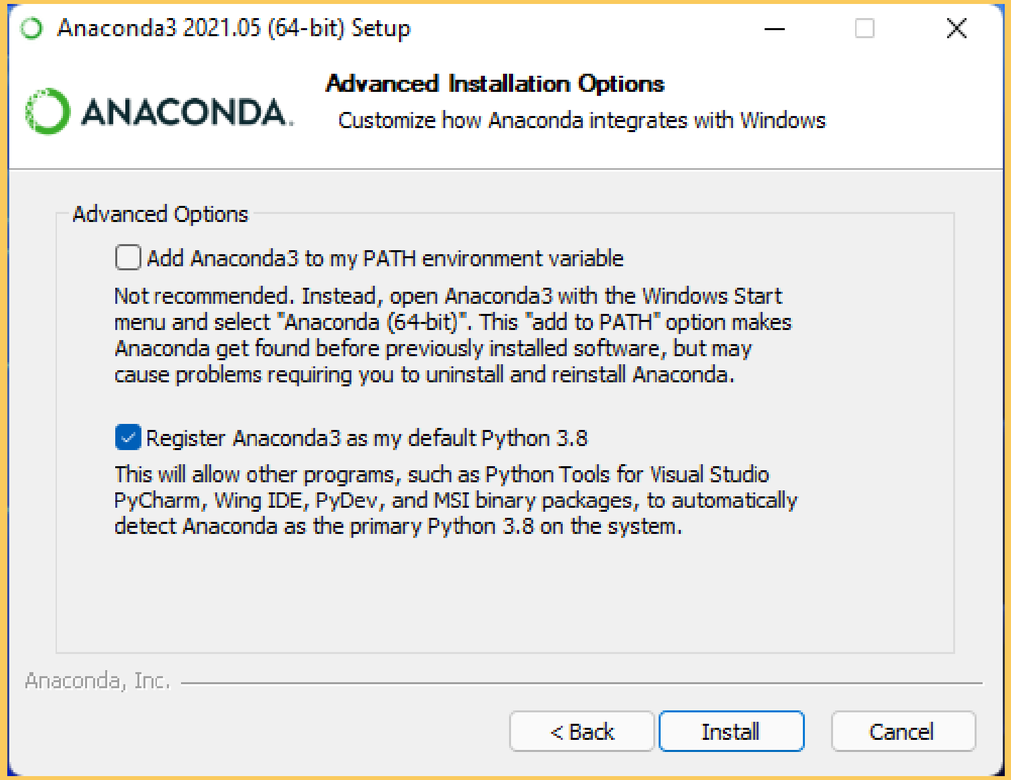
64-bit (Linux on IBM Z & LinuxONE) Installer (292 M)

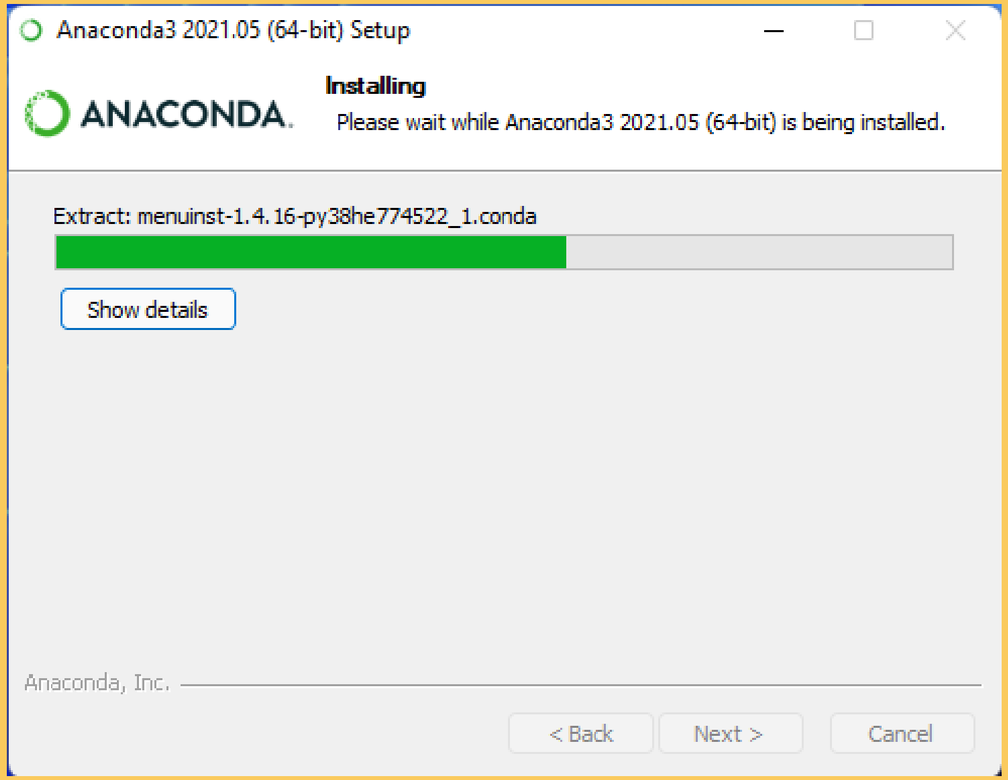

Anaconda3 2021.05 (64-bit) Setup
ANACONDA.
Welcome to Anaconda3 2021.05 (64-bit) Setup
Setup will guide you through the installation of Anaconda3 2021.05 (64-bit).
It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.
Click Next to continue.
Next >
Cancel

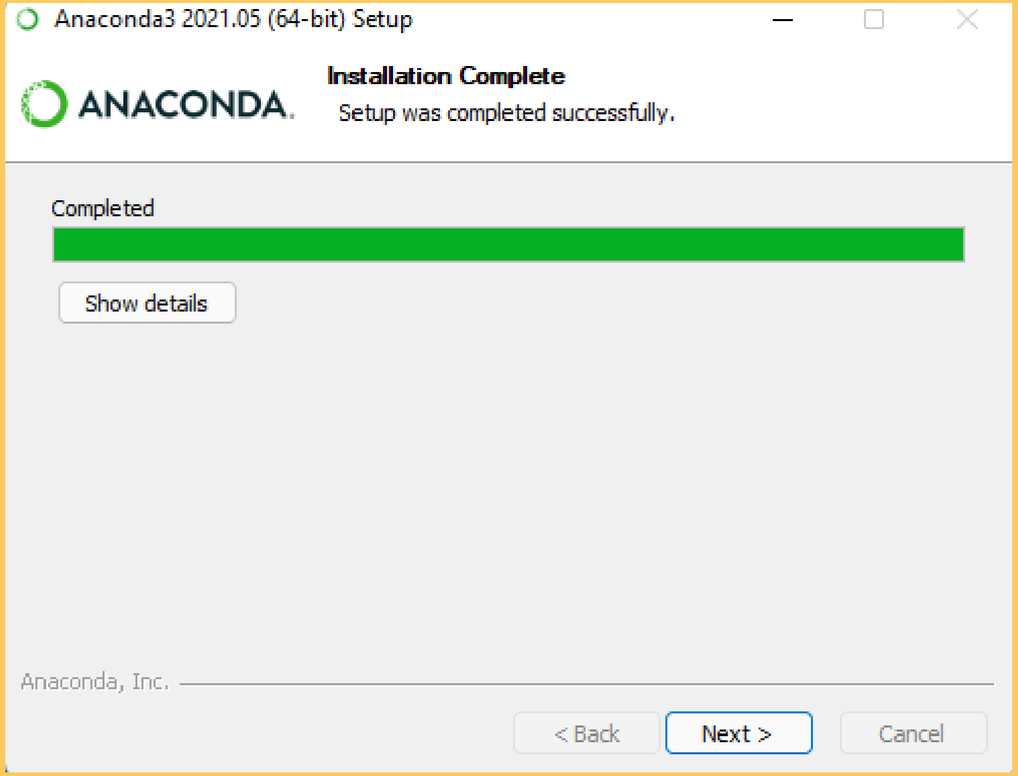

Anaconda3 2021.05 (64-bit) Setup
ANACONDA.
License Agreement
Please review the license terms before installing Anaconda3 2021.05 (64-bit).
Press Page Down to see the rest of the agreement.
==========
End User License Agreement - Anaconda Individual Edition
==========
Copyright 2015-2021, Anaconda, Inc.
All rights reserved under the 3-clause BSD License:
This End User License Agreement (the "Agreement") is a legal agreement between you and Anaconda, Inc. ("Anaconda") and governs your use of Anaconda Individual Edition (which was formerly known as Anaconda Distribution).
If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Anaconda3 2021.05 (64-bit).
Anaconda, Inc.
< Back
I Agree
Cancel

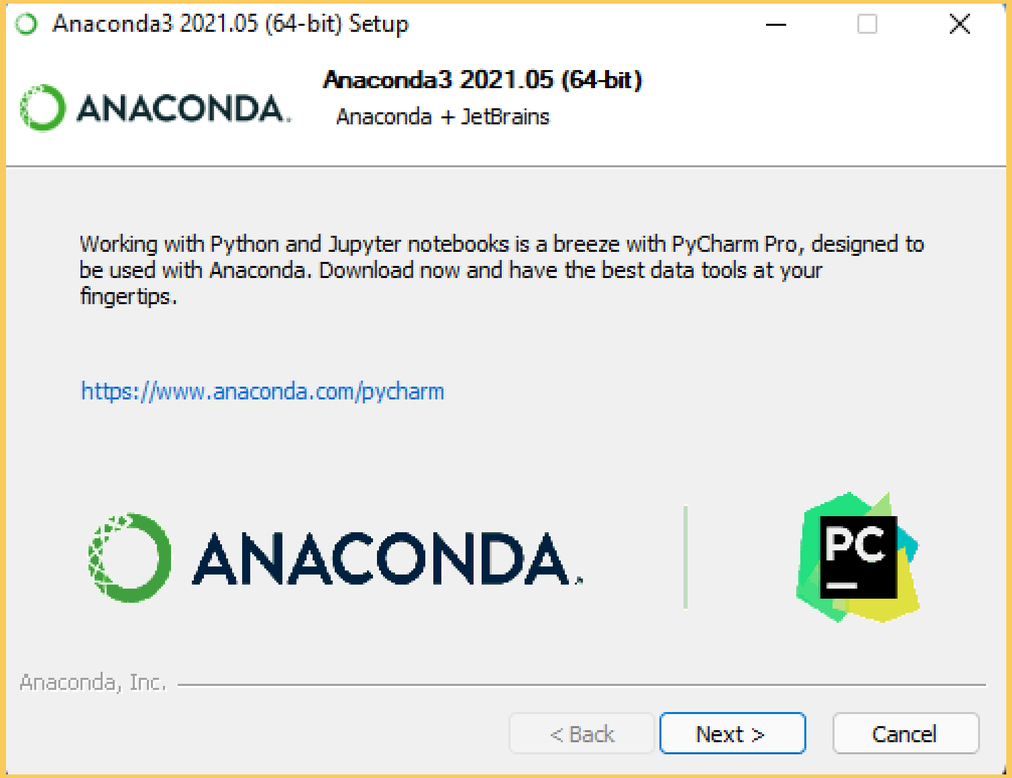

Anaconda3 2021.05 (64-bit) Setup
ANACONDA.
Select Installation Type
Please select the type of installation you would like to perform for Anaconda3 2021.05 (64-bit).
Install for:
Just Me (recommended)
All Users (requires admin privileges)
Anaconda, Inc.
< Back
Next >
Cancel

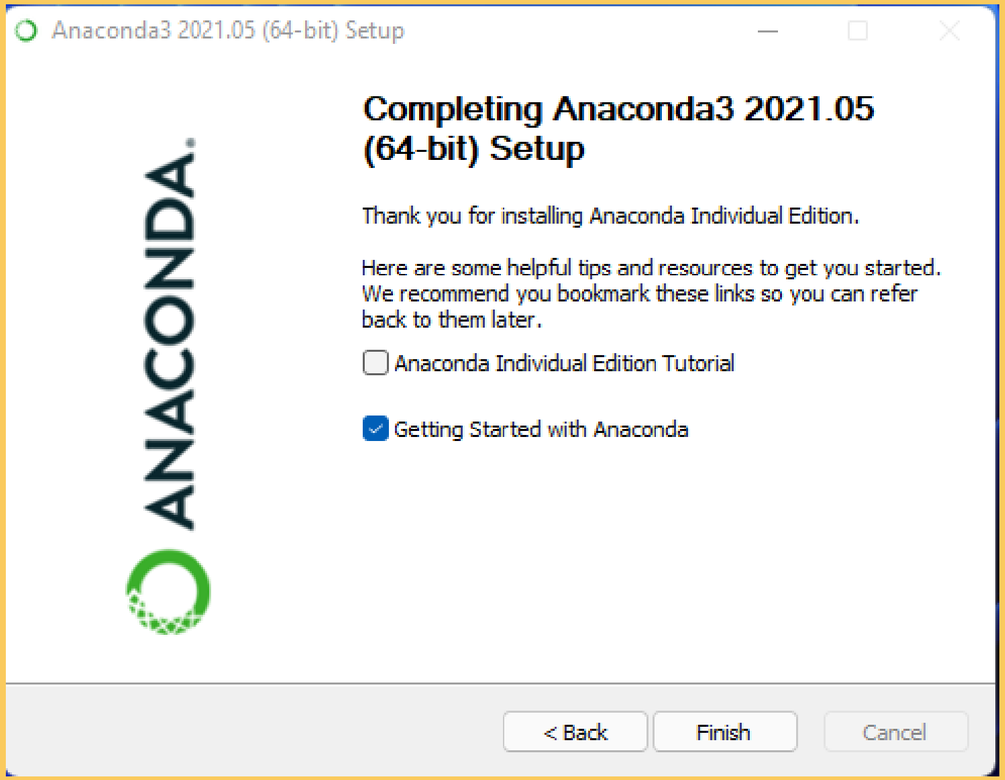

Anaconda3 2021.05 (64-bit) Setup
ANACONDA.
Choose Install Location
Choose the folder in which to install Anaconda3 2021.05 (64-bit).
Setup will install Anaconda3 2021.05 (64-bit) in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.
Destination Folder
C:\Users\piyad\anaconda3
Browse...
Space required: 2.9GB
Space available: 84.4GB
Anaconda, Inc.
< Back
Next >
Cancel


Anaconda3 2021.05 (64-bit) Setup
ANACONDA.
Advanced Installation Options
Customize how Anaconda integrates with Windows
Advanced Options
Add Anaconda3 to my PATH environment variable
Not recommended. Instead, open Anaconda3 with the Windows Start menu and select "Anaconda (64-bit)". This "add to PATH" option makes Anaconda get found before previously installed software, but may cause problems requiring you to uninstall and reinstall Anaconda.
Register Anaconda3 as my default Python 3.8
This will allow other programs, such as Python Tools for Visual Studio PyCharm, Wing IDE, PyDev, and MSI binary packages, to automatically detect Anaconda as the primary Python 3.8 on the system.
Anaconda, Inc.
< Back
Install
Cancel


Anaconda3 2021.05 (64-bit) Setup
ANACONDA.
Installing
Please wait while Anaconda3 2021.05 (64-bit) is being installed.
Extract: menuinst-1.4.16-py38he774522_1.conda
Show details
Anaconda, Inc.
< Back
Next >
Cancel

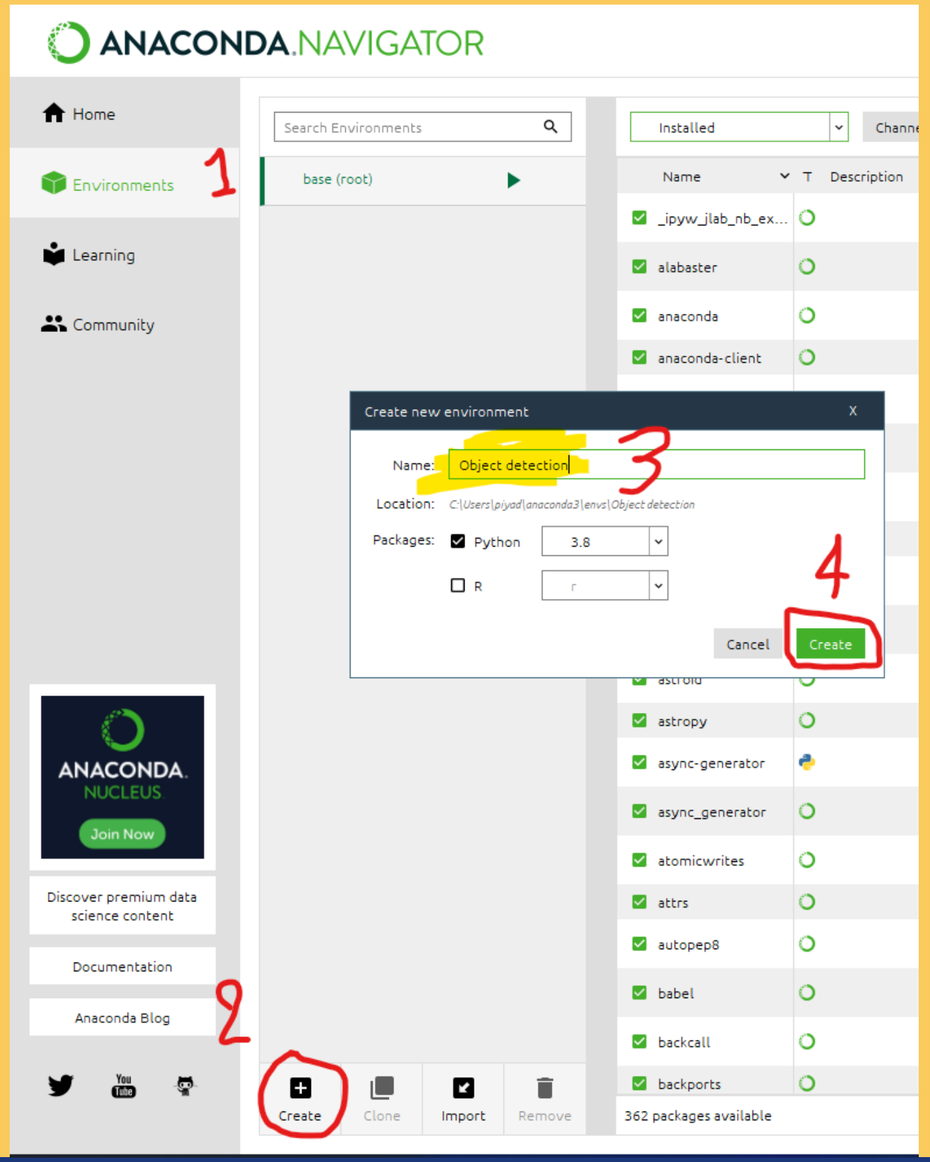

Anaconda3 2021.05 (64-bit) Setup
ANACONDA.
Installation Complete
Setup was completed successfully.
Completed
Show details
Anaconda, Inc.
< Back
Next >
Cancel

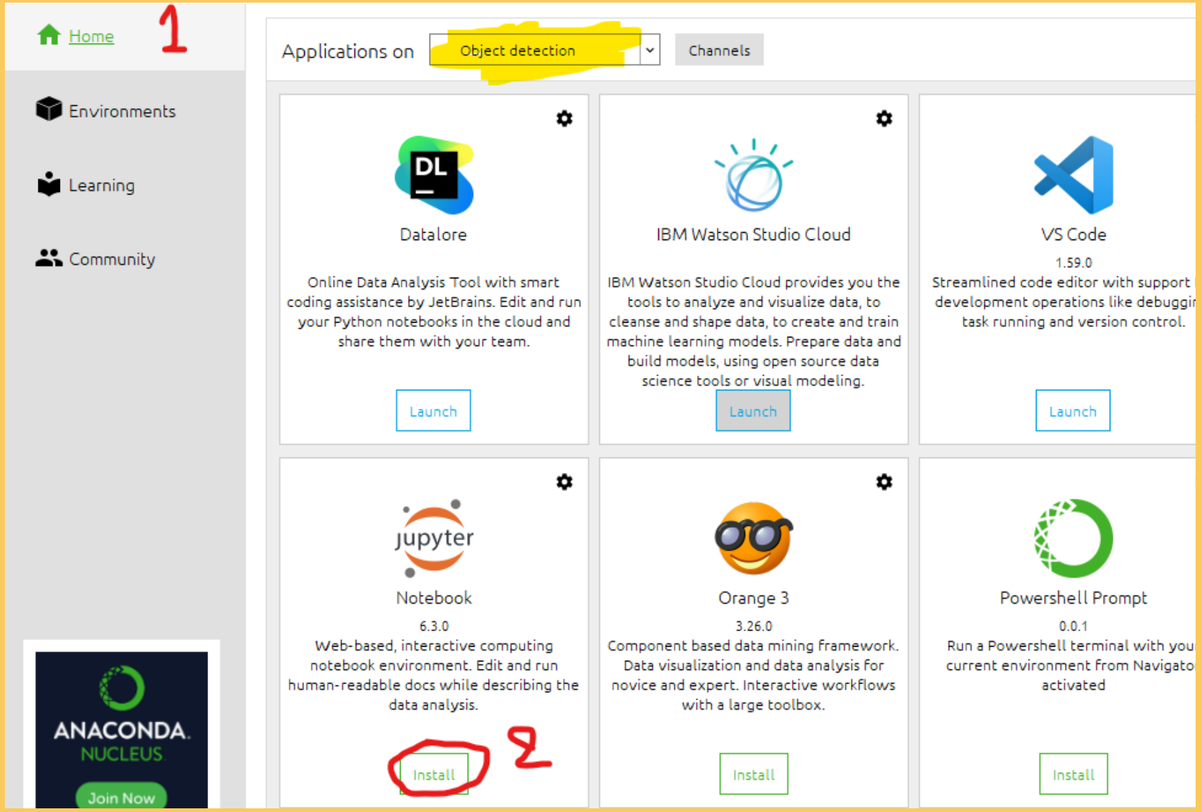

Anaconda3 2021.05 (64-bit) Setup
ANACONDA.
Anaconda3 2021.05 (64-bit)
Anaconda + JetBrains
Working with Python and Jupyter notebooks is a breeze with PyCharm Pro, designed to be used with Anaconda. Download now and have the best data tools at your fingertips.
https://www.anaconda.com/pycharm
ANACONDA.
PC
Anaconda, Inc.
< Back
Next >
Cancel


Anaconda3 2021.05 (64-bit) Setup
ANACONDA.
Completing Anaconda3 2021.05 (64-bit) Setup
Thank you for installing Anaconda Individual Edition.
Here are some helpful tips and resources to get you started. We recommend you bookmark these links so you can refer back to them later.
Anaconda Individual Edition Tutorial
Getting Started with Anaconda
< Back
Finish
Cancel

Anaconda3 (64-bit)
New
Anaconda Navigator (anaconda3)
New
Anaconda Powershell Prompt (anaconda3)
New
Anaconda Prompt (anaconda3)
New
Jupyter Notebook (anaconda3)
New
Reset Spyder Settings (anaconda3)
New
Spyder (anaconda3)
New

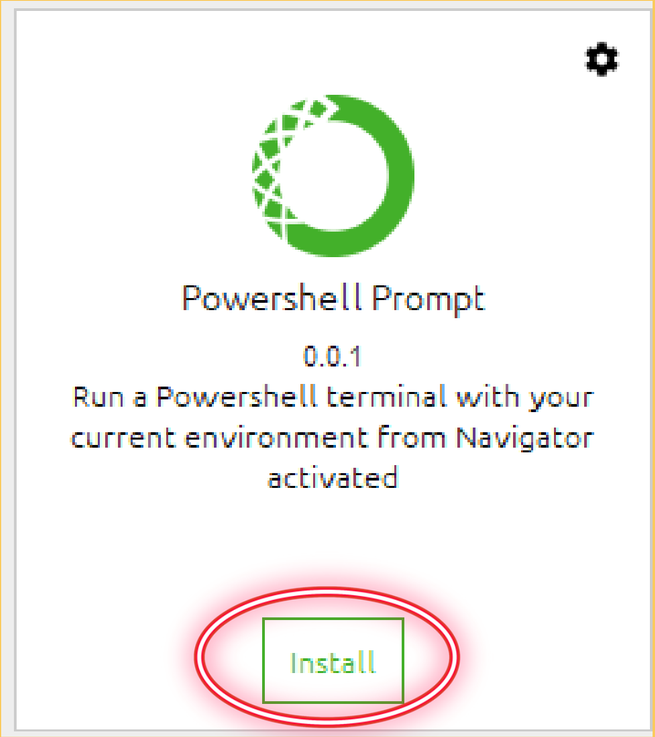
สร้าง Envoriment สำหรับใช้การทำงาน

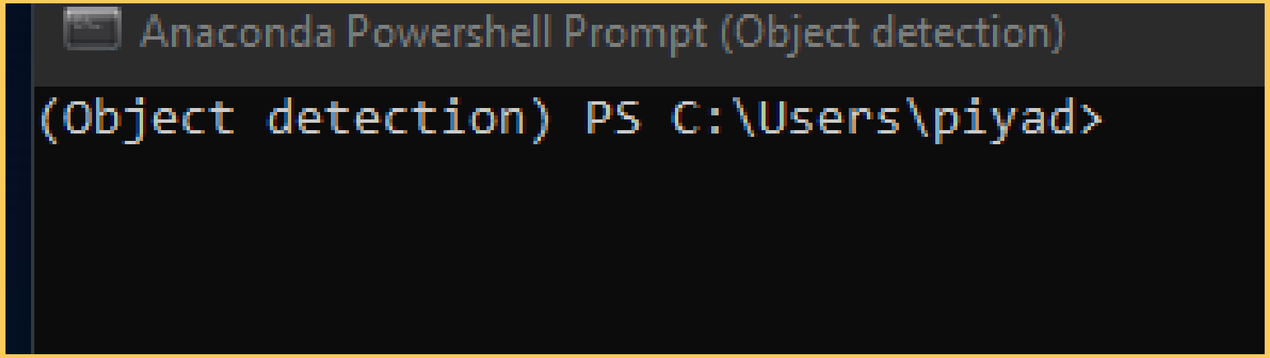




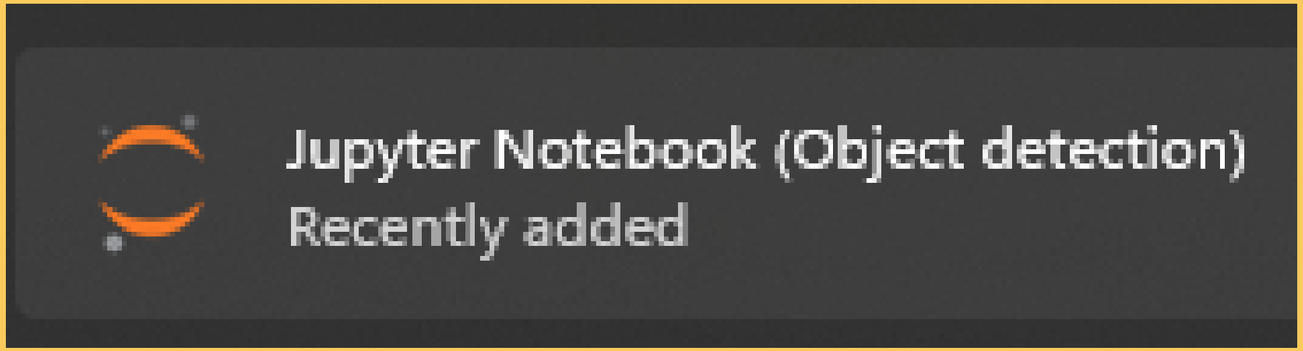
Install application **notebook** on **C:\Users\piyad\anaconda3\envs\Object detection**

สร้าง Envoriment สำหรับใช้การทำงาน

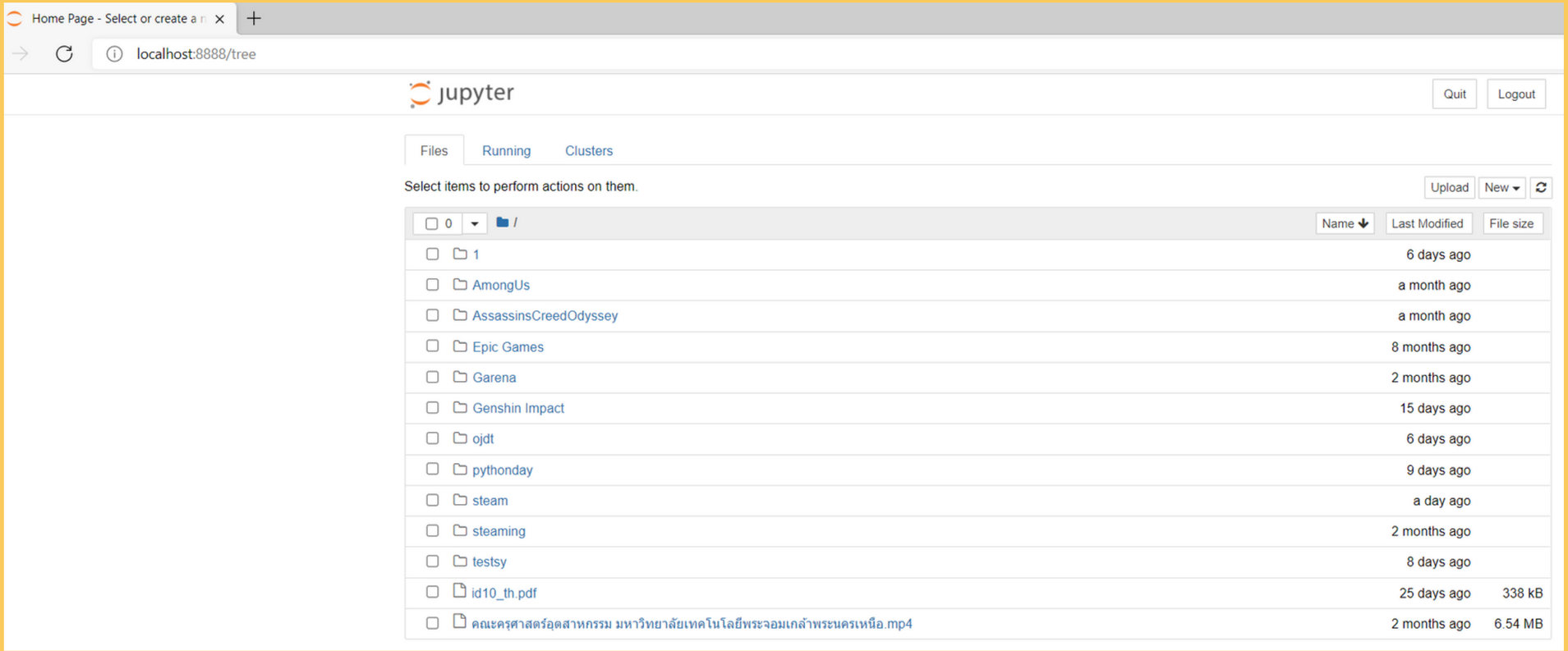




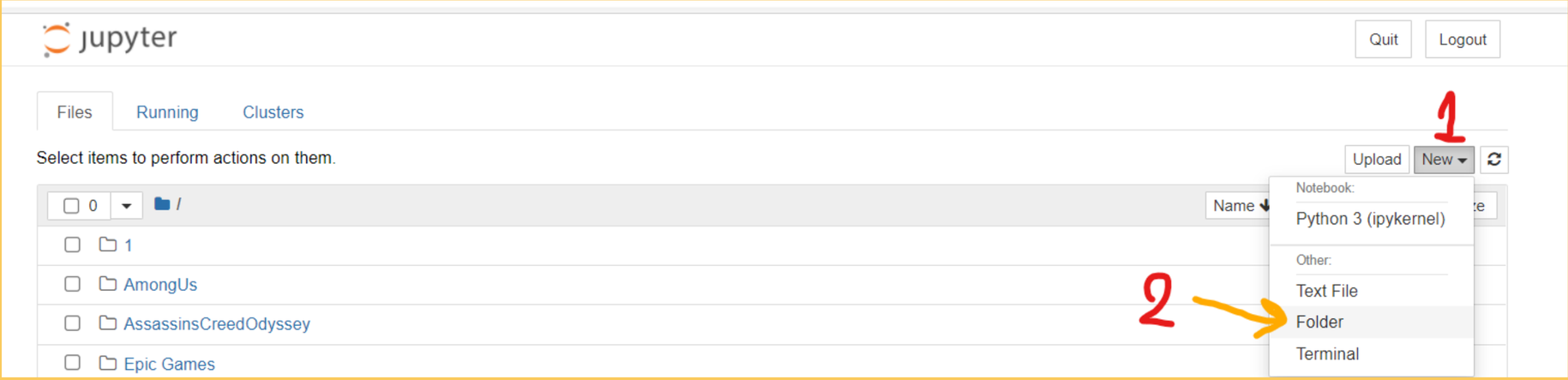
เข้าใช้งาน jupyter Notebook

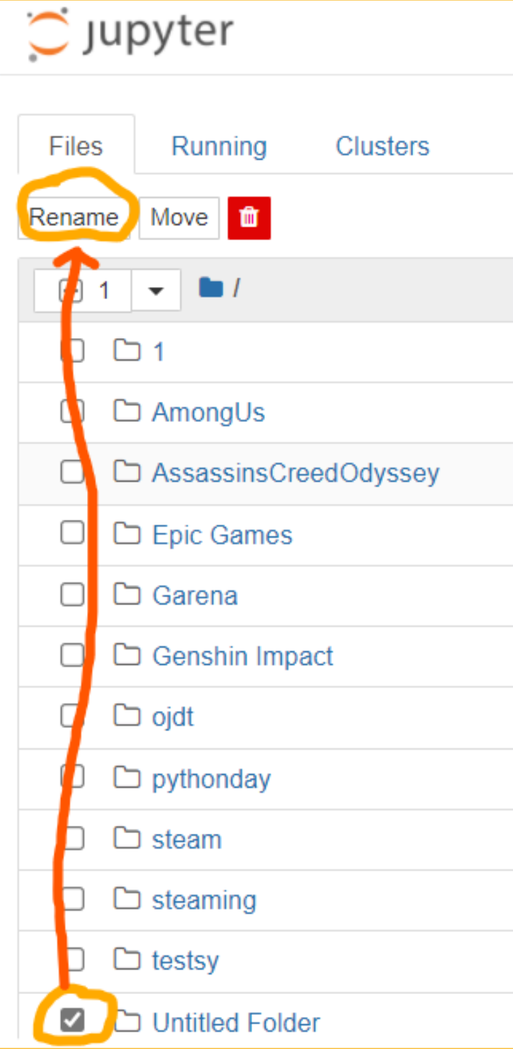


หน้าตาของ Jupyter Notebook



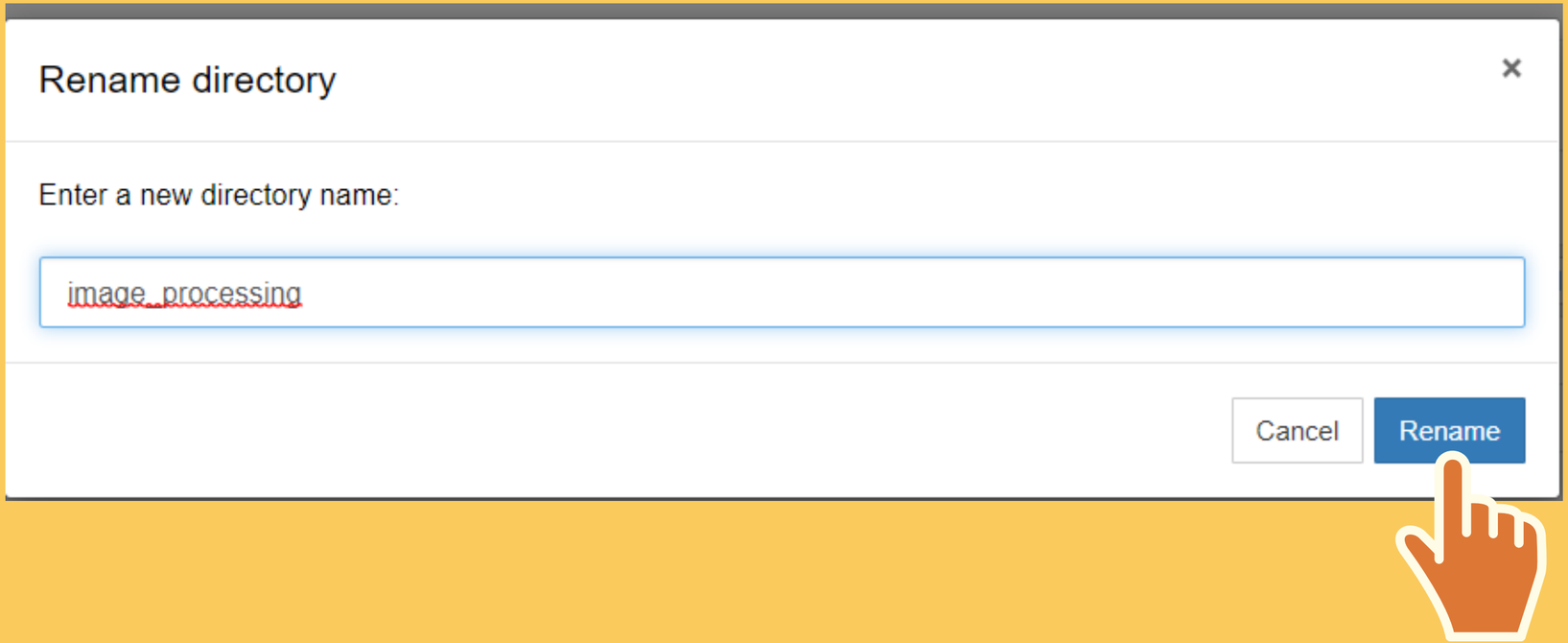
สร้างโฟลเดอร์ เพื่อเก็บชิ้นงาน





เปลี่ยนชื่อโฟล์เดอร์

ตามที่ต้องการแต่ขอแนะนำให้ใช้คำว่า image_processing


Rename directory
Enter a new directory name:
image_processing
Cancel
Rename

image_processing

OpenCV

OpenCV (Open source Computer Vision) เป็นไลบรารีฟังก์ชันการเขียนโปรแกรม (Library of Programming Functions) โดยส่วนใหญ่จะมุ่งเป้าไปที่การแสดงผลด้วยคอมพิวเตอร์แบบเรียลไทม์ (Real-Time Computer Vision)

Opencv

https://pypi.org/project/opencv-python/

Search projects

Help Sponsors Log in Register

opencv-python 4.5.3.56

```
pip install opencv-python
```

Latest version

Released: Jul 11, 2021

Wrapper package for OpenCV python bindings.

Navigation

- Project description
- Release history
- Download files

Project links

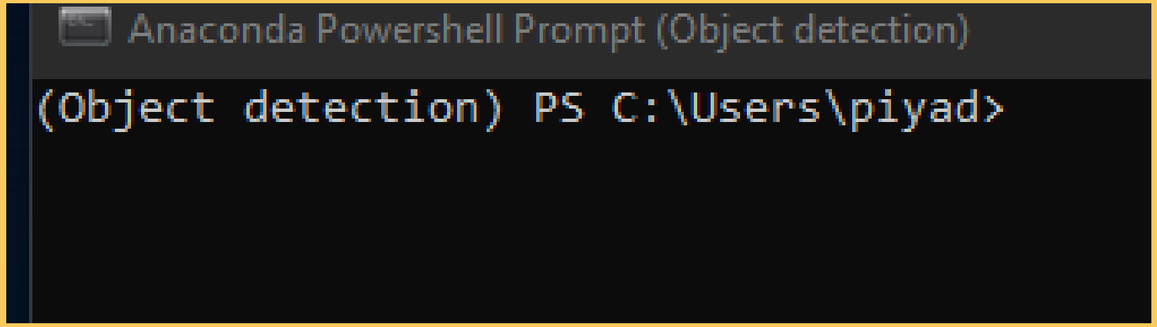
Project description

downloads 119M

OpenCV on Wheels

Pre-built CPU-only OpenCV packages for Python.

Check the manual build section if you wish to compile the bindings from source to enable additional modules such as CUDA.



ขั้นตอนติดตั้ง

เปิด PowerShell

```
Anaconda Powershell Prompt (Object detection)
(Object detection) PS C:\Users\piyad> pip install opencv-python
Collecting opencv-python
  Using cached opencv_python-4.5.3.56-cp38-cp38-win_amd64.whl (34.9 MB)
```

ขั้นตอนติดตั้ง

ป้อนคำสั่ง

pip install opencv-python

NumPy

เป็นชื่อของ library ที่ใช้ในการคำนวนทางคณิตศาสตร์ในภาษา Python ซึ่งภายในถูกเขียนด้วยภาษา C จึงทำงานได้เร็วและมีประสิทธิภาพ โดย NumPy มีความสามารถในการจัดการกับอาเรย์หลายมิติและข้อมูลแบบเมทริกซ์ เราสามารถติดตั้ง NumPy ได้ง่ายผ่าน package installer ของ Python ด้วยคำสั่ง

เป็นแก่นสำคัญของ Deep Learning แล้วยังเป็นแก่นสำคัญของการประยุกต์ใช้ Python ในงานคำนวณต่างๆ ทั้งด้านวิทยาศาสตร์ วิศวกรรมศาสตร์ สถิติ ธุรกิจ กราฟฟิก ฯลฯ อีกด้วย ดังนั้นเรียนรู้ Numpy ไว้มีประโยชน์ต่อชีวิตนักเขียนโปรแกรมมืออาชีพสายคำนวนมากๆ

NUMPY

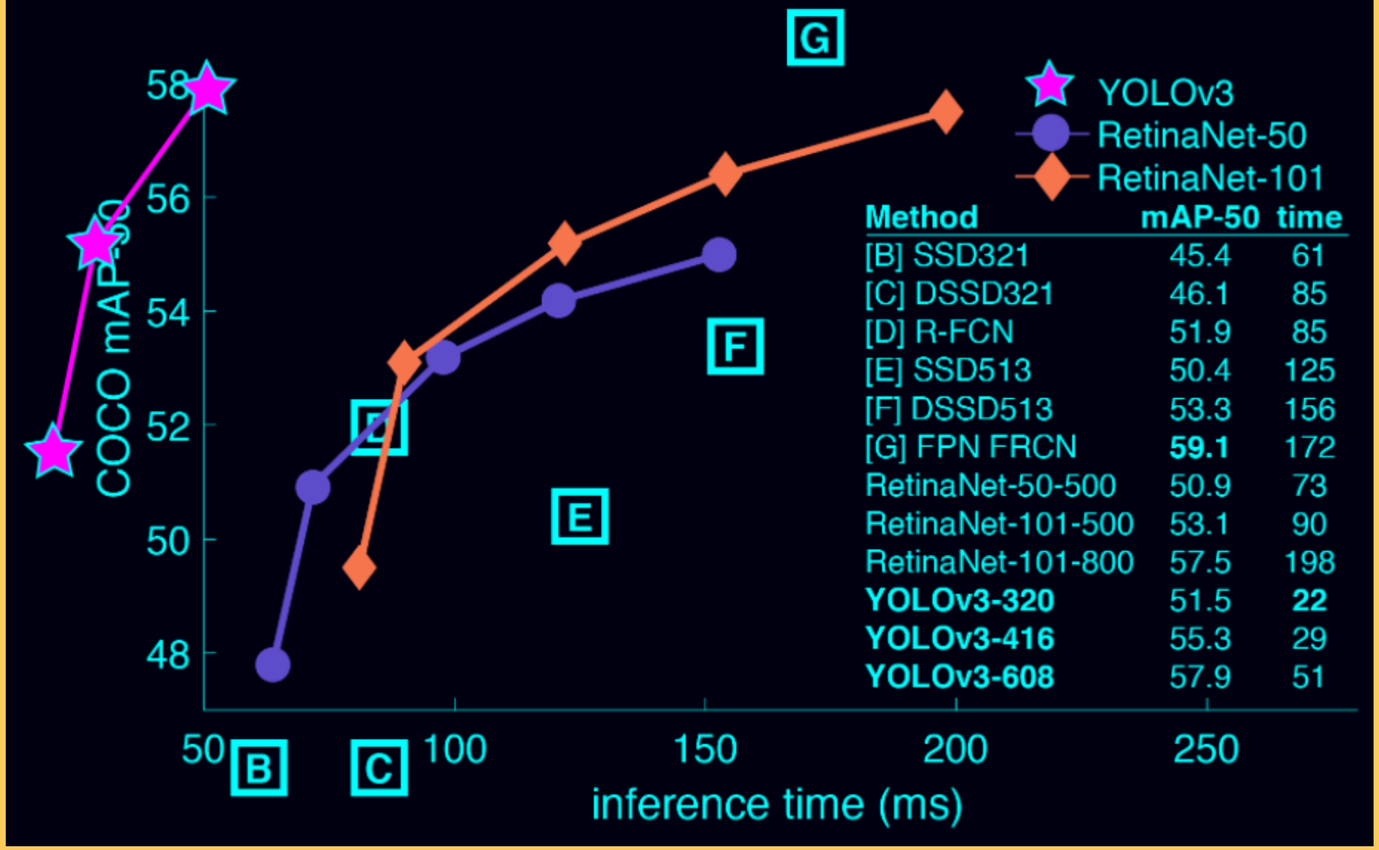
https://numpy.org/install/

YOLO

https://pjreddie.com/darknet/yolo/

Comparison to Other Detectors

YOLOv3 is extremely fast and accurate. In mAP measured at .5 IOU YOLOv3 is on par with Focal Loss but about 4x faster. Moreover, you can easily tradeoff between speed and accuracy simply by changing the size of the model, no retraining required!



Method	mAP-50	time
[B] SSD321	45.4	61
[C] DSSD321	46.1	85
[D] R-FCN	51.9	85
[E] SSD513	50.4	125
[F] DSSD513	53.3	156
[G] FPN FRCN	**59.1**	172
RetinaNet-50-500	50.9	73
RetinaNet-101-500	53.1	90
RetinaNet-101-800	57.5	198
YOLOv3-320	51.5	**22**
YOLOv3-416	55.3	29
YOLOv3-608	57.9	51

SSD321	COCO trainval	test-dev	45.4	-	16		link
DSSD321	COCO trainval	test-dev	46.1	-	12		link
R-FCN	COCO trainval	test-dev	51.9	-	12		link
SSD513	COCO trainval	test-dev	50.4	-	8		link
DSSD513	COCO trainval	test-dev	53.3	-	6		link
FPN FRCN	COCO trainval	test-dev	59.1	-	6		link
Retinanet-50-500	COCO trainval	test-dev	50.9	-	14		link
Retinanet-101-500	COCO trainval	test-dev	53.1	-	11		link
Retinanet-101-800	COCO trainval	test-dev	57.5	-	5		link
YOLOv3-320	COCO trainval	test-dev	51.5	38.97 Bn	45	cfg	weights
YOLOv3-416	COCO trainval	test-dev	55.3	65.86 Bn	35	cfg	weights
YOLOv3-608	COCO trainval	test-dev	57.9	140.69 Bn	20	cfg	weights
YOLOv3-tiny	COCO trainval	test-dev	33.1	5.56 Bn	220	cfg	weights
YOLOv3-spp	COCO trainval	test-dev	60.6	141.45 Bn	20	cfg	weights

YOLOv3-tiny	COCO trainval	test-dev	33.1	5.56 Bn	220	cfg	weights

ดาวน์โหลดทั้งไฟล์ cfg
ลงในโฟลเดอร์ที่สร้างไว้

Why GitHub? Team Enterprise Explore Marketplace Pricing / Sign

pjreddie / darknet

Notifications Star 21.2k Fo

Code Issues 1.7k Pull requests 160 Actions Projects Wiki Security Insight

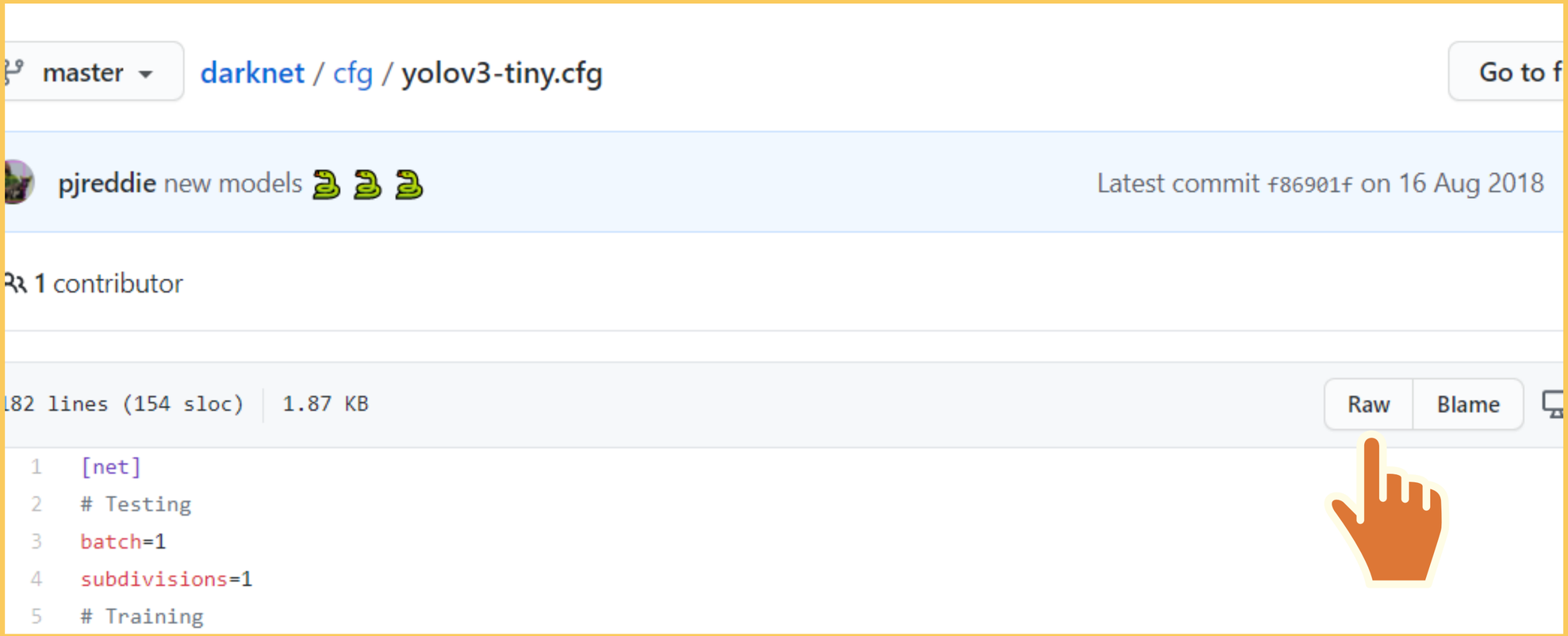
master darknet / cfg / yolov3-tiny.cfg Go to fil

pjreddie new models Latest commit f86901f on 16 Aug 2018

1 contributor

182 lines (154 sloc) 1.87 KB Raw Blame

```
[net]
# Testing
batch=1
```


master
darknet / cfg / yolov3-tiny.cfg
Go to f
pjreddie new models
Latest commit f86901f on 16 Aug 2018
1 contributor
182 lines (154 sloc)
1.87 KB
Raw
Blame
[net]
# Testing
batch=1
subdivisions=1
# Training

กดปุ่ม Raw

https://raw.githubusercontent.com/pj...

```
[net]
# Testing
batch=1
subdivisions=1
# Training
# batch=64
# subdivisions=2
width=416
height=416
channels=3
momentum=0.9
decay=0.0005
angle=0
saturation = 1.5
exposure = 1.5
hue=.1

learning_rate=0.001
burn_in=1000
max_batches = 500200
policy=steps
steps=400000,450000
scales=.1,.1

[convolutional]
batch_normalize=1
filters=16
size=3
stride=1
pad=1
activation=leaky

[maxpool]
size=2
stride=2

[convolutional]
```

ทำการคลิกขวาเพื่อบันทึก
บันทึกลงในโฟล์เดอร์งาน

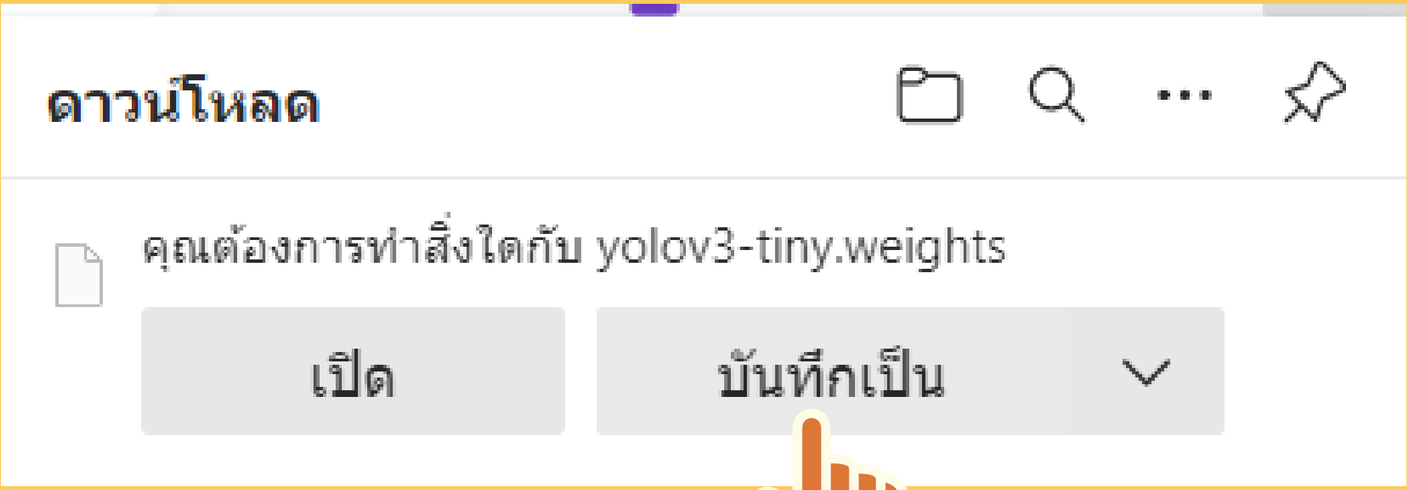
jupyter yolov3.cfg 11 นาทีที่แล้ว

File Edit View Language

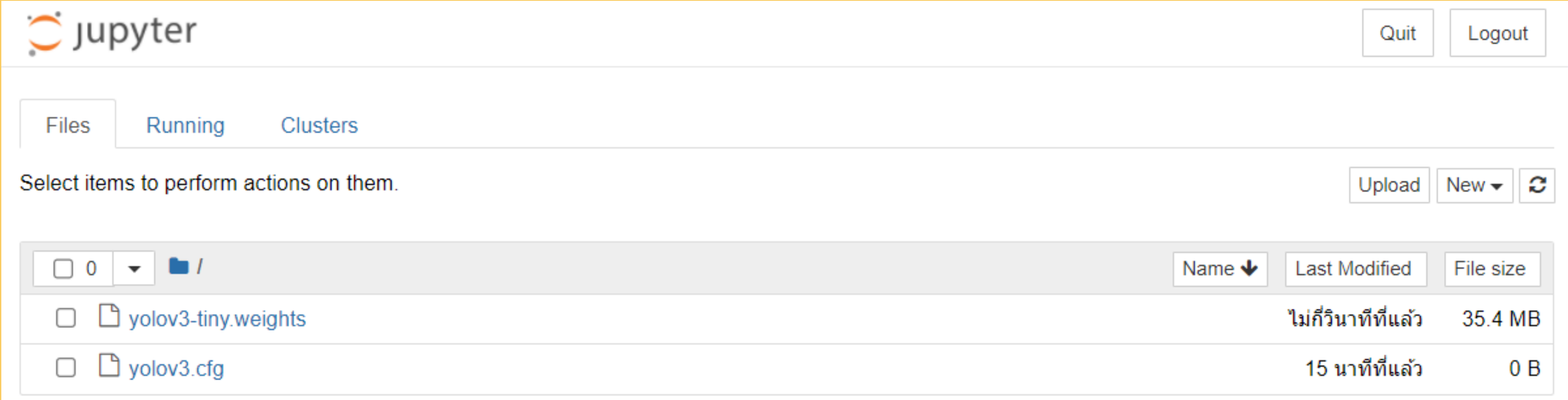
```
[net]
# Testing
batch=1
subdivisions=1
# Training
# batch=64
# subdivisions=2
width=416
height=416
channels=3
momentum=0.9
decay=0.0005
angle=0
saturation = 1.5
exposure = 1.5
hue=.1

learning_rate=0.001
burn_in=1000
max_batches = 500200
policy=steps
steps=400000,450000
```

YOLOv3-tiny	COCO trainval	test-dev	33.1	5.56 Bn	220	cfg	weights

ดาวน์โหลดทั้งไฟล์ weights
ลงในโฟลเดอร์ที่สร้างไว้



ดาวน์โหลดทั้งไฟล์ weights
ลงในโฟลเดอร์ที่สร้างไว้

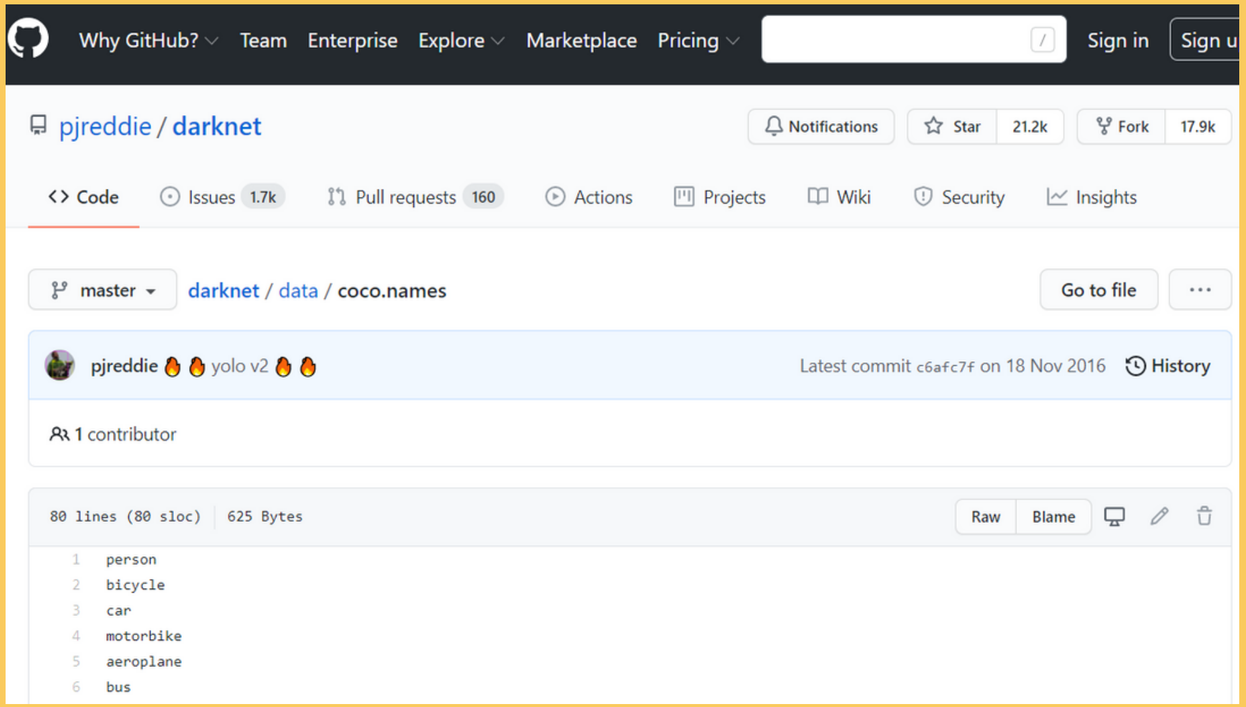


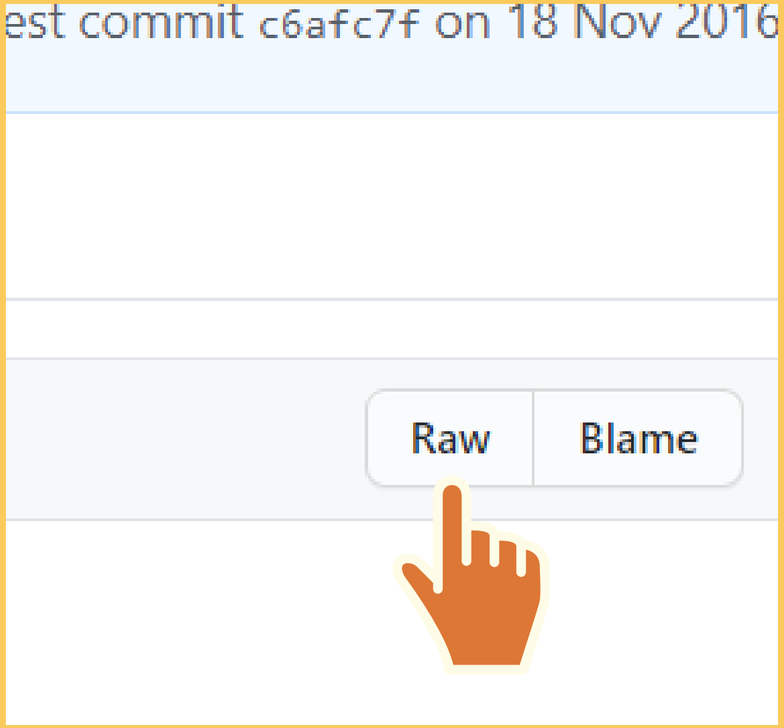
มี 2 ไฟล์ในโฟลเดอร์คือ cfg และ weights

COCO

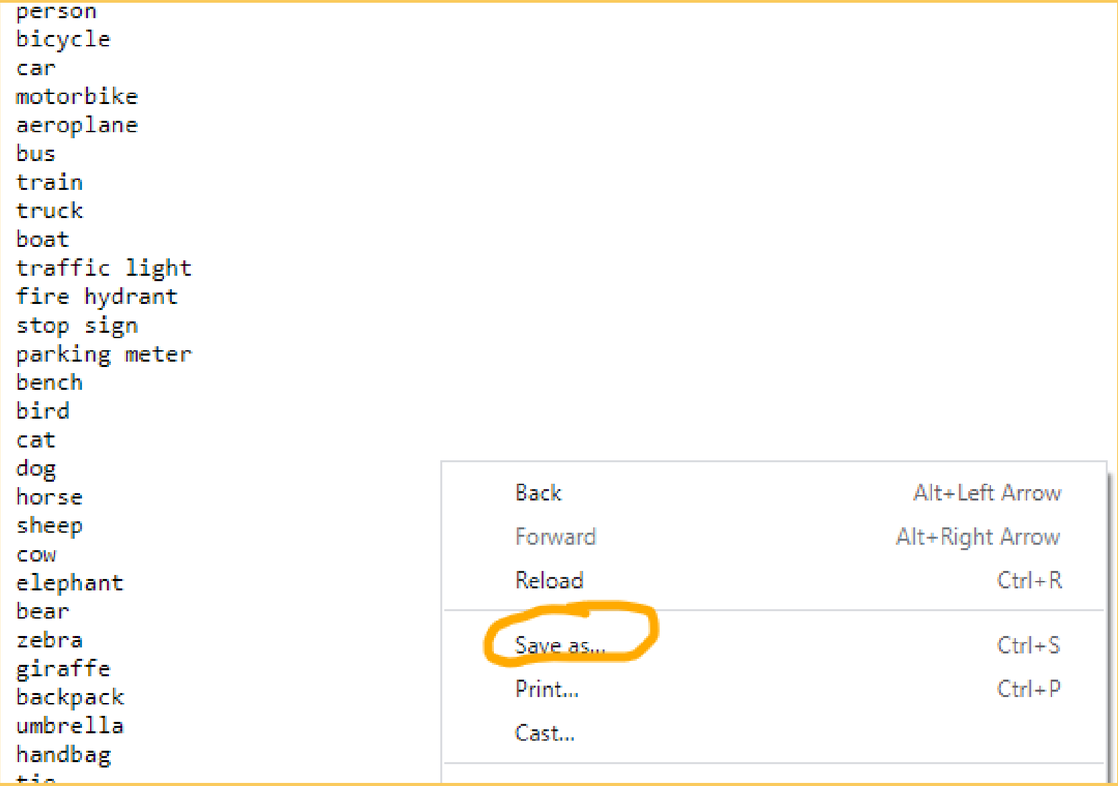
https://github.com/pjreddie/darknet/blob/master/data/coco.names

COCO
Common Objects in Context

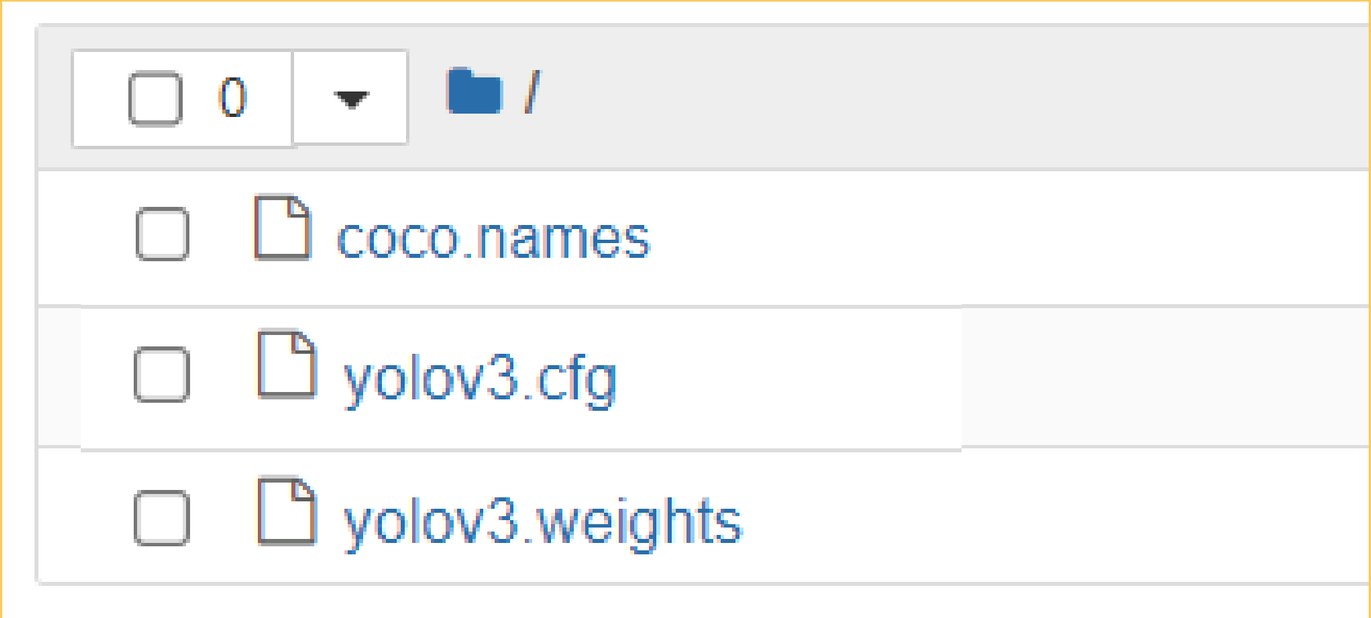

Why GitHub?
Team
Enterprise
Explore
Marketplace
Pricing
Sign in
Sign u
pjreddie / darknet
Notifications
Star 21.2k
Fork 17.9k
Code
Issues 1.7k
Pull requests 160
Actions
Projects
Wiki
Security
Insights
master
darknet / data / coco.names
Go to file
pjreddie yolo v2
Latest commit c6afc7f on 18 Nov 2016
History
1 contributor
80 lines (80 sloc) 625 Bytes
Raw
Blame
1 person
2 bicycle
3 car
4 motorbike
5 aeroplane
6 bus


est commit c6afc7f on 18 Nov 2016
Raw
Blame

กดปุ่ม Raw



ทำการคลิกขวาเพื่อบันทึก
บันทึกลงในโฟล์เดอร์งาน



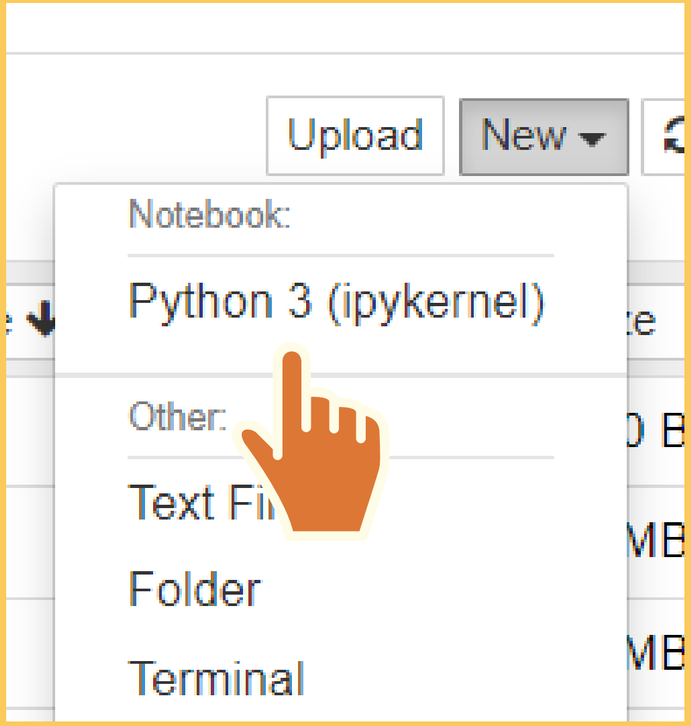
มี 3 ไฟล์ในโฟลเดอร์คือ cfg ,weights และ names

ดาวน์โหลดภาพ

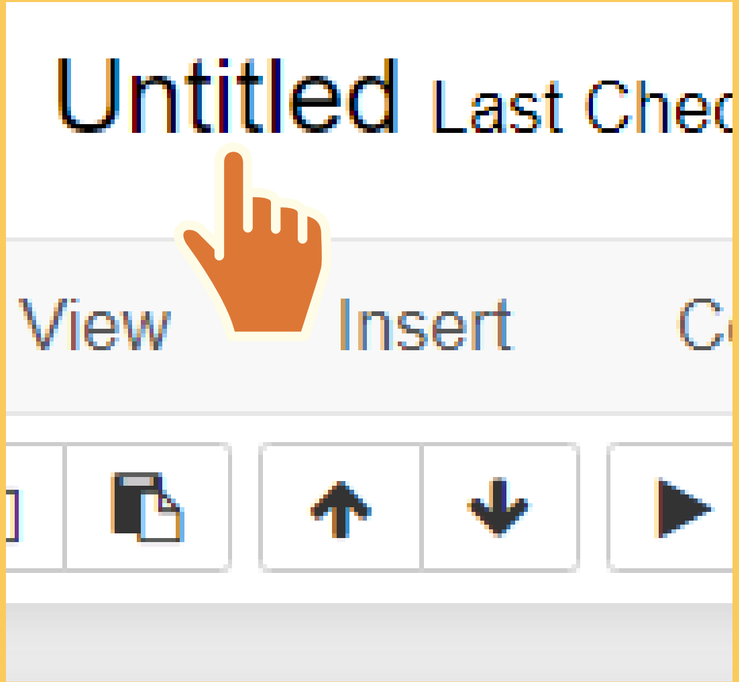
https://bit.ly/2VPF2eo

- coco.names
- example.png
- yolov3-tiny.weights
- yolov3.cfg

มี 4 ไฟล์ในโฟลเดอร์คือ
cfg ,weights ,names และไฟล์ภาพ

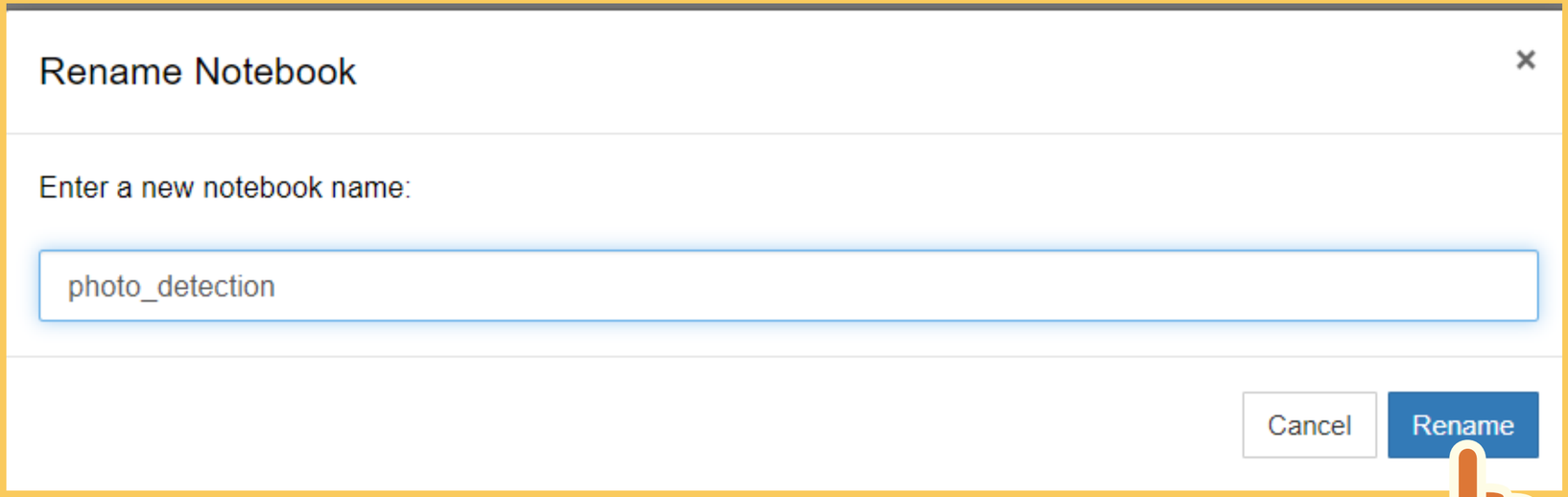


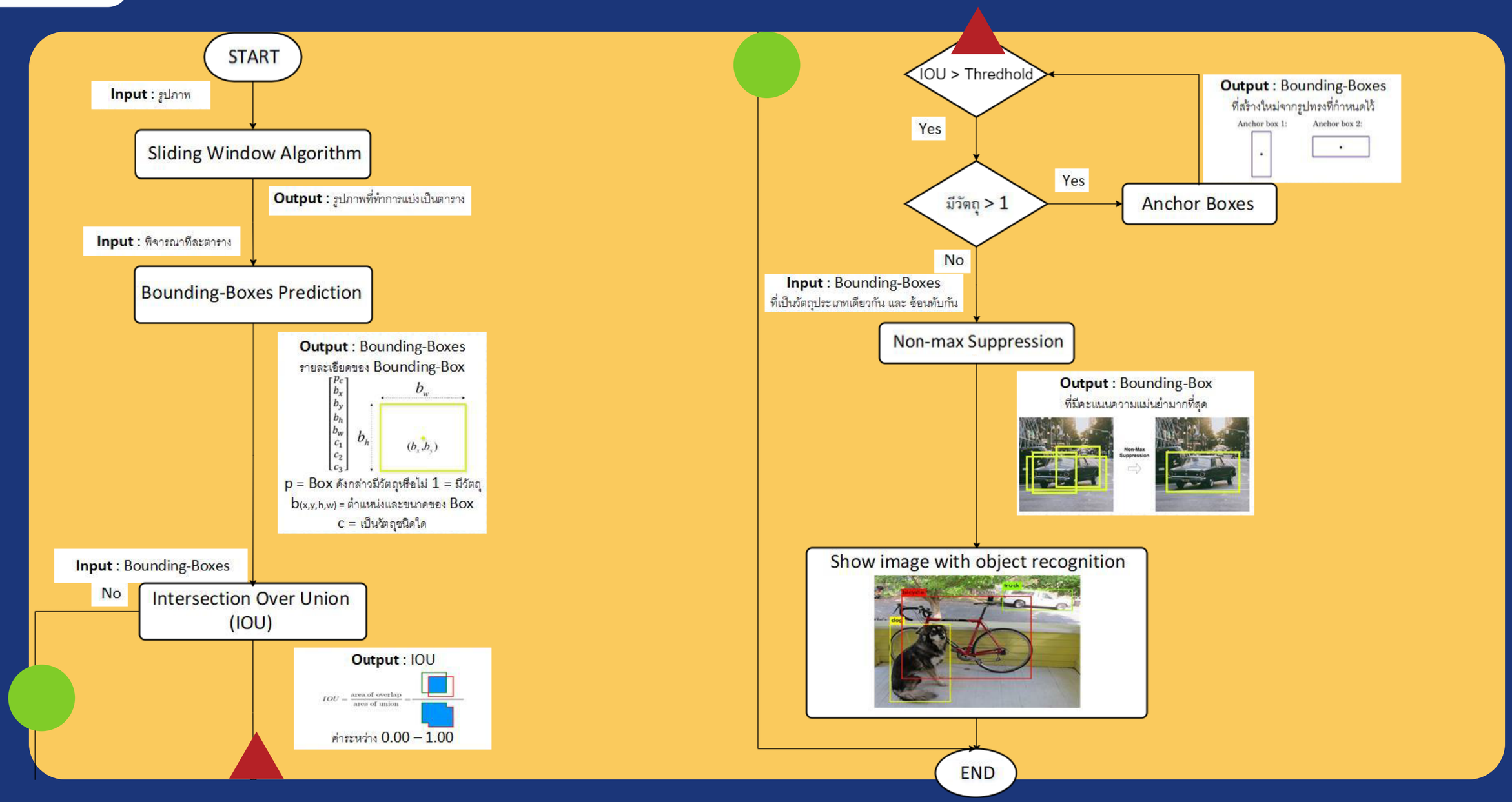
เลือก Python 3



กดเพื่อเปลี่ยนชื่อ

เปลี่ยนชื่อเป็น photo_detection เสร็จสิ้นกดปุ่ม Rename




START
Input : รูปภาพ
Sliding Window Algorithm
Output : รูปภาพที่ทำการแบ่งเป็นตาราง
Input : พิจารณาทีละตาราง
Bounding-Boxes Prediction
Output : Bounding-Boxes
รายละเอียดของ Bounding-Box
p = Box ดังกล่าวมีวัตถุหรือไม่ 1 = มีวัตถุ
b(x,y,h,w) = ตำแหน่งและขนาดของ Box
c = เป็นวัตถุชนิดใด
Input : Bounding-Boxes
No
Intersection Over Union (IOU)
Output : IOU
ค่าระหว่าง 0.00 – 1.00
IOU > Thredhold
Yes
มีวัตถุ > 1
Yes
Anchor Boxes
Output : Bounding-Boxes
ที่สร้างใหม่จากรูปทรงที่กำหนดไว้
Anchor box 1:
Anchor box 2:
No
Input : Bounding-Boxes
ที่เป็นวัตถุประเภทเดียวกัน และ ซ้อนทับกัน
Non-max Suppression
Output : Bounding-Box
ที่มีคะแนนความแม่นยำมากที่สุด
Non-Max Suppression
Show image with object recognition
bicycle
truck
dog
END

สรุป

- Yolo คืออะไร
- กระบวนการทำงาน Yolo
- การติดตั้งเครื่องมือ
    - python
    - Jupiter notebook
    - Library

เริ่มเขียน code กันเถอะ

```
In [2]: import numpy as np
        import time
        import cv2
```

นำเข้า Libraly ต่าง ๆที่ได้ติดตั้งไปก่อนหน้านี้

```
INPUT_FILE='example.png'
LABELS_FILE='coco.names'
CONFIG_FILE='yolov3.cfg'
WEIGHTS_FILE='yolov3-tiny.weights'
CONFIDENCE_THRESHOLD=0.3
```

INPUT_FILE คือ ไฟล์ที่ต้องการนำมา Detection

LABELS_FILE คือ ไฟล์ข้อมูลแอตทริบิวต์ต่าง ๆ ของวัตถุ

CONFIG_FILE คือ ไฟล์กำหนดค่าของโมเดล YOLO

WEIGHT_FILE คือ ไฟล์ที่ผ่านการ train model YOLO

CONFIDENCE_THRESHOLD คือ เกณฑ์ความน่าจะเป็นวัตถุในภาพ

```
LABELS = open(LABELS_FILE).read().strip().split("\n")
```

คำสั่งให้นำ LABEL FILE ซึ่งตอนนี้เป็น COCO.NAMES ให้ระบบนำไปแกะออกมานำไปบรรจุใน LABEL โดยให้เรียงเป็นบรรทัดเอาไว้ (มองไม่เห็น)

```
np.random.seed(4)
COLORS = np.random.randint(0, 255, size=(len(LABELS), 3),
    dtype="uint8")
```

คำสั่งให้สร้างค่า seed ที่ถูกสุ่มขึ้นมาจากนั้นนำไปสุ่มสีที่แตกต่างกัน สำหรับนำไปแสดงเป็นกรอบรอบชิ้นวัตถุที่ถูก Detection

```
net = cv2.dnn.readNetFromDarknet(CONFIG_FILE, WEIGHTS_FILE)
```

คำสั่งให้สร้างตัวแปร net ขึ้นมาเพื่อนำค่า CONFIG และ WEIGHT ส่งไปยังเซิร์ฟเวอร์ของ Darknet เพื่อนำมาคำนวณในโปรแกรมต่อไป

```
image = cv2.imread(INPUT_FILE)
(H, W) = image.shape[:2]
```

คำสั่งสร้างตัวแปร image ขึ้นมาเพื่อค่าของรูปภาพตัวอย่างที่นำเข้ามา

คำสั่งสร้างตัวแปร (H, W) เพื่อเก็บความสูงและความกว้างของภาพไว้

```
ln = net.getLayerNames()
ln = [ln[i[0] - 1] for i in net.getUnconnectedOutLayers()]
```

คำสั่งกำหนดให้ Output เฉพาะส่วนที่ต้องการจาก YOLO

```
blob = cv2.dnn.blobFromImage(image, 1 / 255.0, (416, 416),
    swapRB=True, crop=False)
net.setInput(blob)
start = time.time()
layerOutputs = net.forward(ln)
end = time.time()
```

สร้างตัวแปร blob นำไปลดขนาดรูปภาพที่นำเข้ามาทดสอบให้เหลือ 416*416 ให้สลับสีฟ้ากับน้ำเงิน ให้ส่งค่าไป net.setInput เริ่มนับเวลาเพื่อที่จะคำนวณว่า YOLO ใช้เวลาในการคำนวณไปเท่าไหร่

```
print("[INFO] YOLO took {:.6f} seconds".format(end - start))
```

แสดงค่าเวลาที่ถูกใช้ไปในการคำนวณ YOLO ขึ้นมาในช่อง OUTPUT

```
boxes = []
confidences = []
classIDs = []
```

ประการตัวแปร Array ทั้งหมด 3 ตัวคือ boxes, confidences, classIDs

```
for output in layerOutputs:
```

คำสั่ง วนซ้ำแต่ละเอาต์พุตของเลเยอร์

```
for detection in output:
```

คำสั่ง วนซ้ำการตรวจจับแต่ละครั้ง

```
scores = detection[5:]
classID = np.argmax(scores)
confidence = scores[classID]
```

คำสั่ง แยก class ID และ ค่า confidence (เช่น ความน่าจะเป็น) ของการตรวจจับวัตถุปัจจุบัน

```
if confidence > CONFIDENCE_THRESHOLD:
```

หากค่า confidence ที่คาดการณ์ว่าตรวจจับมีค่ามากกว่าเกณฑ์ความน่าจะเป็นวัตถุในภาพ

```
box = detection[0:4] * np.array([W, H, W, H])
(centerX, centerY, width, height) = box.astype("int")
```

ปรับขนาด bounding box รูปโดยสัมพันธ์กับขนาดของรูปภาพ YOLO จะส่งตำแหน่งตรงกลางbounding box (x, y) ของกรอบ ล้อมรอบตามด้วยความกว้างและความสูงของกล่อง

```
x = int(centerX - (width / 2))
y = int(centerY - (height / 2))
```

ใช้จุดตำแหน่งตรงกลาง bounding box (x, y) - ที่มีการ detection เพื่อหามุมด้านบนและด้านซ้ายของ bounding box

```
boxes.append([x, y, int(width), int(height)])
confidences.append(float(confidence))
classIDs.append(classID)
```

อัปเดตรายการ bounding box confidence และ classID

```
idxs = cv2.dnn.NMSBoxes(boxes, confidences, CONFIDENCE_THRESHOLD,
    CONFIDENCE_THRESHOLD)
```

คำสั่งสร้างตัวแปร idxs เก็บค่า boxes, confidence, CONFIDENCE_THRESHOLD, CONFIDENCE_THRESHOLD

```
if len(idxs) > 0:
```

คำสั่งถ้ามีการตรวจสอบว่ามีการตรวจพบ idxs อย่างน้อยหนึ่งรายการ

```
for i in idxs.flatten():
```

คำสั่ง วนรอบซ้ำดัชนีที่เราเก็บไว้ใน idxs

```
(x, y) = (boxes[i][0], boxes[i][1])
(w, h) = (boxes[i][2], boxes[i][3])
```

แยกค่า bounding box กล่องขอบเขต

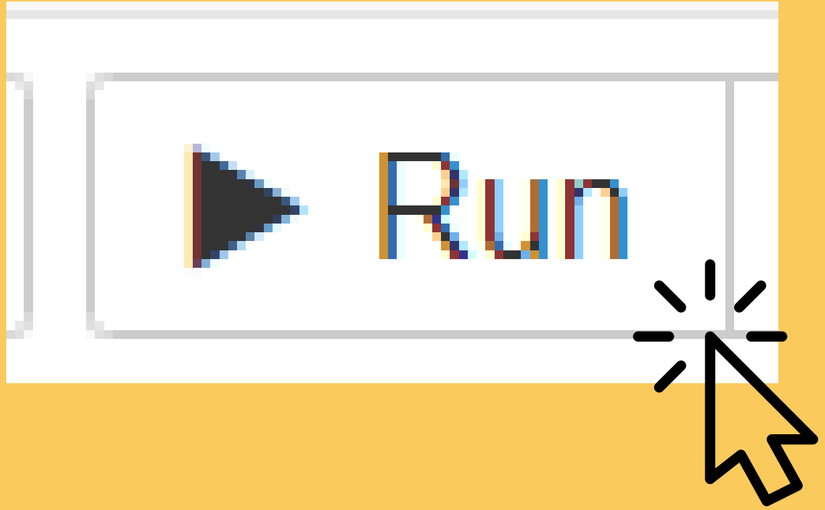
```
color = [int(c) for c in COLORS[classIDs[i]]]

cv2.rectangle(image, (x, y), (x + w, y + h), color, 2)
text = "{}: {:.4f}".format(LABELS[classIDs[i]], confidences[i])
cv2.putText(image, text, (x, y - 5), cv2.FONT_HERSHEY_SIMPLEX
    0.5, color, 2)
```

****

```
cv2.imshow('Image',image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

คำสั่ง แสดง output image

ลองกดทดสอบกัน



ภาพ ที่ผ่านการ detection

Workshop 1 ❤

ให้ผู้อบรมเลือกภาพในโฟลเดอร์ตาม
ลิ้งค์ด้านล่าง จากนั้นไปแก้โค๊ดส่วนนึง
เพื่อให้ YOLO ทำการ Detection

https://bit.ly/3iPakL6

Workshop 1 ❤

ผู้เข้าอบรมต้องส่งงานโดยการ Print Screen จากนั้น Save ส่งภาพมาตามฟอร์มลิ้งค์ด้านล่าง

https://bit.ly/3iLl0Kz

Workshop 2 ❤

ให้ผู้อบรมเลือกวัตถุที่อยู่ใกล้ๆตัวมา
เพื่อให้ YOLO ทำการ Detectionโดย
ให้หาวัตถุมาไว้ในภาพมากกว่า2ชิ้น

Workshop 2 ❤

ผู้เข้าอบรมต้องส่งงานโดยการ Print Screen จากนั้น Save ส่งภาพมาตามฟอร์มลิ้งค์ด้านล่าง

https://bit.ly/39tMHSI