嵌入式系統總整與實作

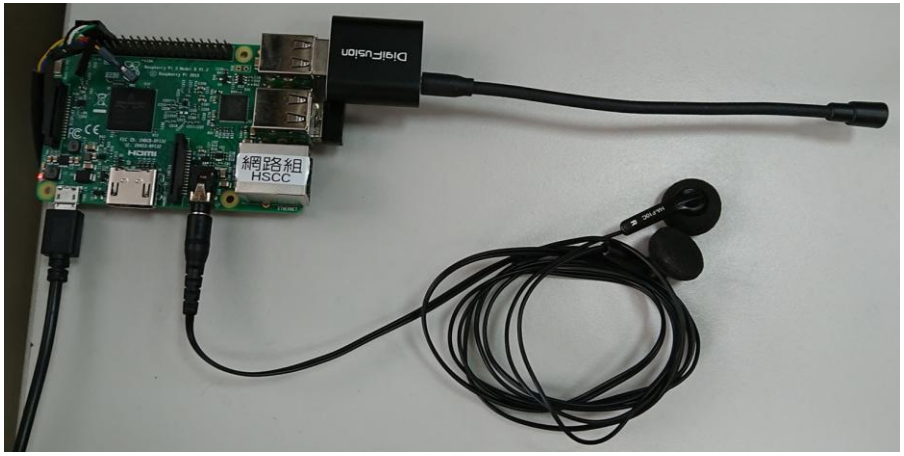
曾煜棋、吳昆儒

National Yang Ming Chiao Tung University

	日期	主題
	2/21	0. 課程介紹
	2/28	梅竹賽!!
	3/7	1. 嵌入式開發板 - 樹莓派介紹與設定 (headless)
	3/14	2. 連接感測器 (GPIO, I2C)
	3/21	3. 整合感測資訊 (IMU)
	3/28	4. 整合音訊資料 (麥克風, 語音識別)
	4/4	清明節放假
期中考周 (4/7-4/11)	4/11	5. 整合視覺資料 (攝影機, 影像辨識)
	4/18	期中考Midterm, Project分組
	4/25	6. 嵌入式模型 (mediapipe, video, audio, text)
	5/2	7. 喚醒詞原理 (by 台灣樹莓派)
	5/9	Final Project – Proposal 分組報告 (online?)
	5/16	8. 樹莓派核心編譯 (Cross compile, Kernel)
	5/23	9. 嵌入式套件編譯
	5/30	端午節放假
期末考周 (6/2-6/6)	6/6	Final project準備周, Q&A (學期考試周)
	6/13	**Final Project demonstration**

Last week

- 嵌入式應用: 語音識別
  1. Audio feature
  2. Speech to text (STT)
  3. Text to speech (TTS)



- 語音識別 (Speech recognition)
  - 自動語音辨識 (Automatic Speech Recognition, ASR)
  - 語音轉文字識別 (Speech To Text, STT)
  - 自然語言處理 (Natural Language Processing, NLP)
  - 語音合成 (Text to Speech, TTS)
  - 大語言模型 (Large Language Model, LLM)

This week

- 整合視覺資料 (攝影機, 影像辨識)
  - Part1: 安裝相機模組
  - Part2: 人臉偵測, 輪廓偵測
  - Part3: 物件辨識 (tensorflow, YOLO)

- Install dependency
  - sudo apt install python3-opencv
  - pip install imutils
  - pip install dlib

Part1: 安裝相機模組

https://cdn.sparkfun.com/datasheets/Dev/RaspberryPi/ov5647_full.pdf

PI Camera Spec.

- Sensor: OmniVision OV5647 (5MP)
- 靜態拍照最高解析度:2592 x 1944 pixel
- Pixel Size:1.4 x 1.4 µm
- Lens: f=3.6 mm, f/2.9
- Angle of View:54 x 41 degrees
- Field of View:2.0 x 1.33 m at 2 m
- Fixed Focus:1m to infinity
- 動態攝影最高解析度:
  1080p@30 FPS with H.264/AVC

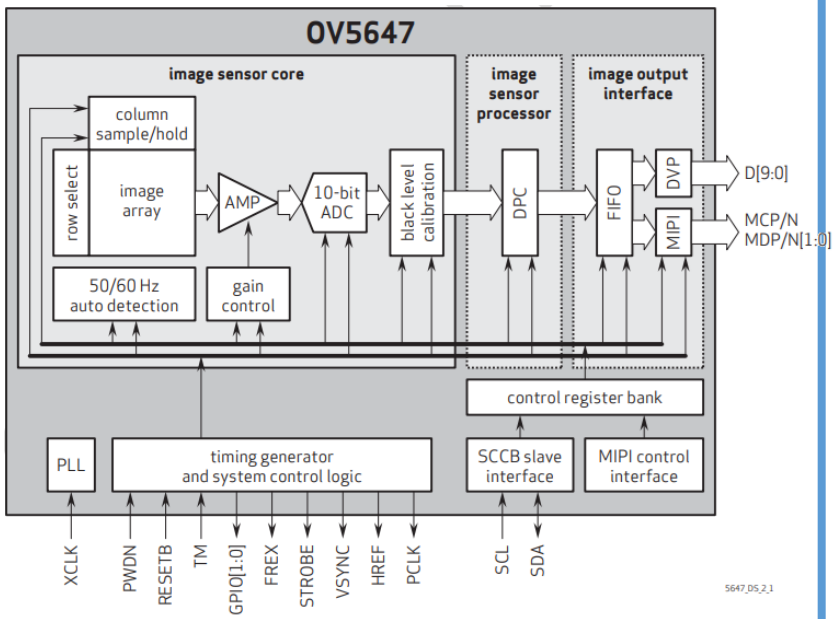
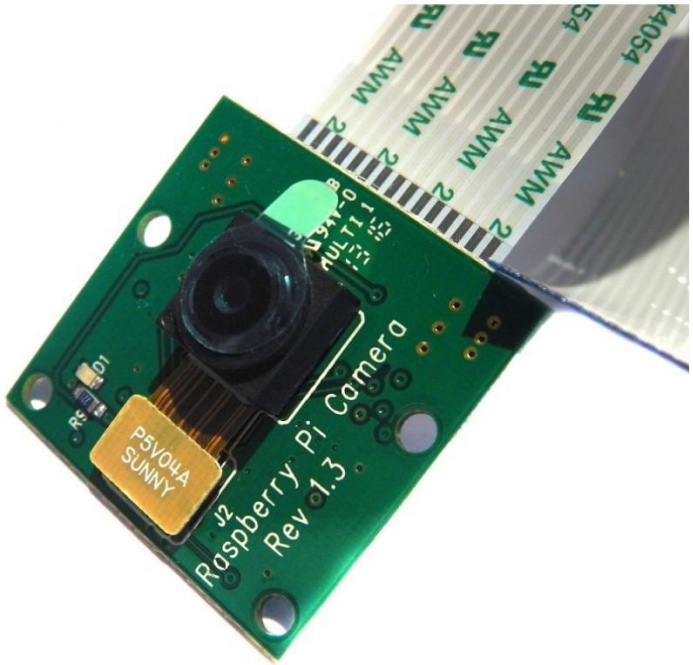


table 2-1 format and frame rate

format	resolution	frame rate	scaling method	pixel clock
5 Mpixel	2592x1944	15 fps	full resolution	80 MHz
1080p	1920x1080	30 fps	cropping	68 MHz
960p	1280x960	45 fps	cropping, subsampling/ binning	91.2 MHz
720p	1280x720	60 fps	cropping, subsampling/ binning	92 MHz
VGA	640x480	90 fps	cropping, subsampling/ binning	46.5 MHz
QVGA	320x240	120 fps	cropping, subsampling/ binning	32.5 MHz

Install PI camera

15-Pins, CSI interface

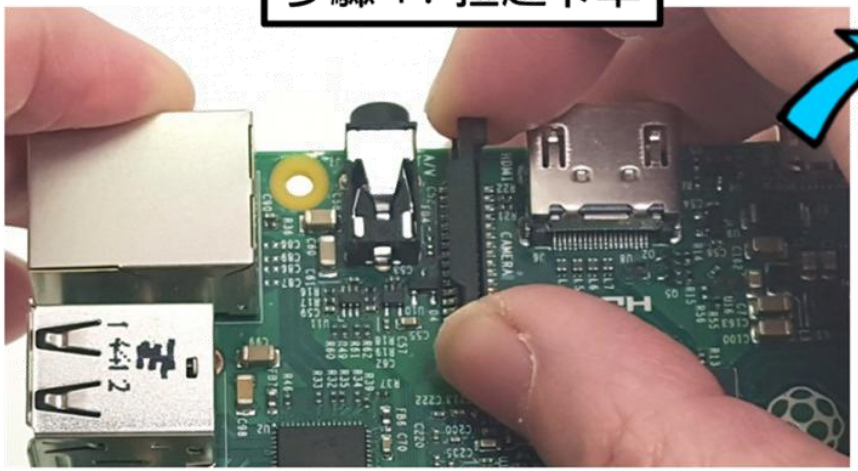


https://www.dexterindustries.com/howto/installing-the-raspberry-pi-camera/

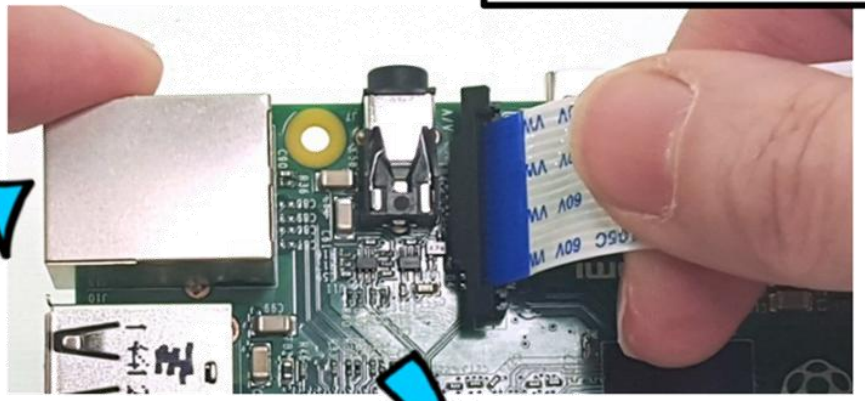
在關機的狀態下安裝 Camera

關機指令： $ sudo poweroff

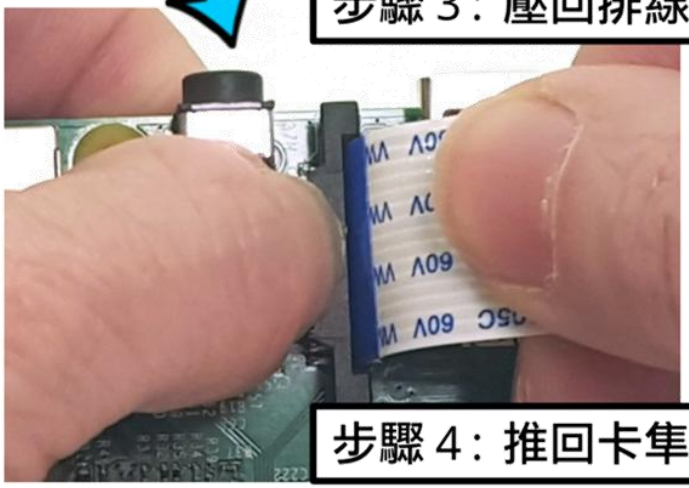
步驟 1：拉起卡隼



步驟 2：插入排線



步驟 3：壓回排線



步驟 4：推回卡隼

安裝完畢標準：
1. 排線已經插到底
2. 排線不能一拉就起來
3. 露出的金屬線和卡隼是水平的

(By 台灣樹莓派)

Camera commands

- Take a picture: raspistill
  - 3秒後拍照, 並編碼成png格式,長640x寬480, 無預覽
  - raspistill -n -t 3000 -o test.png -e png -w 640 -h 480
    - n: Do not display a preview window
    - t: timeout, Time before the camera takes picture and shuts down
    - o: output filename
    - e: Encoding to use for output file (jpg, bmp, gif, and png)
    - w: Set image width <size>
    - h: Set image height <size>

Camera commands

- Record a clip: raspivid
  - 錄5秒的1080p30影片, 長640x寬480, 無預覽
  - raspivid -n -t 5000 -w 640 -h 480 -o video.h264
    - t: Time (in ms) to capture for. Default = 5 sec.
    - o: output filename
    - w: Set image width <size>
    - h: Set image height <size>

- Official document
  - https://www.raspberrypi.com/documentation/accessories/camera.html

Error message?

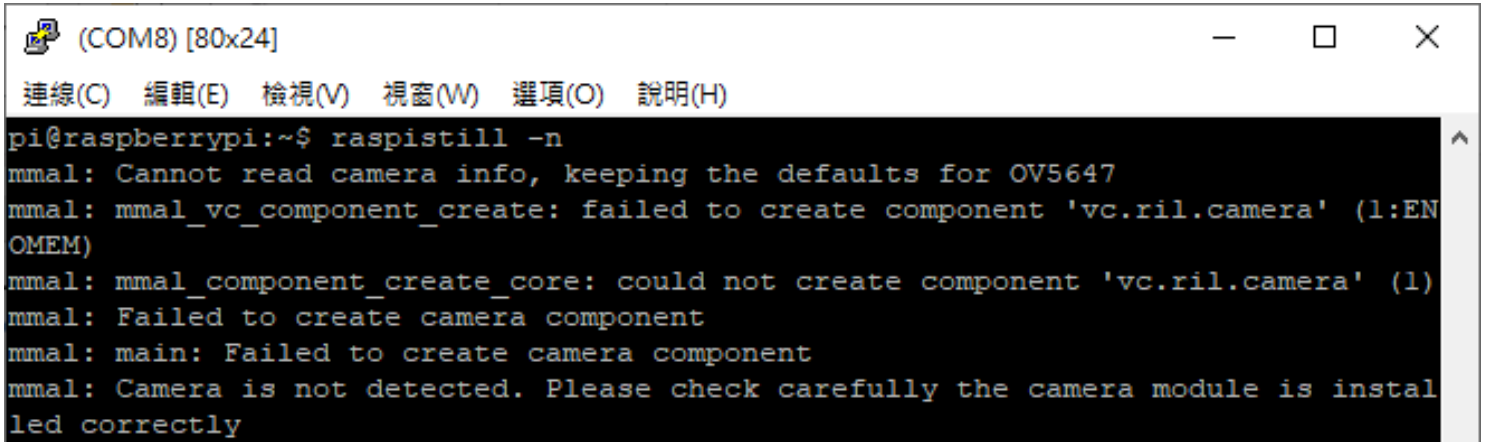
- Msg: Camera is not enabled in this build

- Sol: go to “sudo raspi-config”, then enable camera

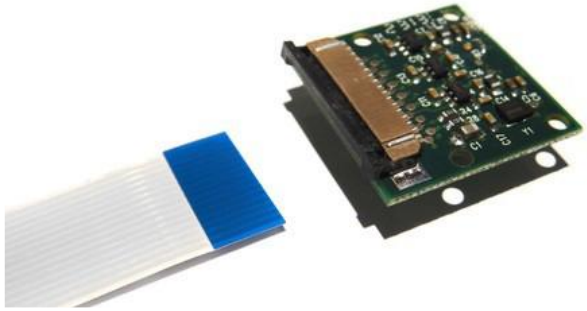
https://www.modmypi.com/blog/how-to-replace-the-raspberry-pi-camera-cable

Error message?

- Msg: Camera is not detected



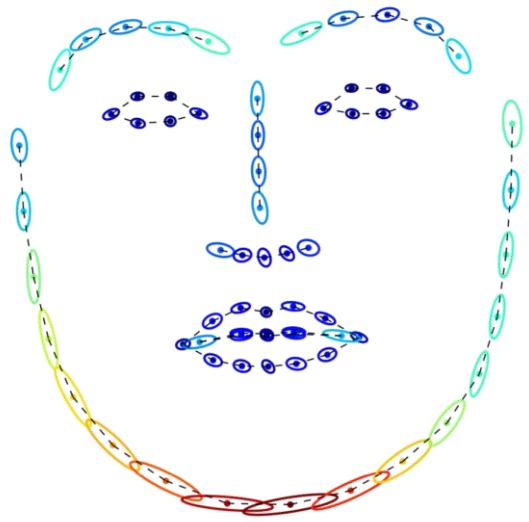
- Sol:
  - 重新安裝camera,或是更換排線
  - 或是檢查camera module是否鬆脫
  - 換一個相機模組試試

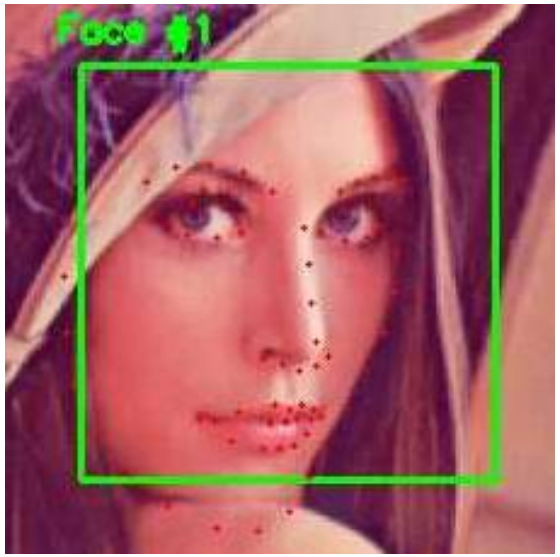


Part2: 人臉偵測, 輪廓偵測

人臉偵測, 輪廓偵測

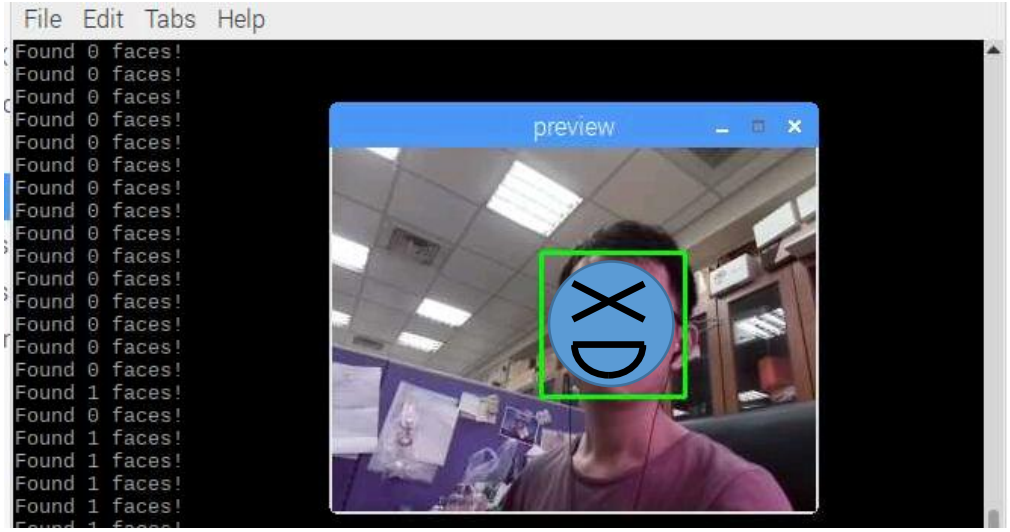
1. 人臉偵測 (by Haar Feature)
2. 輪廓偵測 (by Haar Feature and machine learning)





A. Facial detection

- Two sample code (go to 1.facial)
  - Analyze image
    - python 1.1image_face_detect.py
  - Analyze stream from camera (press q or ctrl+c to stop)
    - python 1.2camera_face_detect.py



A. Facial detection

- python 1.1image_face_detect.py

```
import sys
import cv2

imagePath = "img.jpg"

# Create the haar cascade
cascPath = "model/haarcascade_frontalface_default.xml"
faceCascade = cv2.CascadeClassifier(cascPath)

# Read the image
image = cv2.imread(imagePath)
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
```

A. Facial detection

```
# Detect faces in the image
faces = faceCascade.detectMultiScale(
    gray,
    scaleFactor=1.1,
    minNeighbors=5,
    minSize=(30, 30)
)

print "Found {0} faces!".format(len(faces))

# Draw a rectangle around the faces
for (x, y, w, h) in faces:
    cv2.rectangle(image, (x, y), (x+w, y+h), (0, 255, 0), 2)

cv2.imshow("preview", image)
cv2.waitKey(0)

cv2.destroyAllWindows()
```

Face detection flow

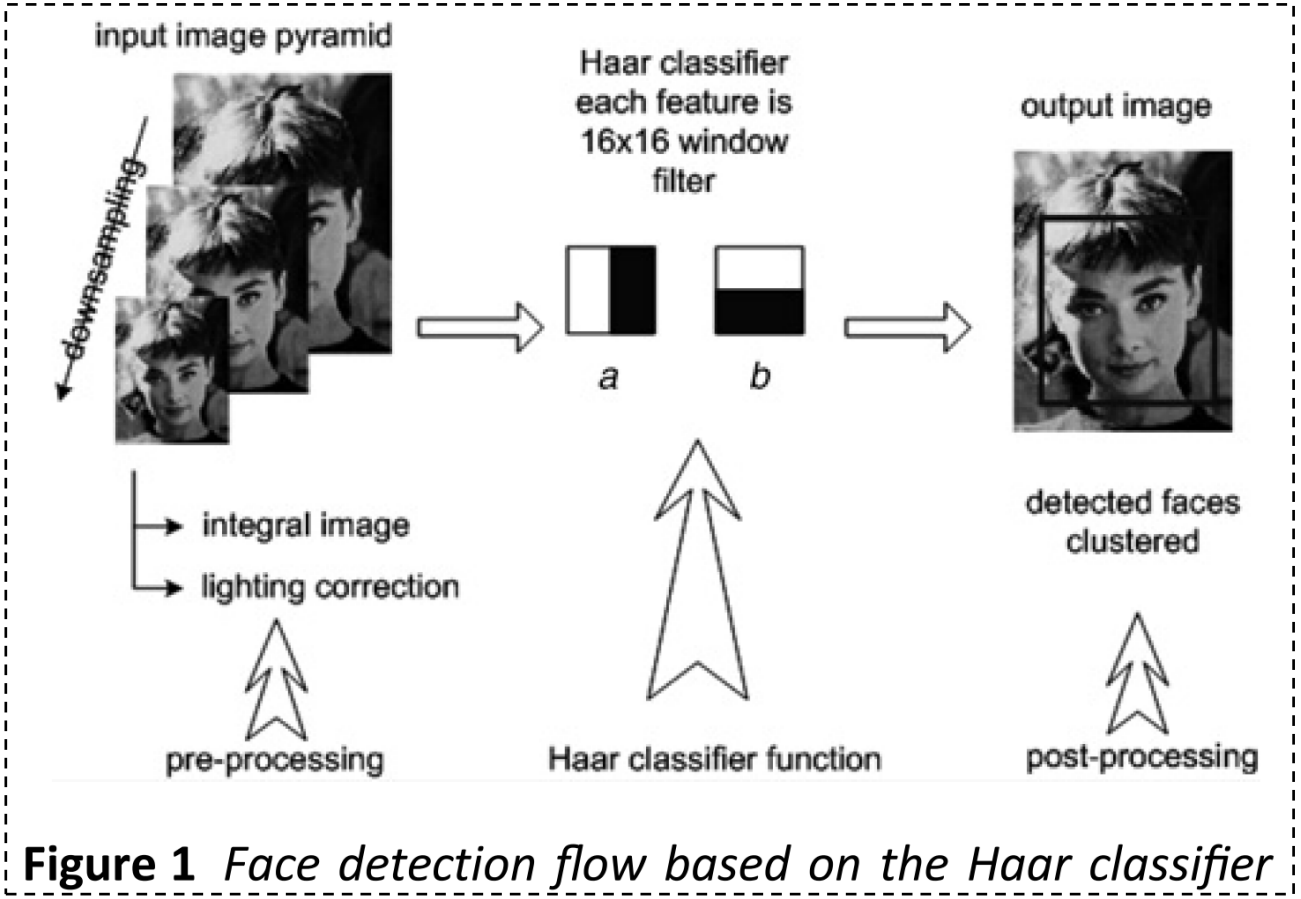


Figure 1 *Face detection flow based on the Haar classifier*

Paper: Field programmable gate array-based Haar classifier for accelerating face detection algorithm

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

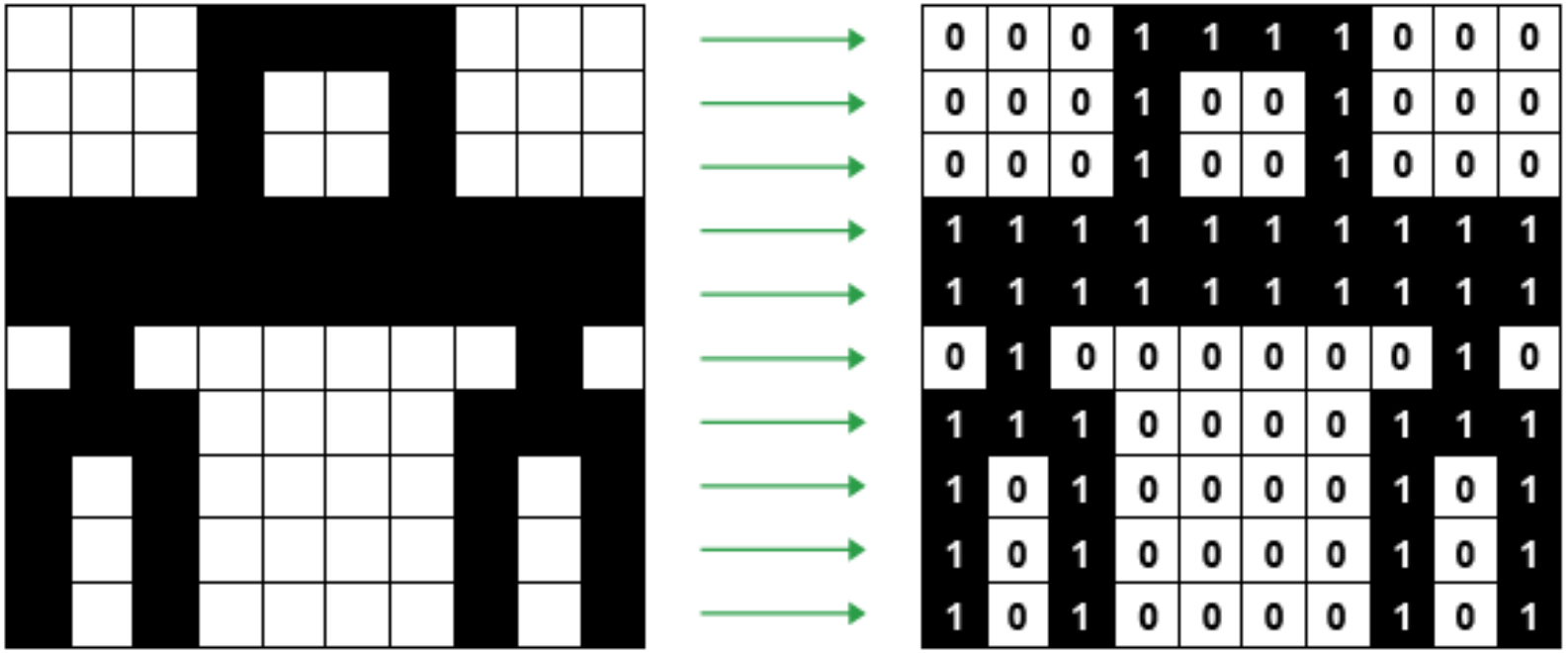
Cascade Classification

- Haar Feature-based Cascade Classifier for Object Detection
  - The object detector described below has been initially proposed by Paul Viola [Viola01] and improved by Rainer Lienhart [Lienhart02].
  - A classifier is trained with a few hundred sample views of a particular object (i.e., a face or a car), called positive examples
    - Output 1: the region is likely to show the object (i.e., face/car)
    - Output 0: otherwise

- [Viola01] Paul Viola and Michael J. Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. IEEE CVPR, 2001. https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/viola-cvpr-01.pdf
- [Lienhart02] Rainer Lienhart and Jochen Maydt. An Extended Set of Haar-like Features for Rapid Object Detection. IEEE ICIP, Vol. 1, pp. 900-903, Sep. 2002. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.214.9150&rep=rep1&type=pdf

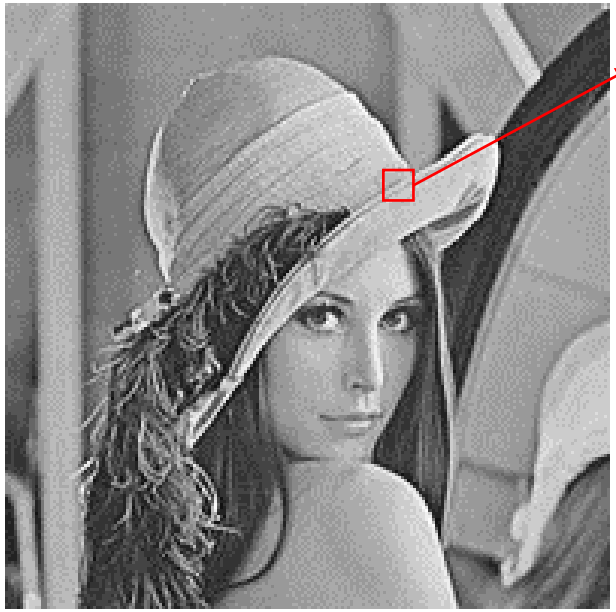
Bitmap images

- Example: black-and-white image



Bitmap images

- Example: grayscale picture
  - 8 bits per pixel
  - This pixel depth allows 256 different intensities



154	108	198	216	52
61	168	148	52	45
72	80	55	134	39
89	129	232	204	155
156	99	118	125	83

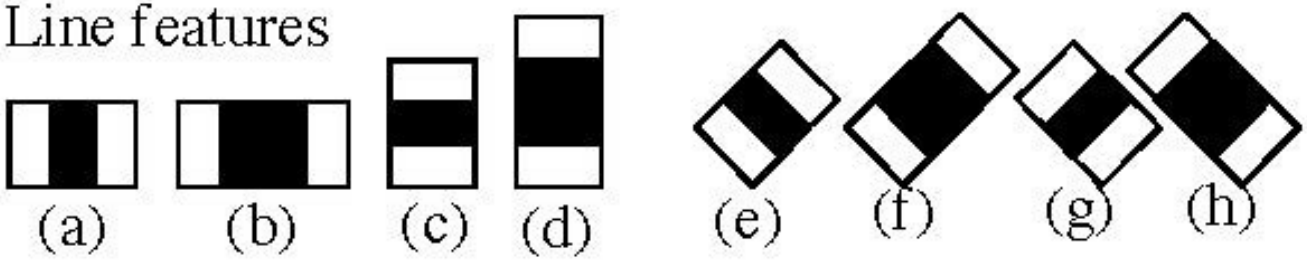
Camera sees this

Fig source: https://zh.wikipedia.org/wiki/%E8%90%8A%E5%A8%9C%E5%9C%96

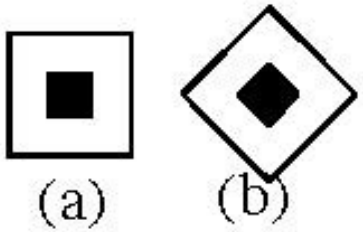
Haar-Like Features

1. Edge features

2. Line features



3. Center-surround features



Find features

- Pick a scale (ex: 24x24 pixels) for the feature
- Slide it across the image
- Compute the average pixel values under the white area and the black area
- If the difference between the areas is above some threshold, the feature matches

Visualizing Cascade Classifiers by OpenCV
https://www.youtube.com/watch?v=vTBnlsz4g-U

Find features

1. Calculate the average of white/black pixel
2. Calculate the difference

0	0	1	1
0	0	1	1
0	0	1	1
0	0	1	1

Image

0	0	1	1
0	0	1	1
0	0	1	1
0	0	1	1

Edge feature

$$\Delta = black - whilte = 1$$

0.1	0.2	0.6	0.8
0.2	0.3	0.8	0.6
0.2	0.1	0.6	0.8
0.2	0.1	0.8	0.9

Image

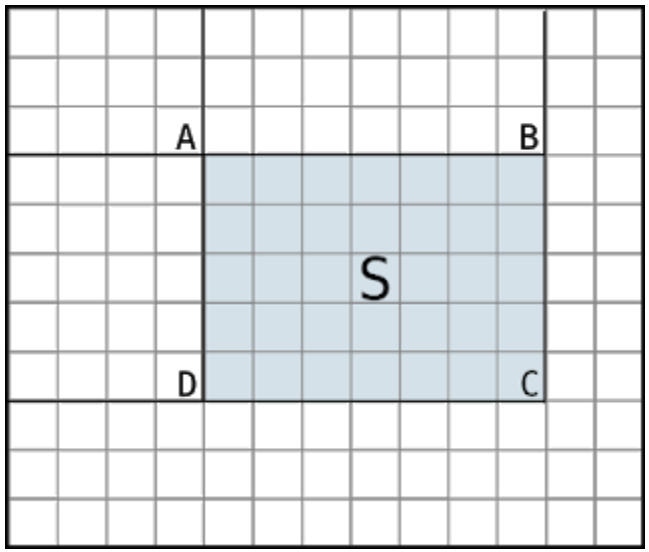
0	0	1	1
0	0	1	1
0	0	1	1
0	0	1	1

Edge feature

$$\Delta = \frac{0.6 + 0.8 + \cdots}{8} - \frac{0.1 + 0.2 + \cdots}{8}$$
$$= 0.7375 - 0.175 = 0.56$$

Integral Image (1/3)

- a quick and effective way of calculating the sum of values (pixel values) of a rectangular subset of a grid
- It can also used for calculating the average intensity within a given image.



Sum = Value(C) - Value(B) - Value(D) + Value(A)

Integral Image (2/3)

I(x, y): 原始影像的像素值

0.1	0.2	0.6	0.8
0.2	0.3	0.8	0.6
0.2	0.1	0.6	0.8
0.2	0.1	0.8	0.9

Original image

S(x, y): 積分圖的值

0.1	0.3	0.9	1.7
0.3	0.8	2.2	3.6
0.5	1.1	3.1	5.3
0.7	1.4	4.2	7.3

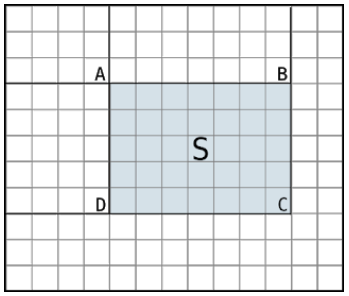
integral image

$$S(x, y) = I(x, y) + S(x-1, y) + S(x, y-1) - S(x-1, y-1)$$

Integral Image (3/3)

0.1	0.2	0.6	1.7
0.2	0.3	0.8	3.6
0.2	0.1	0.6	5.3
0.2	0.1	0.8	7.3

0.1 A	0.3	**0.9** B	1.7
0.3	0.8	2.2	3.6
0.5 D	1.1	**3.1** C	5.3
0.7	1.4	4.2	7.3



Calculate the are summation

0.1	0.3	0.9	1.7
0.3	0.8	2.2	3.6
0.5	1.1	3.1 C	5.3
0.7	1.4	4.2	7.3

0.1	0.3	0.9 B	1.7
0.3	0.8	2.2	3.6
0.5	1.1	3.1	5.3
0.7	1.4	4.2	7.3

0.1 A	0.3	0.9	1.7
0.3	0.8	2.2	3.6
0.5	1.1	3.1	5.3
0.7	1.4	4.2	7.3

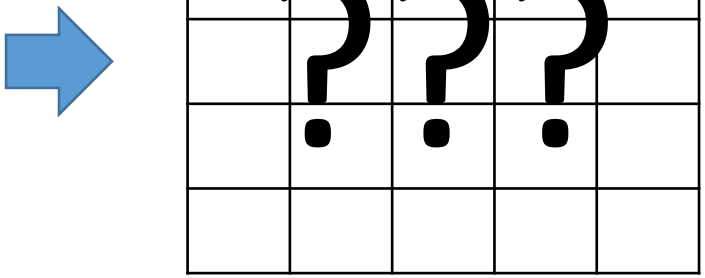
0.1	0.3	0.9	1.7
0.3	0.8	2.2	3.6
0.5 D	1.1	3.1	5.3
0.7	1.4	4.2	7.3

Sum = Value(C) - Value(B) + Value(A) - Value(D)

Discussion1

- How to calculate the area summation by integral image?
  - 1. Write down the value of integral image
  - 2. Sum = Value(C) - Value(B) + Value(A) - Value(D) = ?????

0.1	0.2	0.3	0.5	0.8
0.2	0.2	0.3	0.7	0.9
0.1	0.2	0.4	0.7	0.9
0.3	0.1	0.2	0.8	0.2
0.1	0.2	0.4	0.6	0.1



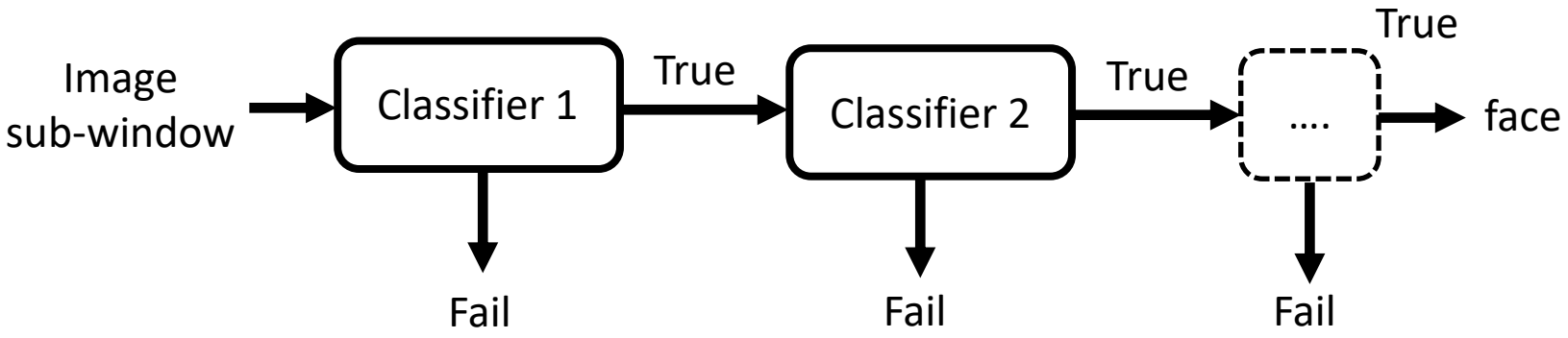
AdaBoost

- Adaptive Boosting
  - Try out multiple weak classifiers over several rounds
  - Select the best weak classifier in each round and combining the best weak classifiers to create a strong classifier

Data point	Classifier 1	Classifier 2	Classifier 3	…
P_1	Pass	Fail	Fail	…
P_2	Pass	Pass	Pass	…
P_3	Fail	Pass	Pass	…
…	…	…	…	…

Cascades

- Haar cascades consists of a series of weak classifiers
  - barely better than 50% correct
  - If an area passes a single classifier, go to the next classifier; otherwise, area doesn't match



Recall the code

```
# Detect faces in the image
faces = faceCascade.detectMultiScale(
    gray,
    scaleFactor=1.1,
    minNeighbors=5,
    minSize=(30, 30)
)
```

- **scaleFactor** – Parameter specifying how much the image size is reduced at each image scale.
- **minNeighbors** – Parameter specifying how many neighbors each candidate rectangle should have to retain it.
- **minSize** – Minimum possible object size. Objects smaller than that are ignored.

https://docs.opencv.org/3.4/d1/de5/classcv_1_1CascadeClassifier.html

Related parameters

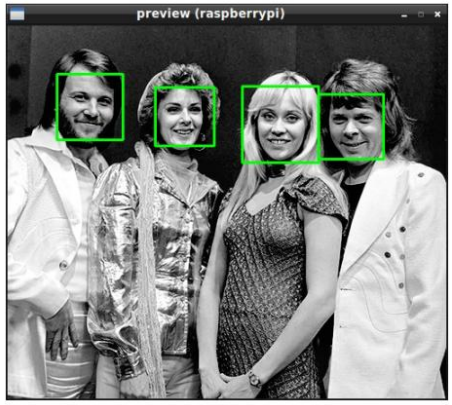
CascadeClassifier::detectMultiScale

- **Image**: Matrix of the type CV_8U containing an image where objects are detected.
- **Objects**: Vector of rectangles where each rectangle contains the detected object, the rectangles may be partially outside the original image.
- **scaleFactor:** Parameter specifying how much the image size is reduced at each image scale.
- **minNeighbors:** Parameter specifying how many neighbors each candidate rectangle should have to retain it.
- **flags**: Parameter with the same meaning for an old cascade as in the function cvHaarDetectObjects. It is not used for a new cascade.
- **minSize:** Minimum possible object size. Objects smaller than that are ignored.
- **maxSize**: Maximum possible object size. Objects larger than that are ignored. If maxSize == minSize model is evaluated on single scale.

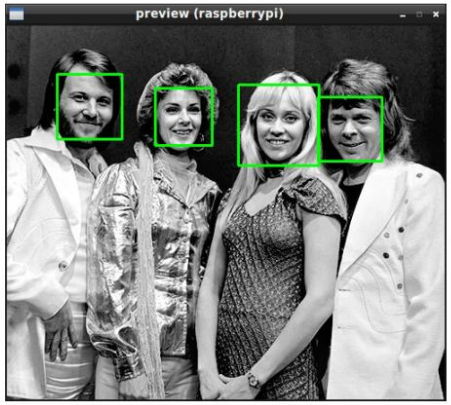
Try to use different **parameters**, you will get different results.

scaleFactor
minNeighbors=5, minSize=(30, 30)

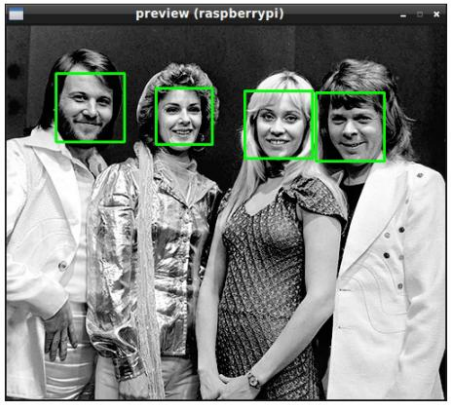
scaleFactor: Parameter specifying how much the image size is reduced at each image scale.



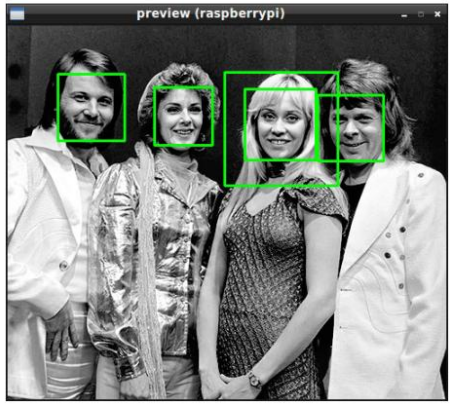
scaleFactor=1.1



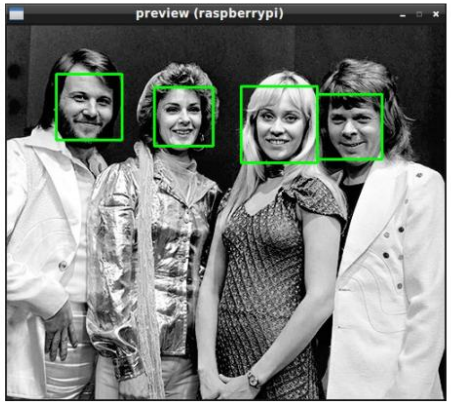
1.2



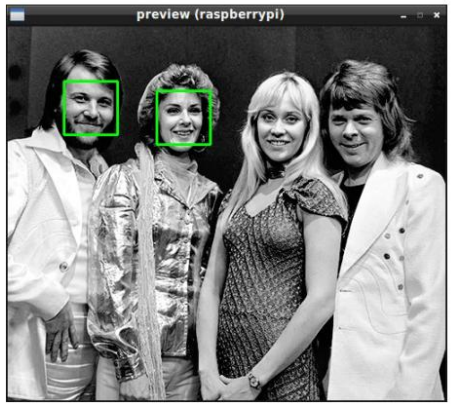
1.3



1.4



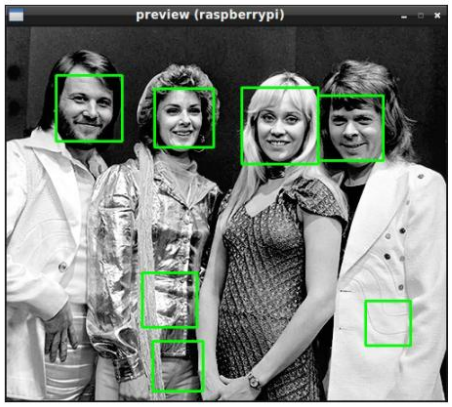
1.5



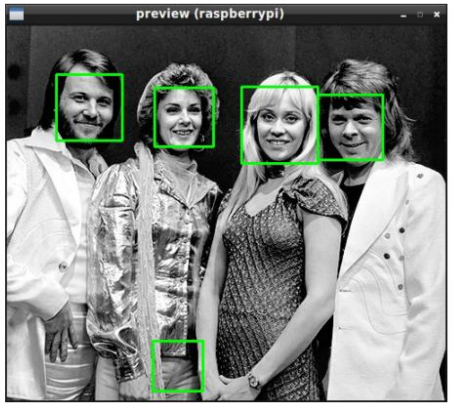
1.6

minNeighbors
scaleFactor=1.1, minSize=(30, 30)

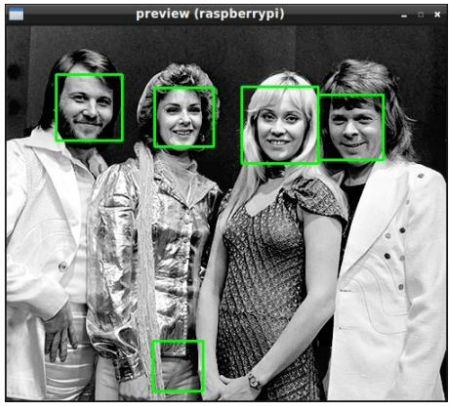
minNeighbors: Parameter specifying how many neighbors each candidate rectangle should have to retain it



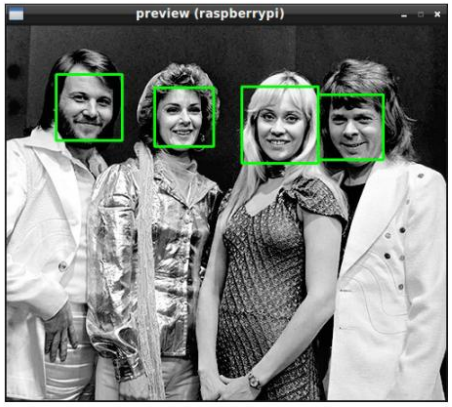
minNeighbors=1



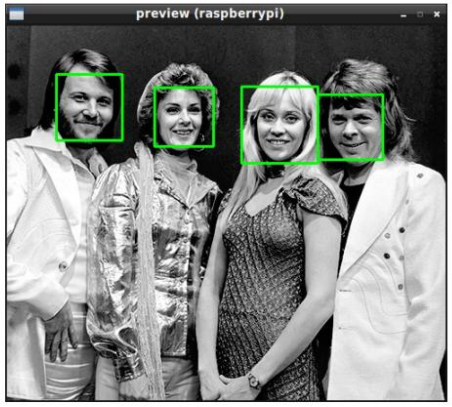
2



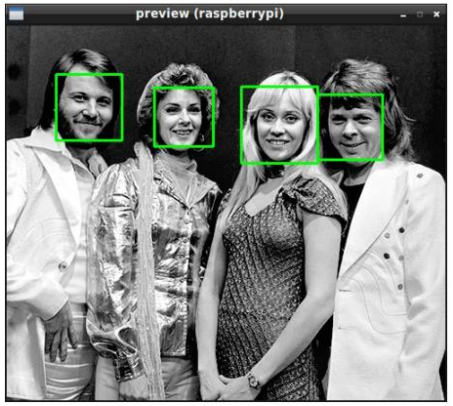
3



5



10

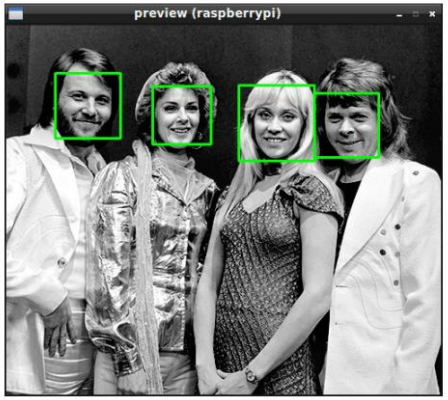


20

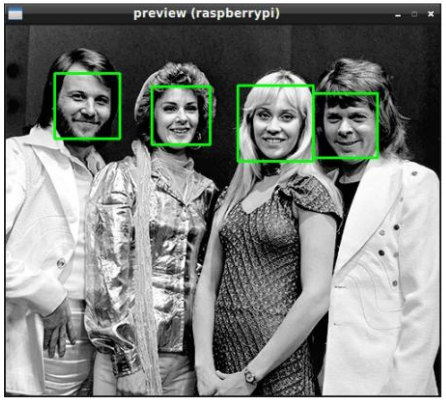
By 台灣樹莓派

minSize(x, y)
scaleFactor=1.1, minNeighbosr=5

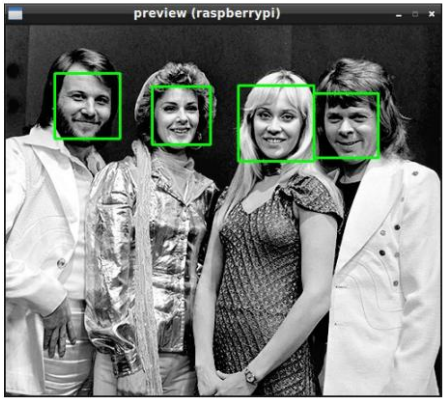
minSize: Minimum possible object size. Objects smaller than that are ignored



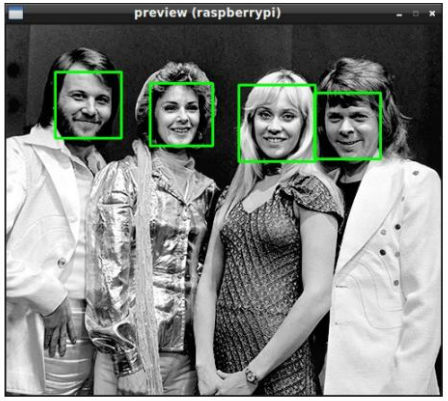
minSize=(15, 15)



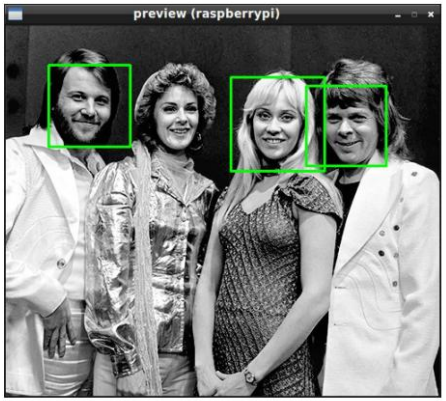
(30, 30)



(60, 60)



(90, 90)



(120 ,120)

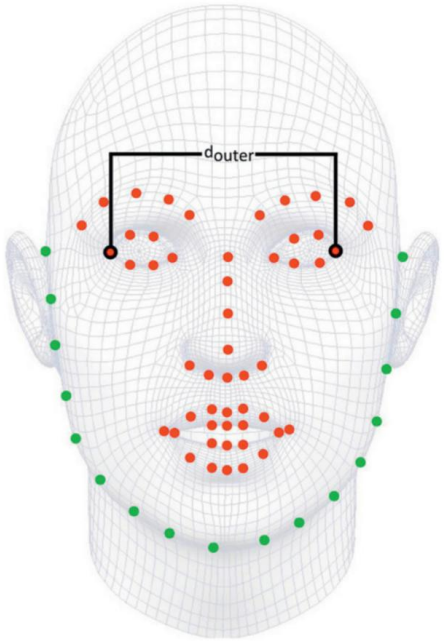
(150, 150)

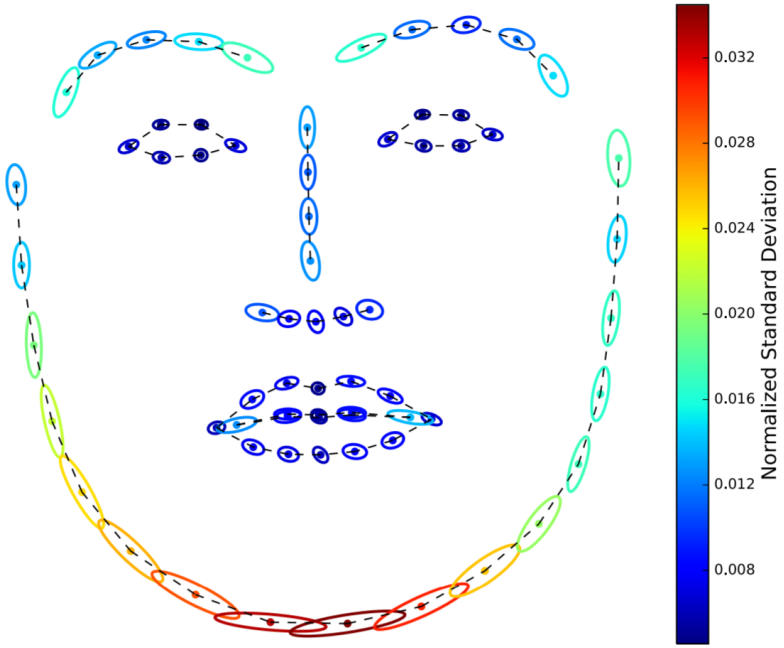
By 台灣樹莓派

https://ibug.doc.ic.ac.uk/resources/facial-point-annotations/

B. Facial landmark

- 300 Faces In-The-Wild Challenge: database and results
  https://ibug.doc.ic.ac.uk/media/uploads/documents/sagonas_2016_imavis.pdf



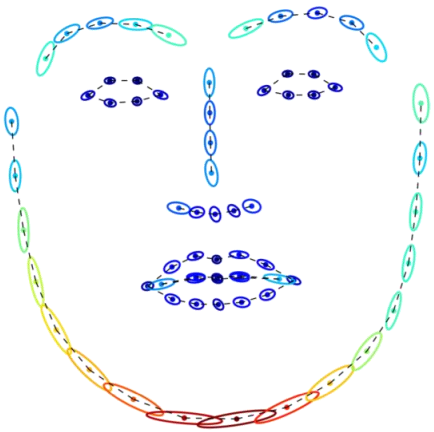


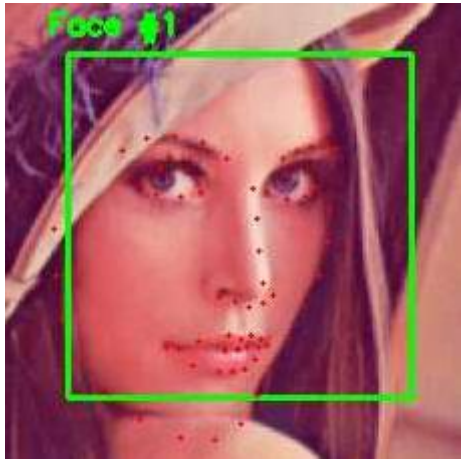
Two steps for capturing facial landmark:

A. Face detection (OpenCV or dlib)
B. Draw landmark (dlib)

B. Facial landmark

- Two sample code with Haar cascade and dlib
  - dlib: machine learning algorithms and tools
  - Analyze image
    - 2.1image_facial_landmarks.py
    - 2.2image_facial_landmarks.py





Error message

- ImportError: No module named dlib?
  - Sol: pip install dlib

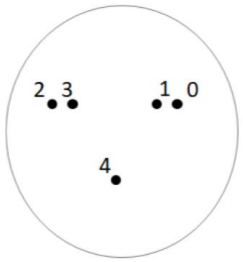
- ImportError: libopenblas.so.0: cannot open shared object file: No such file or directory
  - Sol: sudo apt install libopenblas-dev

```
pi@raspberrypi:~/lab5_camera $ python 2.1image_facial_landmarks.py
Traceback (most recent call last):
  File "/home/pi/lab5_camera/2.1image_facial_landmarks.py", line 4, in <module>
    import dlib
  File "/home/pi/.local/lib/python3.9/site-packages/dlib/__init__.py", line 19, in <module>
    from _dlib_pybind11 import *
ImportError: libopenblas.so.0: cannot open shared object file: No such file or directory
```

Face detection (OpenCV)

2.1image_facial_landmarks.py

```
# load OpenCV's Haar cascade for face detection,
# (This is faster than dlib's built-in HOG detector, but less accurate.)
detector_file = "model/haarcascade_frontalface_default.xml"
detector = cv2.CascadeClassifier(detector_file)

# create the facial landmark                    (shape_predictor_5_face_landmarks.dat)
predictor.predictor_file = "model/shape_predictor_68_face_landmarks.dat"
predictor = dlib.shape_predictor(predictor_file)

# detect faces in the grayscale frame by opencv's method: (x, y, w, h)
rects = detector.detectMultiScale(gray, scaleFactor=1.1,
    minNeighbors=5, minSize=(30, 30),
    flags=cv2.CASCADE_SCALE_IMAGE)
```



5 points

68 points

http://dlib.net/face_detection_ex.cpp.html
http://dlib.net/python/index.html#dlib.get_frontal_face_detector
https://docs.opencv.org/2.4/modules/imgproc/doc/miscellaneous_transformations.html

Face detection (dlib)

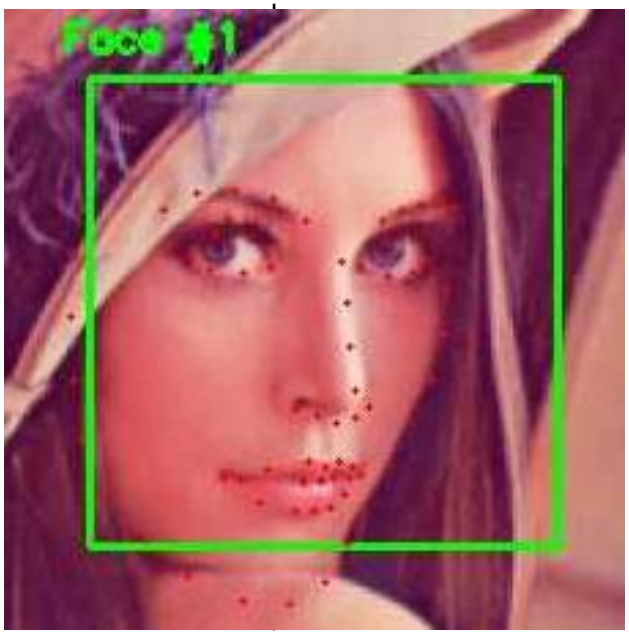
2.2image_facial_landmarks.py

```
# initialize dlib's face detector (HOG-based)
# then create the facial landmark predictor
detector = dlib.get_frontal_face_detector()

# load the input image, resize it, and convert it to grayscale
image = cv2.imread("img.jpg”)
image = imutils.resize(image, width=500)

# cvtColor: Converts an image from one color space to another.
# Here, convert a RGB image to gray
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

# detect faces in the grayscale image by dlib’s method (HOG)
# The 1 in the second argument indicates that we should upsample the image 1 time.
# This will make everything bigger and allow us to detect more faces.
rects = detector(gray, 1)
```

https://github.com/jrosebr1/imutils/blob/master/imutils/face_utils/helpers.py
http://dlib.net/face_detector.py.html

Draw landmark

```
# loop over the face detections
face_counter = 0
for (x, y, w, h) in rects:
    # construct a dlib rectangle object from the Haar cascade bounding box
    rect = dlib.rectangle(int(x), int(y), int(x + w), int(y + h))

    # determine the facial landmarks for the face region,
    # then convert the facial landmark (x, y)-coordinates to a NumPy array
    shape = predictor(gray, rect)
    shape = face_utils.shape_to_np(shape)

    # convert dlib's rectangle to a OpenCV-style bounding box
    # [i.e., (x, y, w, h)], then draw the face bounding box
    (x, y, w, h) = face_utils.rect_to_bb(rect)
    cv2.rectangle(image, (x, y), (x + w, y + h), (0, 255, 0), 2)

    # show the face number
    cv2.putText(image, "Face #{}".format(face_counter + 1), (x - 10, y - 10),
        cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)

    # loop over the (x, y)-coordinates for the facial landmarks and draw them on the image
    for (x, y) in shape:
        cv2.circle(image, (x, y), 1, (0, 0, 255), -1)

    face_counter = face_counter + 1
```



https://www.sciencedirect.com/science/article/pii/S0167865511000122

Face recognition using Histograms of Oriented Gradients

Histogram of Oriented Gradients: HOG, 方向梯度直方圖。
用於計算畫面中每個像素點的灰度變化，形成圖片的梯度。

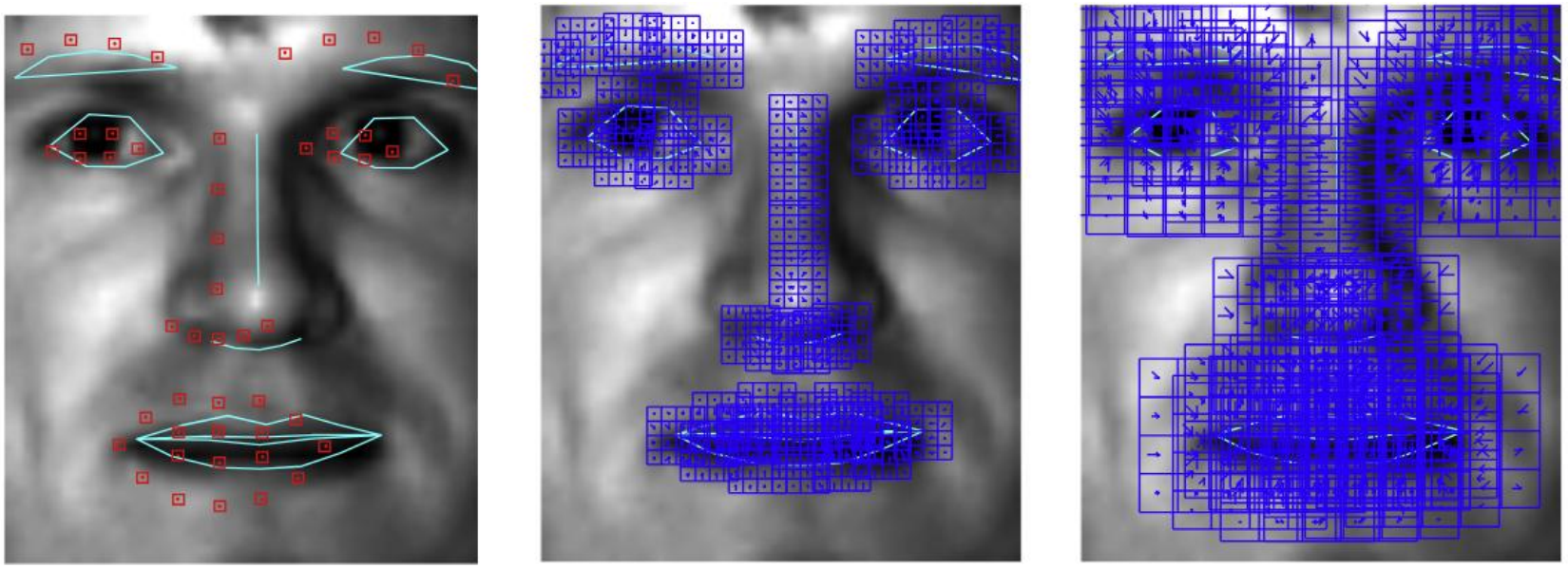
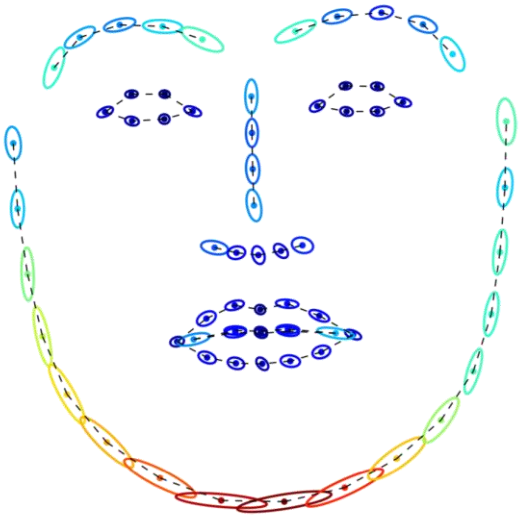
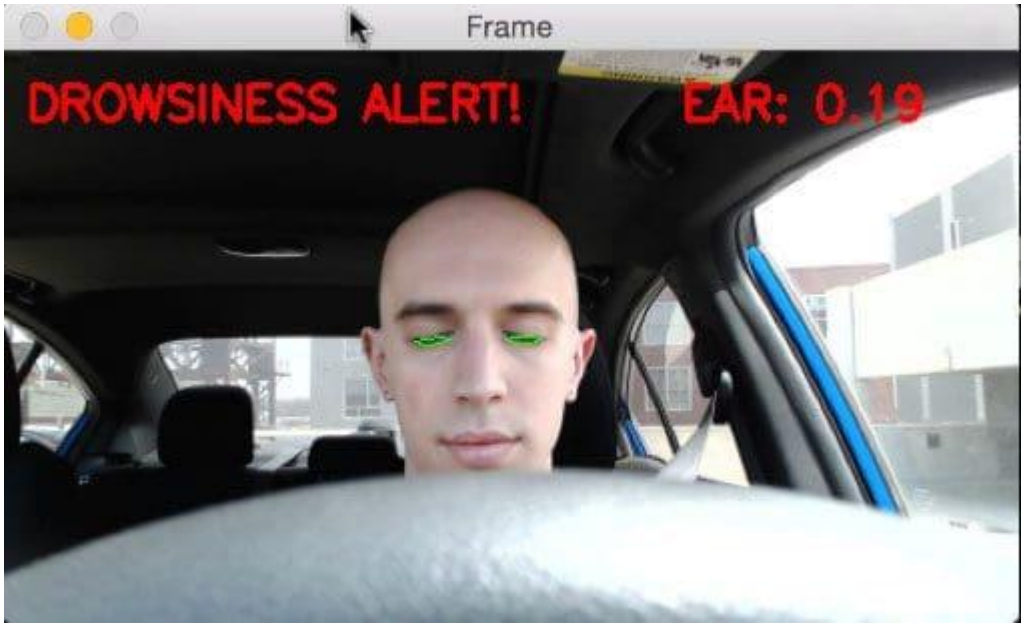


Fig. 2. Left: Initialized landmarks (red boxes) and result for the AAM fitting in an image from the Yale database. Center: extracted HOG descriptors, patch size = 24 24. Right: extracted HOG descriptors, patch size = 64 64.

Quiz 1

- Design drowsiness detection based on facial landmark
  - Hint: 提取眼睛四周的資訊, 觀察是否有閉上眼睛

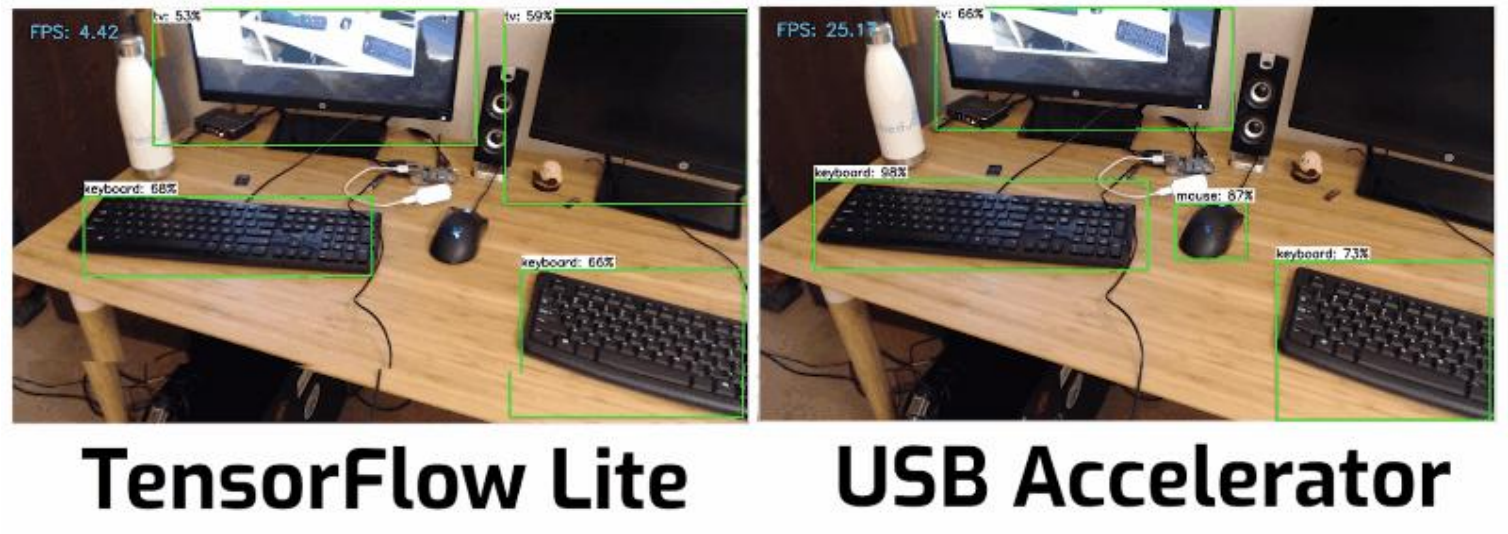




Part3: 物件辨識 (tensorflow, YOLO)

Object detection

- Google provides a sample quantized **SSDLite-MobileNet-v2** object detection model which is trained off the **MSCOCO dataset** and converted to run on **TensorFlow Lite**. It can detect and identify 80 different common objects, such as people, cars, cups, etc.



https://github.com/EdjeElectronics/TensorFlow-Lite-Object-Detection-on-Android-and-Raspberry-Pi/blob/master/deploy_guides/Raspberry_Pi_Guide.md

https://ai.googleblog.com/2018/04/mobilenetv2-next-generation-of-on.html
https://arxiv.org/abs/1801.04381v4

SSDLite-MobileNet-v2

- MobileNetV2 is a very effective feature extractor for object detection and segmentation. For example, for detection when paired with the newly introduced SSDLite [2] the new model is about 35% faster with the same accuracy than MobileNetV1.

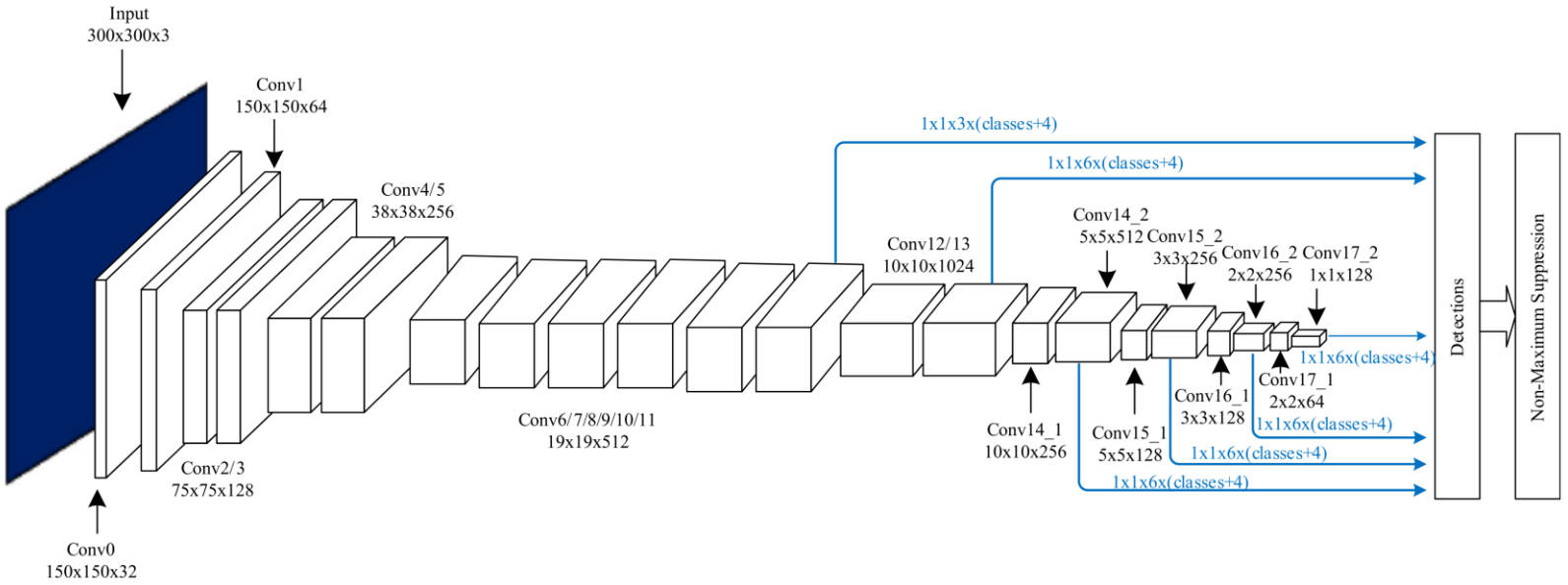
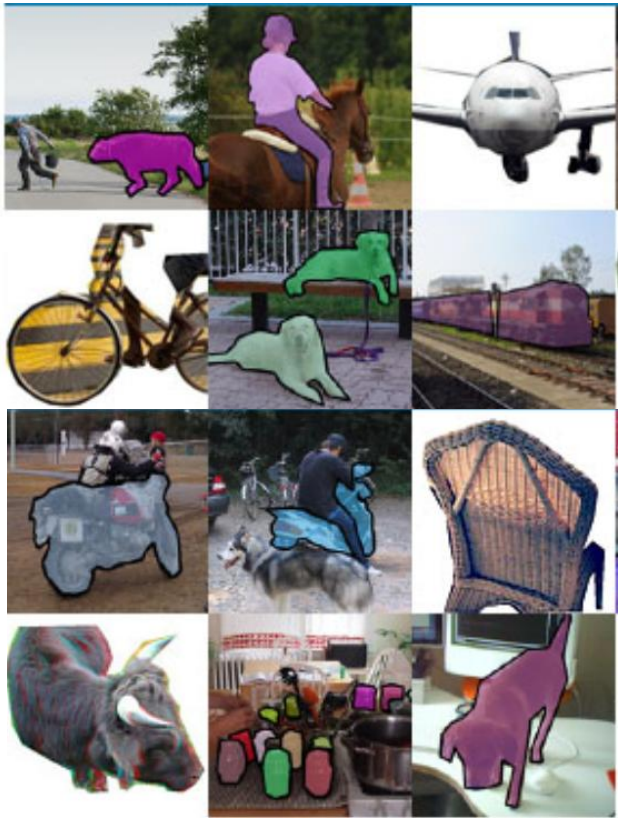


Fig source: https://link.springer.com/article/10.1007/s11042-022-13058-w

MSCOCO dataset

- COCO is a large-scale object detection, segmentation, and captioning dataset.
- COCO has several features:
  - Object segmentation
  - Recognition in context
  - Superpixel stuff segmentation
  - 330K images (>200K labeled)
  - 1.5 million object instances
  - 80 object categories
  - 91 stuff categories
  - 5 captions per image
  - 250,000 people with keypoints

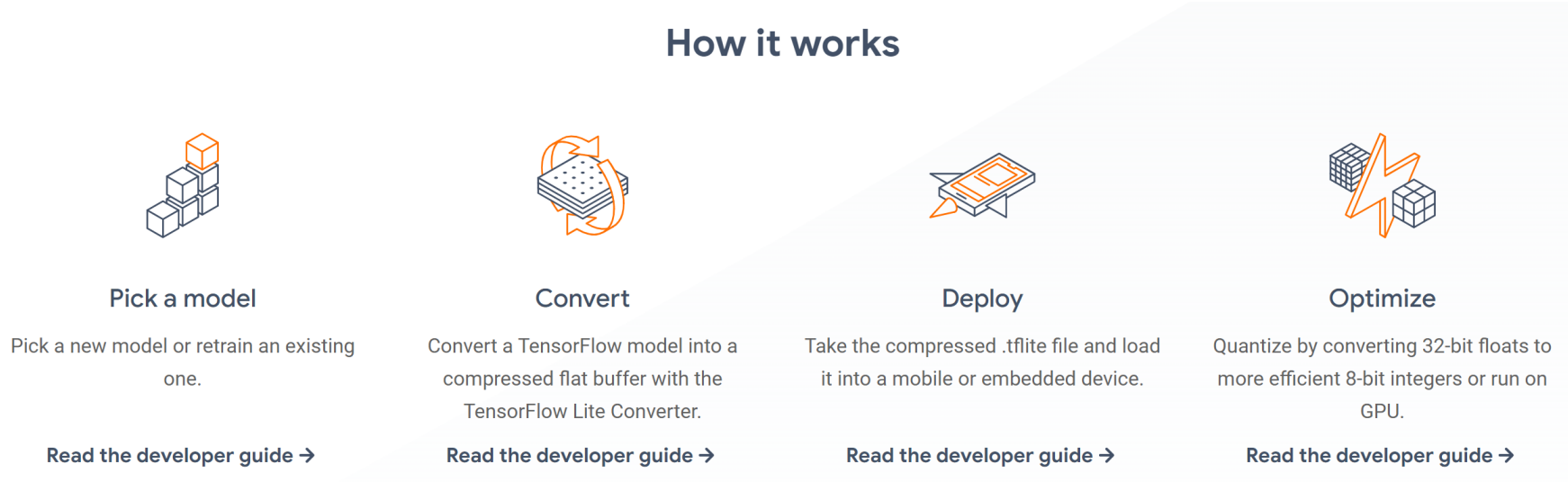


Dataset example

https://www.tensorflow.org/lite

TensorFlow Lite

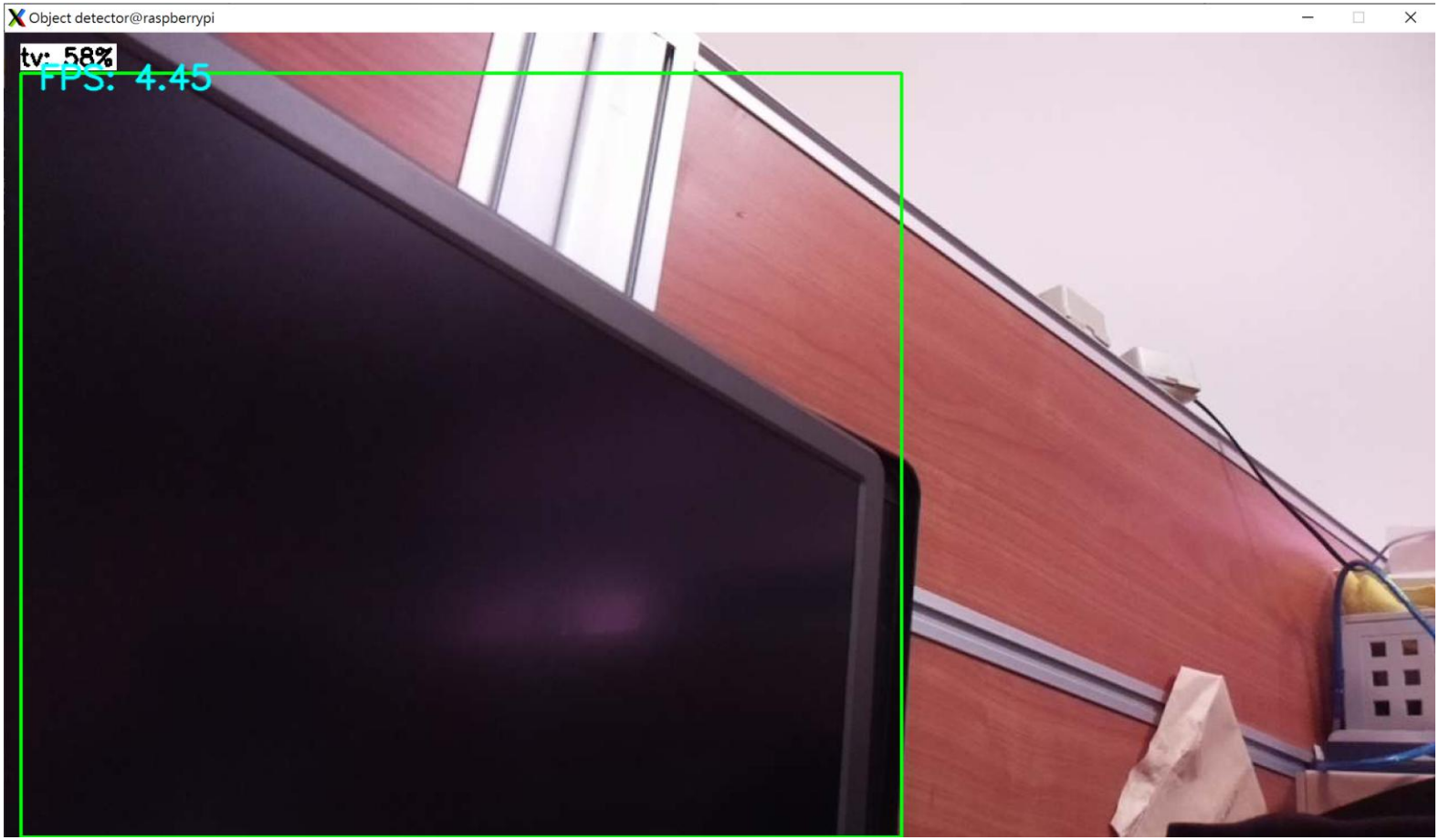
- TensorFlow Lite is a mobile library for deploying models on mobile, microcontrollers and other edge devices.



Build TensorFlow Lite Object Detection

- git clone https://github.com/EdjeElectronics/TensorFlow-Lite-Object-Detection-on-Android-and-Raspberry-Pi.git
- mv TensorFlow-Lite-Object-Detection-on-Android-and-Raspberry-Pi tflite
- cd tflite/
- bash get_pi_requirements.sh
- wget https://storage.googleapis.com/download.tensorflow.org/models/tflite/coco_ssd_mobilenet_v1_1.0_quant_2018_06_29.zip
- unzip coco_ssd_mobilenet_v1_1.0_quant_2018_06_29.zip -d Sample_TFLite_model
- pip install numpy==1.23.5
- python TFLite_detection_webcam.py --modeldir=Sample_TFLite_model

Result



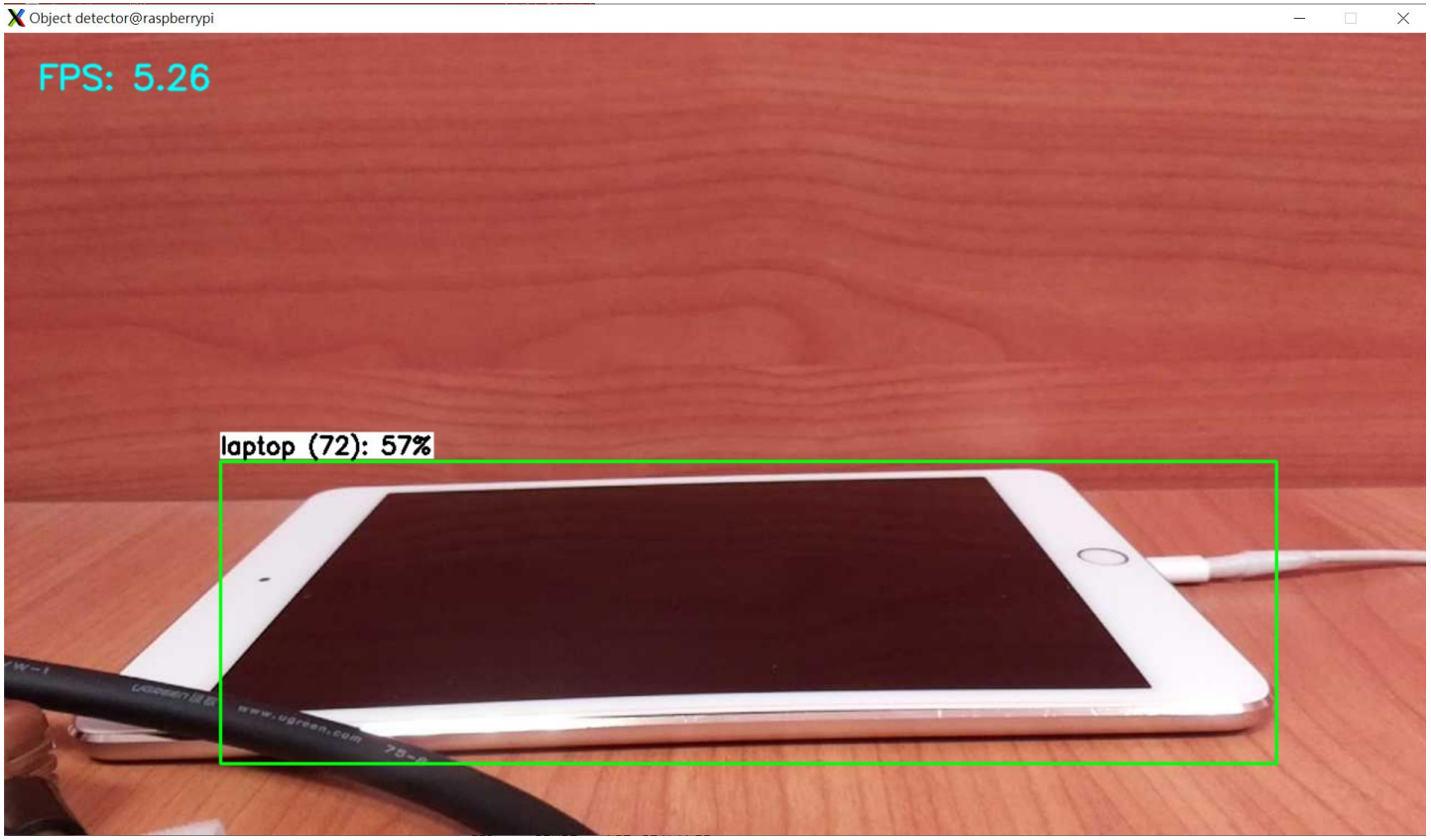
Lablemap info.

1	???	sheep	baseball glove	donut	toaster
2	person	cow	skateboard	cake	sink
3	bicycle	elephant	surfboard	chair	refrigerator
4	car	bear	tennis racket	couch	???
5	motorcycle	zebra	bottle	potted plant	book
6	airplane	giraffe	???	bed	clock
7	bus	???	wine glass	???	vase
8	train	backpack	cup	dining table	scissors
9	truck	umbrella	fork	???	teddy bear
10	boat	???	knife	???	hair drier
11	traffic light	???	spoon	toilet	toothbrush
12	fire hydrant	handbag	bowl	???	
13	???	tie	banana	tv	
14	stop sign	suitcase	apple	laptop	
15	parking meter	frisbee	sandwich	mouse	
16	bench	skis	orange	remote	
17	bird	snowboard	broccoli	keyboard	
18	cat	sports ball	carrot	cell phone	
19	dog	kite	hot dog	microwave	
20	horse	baseball bat	pizza	oven	

Under Sample_TFLite_model, you can find labelmap.txt

Discussion 2

- How to modify the sample code to show class id?



YOLO: Real-Time Object Detection

- YOLO (You only look once)

- Prior detection systems repurpose classifiers or localizers to perform detection. They apply the model to an image at multiple locations and scales. High scoring regions of the image are considered detections.

- They use a totally different approach. They apply a single neural network to the full image. This network divides the image into regions and predicts bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicted probabilities.

Run YOLO with ultralytics

- Ultralytics YOLOv8 is a cutting-edge, state-of-the-art (SOTA) model that builds upon the success of previous YOLO versions and introduces new features and improvements to further boost performance and flexibility. YOLOv8 is designed to be fast, accurate, and easy to use, making it an excellent choice for a wide range of object detection and tracking, instance segmentation, image classification and pose estimation tasks.

Steps

- wget https://github.com/KumaTea/pytorch-arm/releases/download/v1.8.1/torch-1.8.1-cp39-cp39-linux_armv7l.whl
- wget https://github.com/KumaTea/pytorch-arm/releases/download/v1.8.1/torchvision-0.9.1-cp39-cp39-linux_armv7l.whl
- pip install torch-1.8.1-cp39-cp39-linux_armv7l.whl
- pip install torchvision-0.9.1-cp39-cp39-linux_armv7l.whl
- pip install ultralytics

遇到錯誤訊息?

- 網路不穩的時候, 安裝套件會遇到錯誤訊息

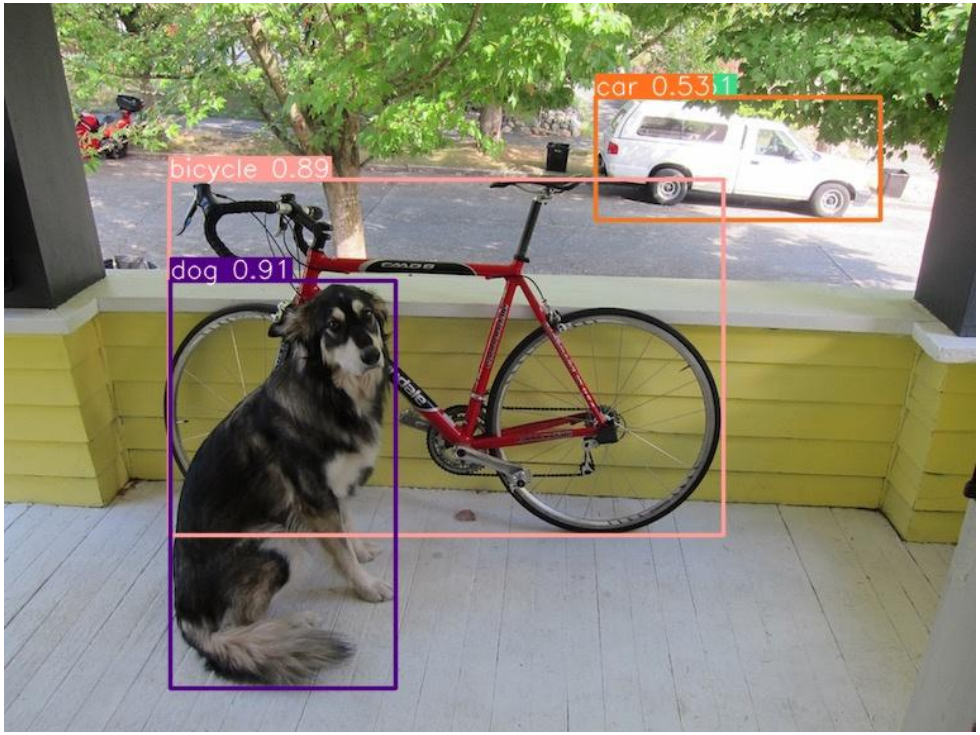
```
WARNING: Retrying (Retry(total=4, connect=None, read=None, redirect=None, status=None)) after connection broken by 'ProtocolError('Connec
tion aborted.', RemoteDisconnected('Remote end closed connection without response'))': /simple/pandas/
Collecting pandas>=1.1.4
  Downloading https://www.piwheels.org/simple/pandas/pandas-2.2.3-cp39-cp39-linux_armv7l.whl (43.4 MB)
     |                                | 2.9 MB 42 kB/s eta 0:15:50
ERROR: THESE PACKAGES DO NOT MATCH THE HASHES FROM THE REQUIREMENTS FILE. If you have updated the package versions, please update the has
hes. Otherwise, examine the package contents carefully; someone may have tampered with them.
    pandas>=1.1.4 from https://www.piwheels.org/simple/pandas/pandas-2.2.3-cp39-cp39-linux_armv7l.whl#sha256=ab70a99105c9ba224ee41fea63c9
73f979494a0560c0eaa69d879fbeac00ae3b (from ultralytics):
        Expected sha256 ab70a99105c9ba224ee41fea63c973f979494a0560c0eaa69d879fbeac00ae3b
             Got        26022b180d6eb9e437b93d83aef604acdbcdd40f56d06c33b2de5dbc3587fcb5

pi@raspberrypi:~ $ wget https://www.piwheels.org/simple/pandas/pandas-2.2.3-cp39-cp39-linux_armv7l.whl
```

- 解決方法
  - 下載該套件, 再pip install
- Ex:
  - wget https://www.piwheels.org/simple/pandas/pandas-2.2.3-cp39-cp39-linux_armv7l.whl
  - pip install pandas-2.2.3-cp39-cp39-linux_armv7l.whl

Sample code

```
from ultralytics import YOLO
import cv2
import time

model = YOLO('yolov8n.pt')
img = cv2.imread('dog.jpg')

# First run to 'warm-up' the model
model.predict(source=img, save=False, save_txt=False, conf=0.5, verbose=False)

# Second run
t_start = time.monotonic()
results = model.predict(source=img, save=False, save_txt=False, conf=0.5, verbose=False)
dt = time.monotonic() - t_start
print("Predict Time:", dt)

# Show results
boxes = results[0].boxes
names = model.names
confidences, class_ids = boxes.conf, boxes.cls.int()
rects = boxes.xyxy.int()
for ind in range(boxes.shape[0]):
    print("Rect:", names[class_ids[ind].item()], confidences[ind].item(), rects[ind].tolist())
```

Results




```
pi@raspberrypi:~ $ time python yolov8.py
Predict Time: 8.565574948996073
Rect: dog 0.9080022573471069 [131, 219, 309, 541]
Rect: bicycle 0.8887635469436646 [130, 139, 568, 420]
Rect: car 0.5305861234664917 [467, 74, 692, 171]
Rect: truck 0.5086143016815186 [467, 74, 692, 171]

real    0m23.388s
user    0m23.747s
sys     0m1.993s
```

分析一張圖片 需要9秒...

Inference Arguments

- model.predict() accepts multiple arguments that can be passed at inference time to override defaults:

Inference arguments:

Argument	Type	Default	Description
`source`	`str`	`'ultralytics/assets'`	Specifies the data source for inference. Can be an image path, video file, directory, URL, or device ID for live feeds. Supports a wide range of formats and sources, enabling flexible application across different types of input.
`conf`	`float`	`0.25`	Sets the minimum confidence threshold for detections. Objects detected with confidence below this threshold will be disregarded. Adjusting this value can help reduce false positives.
`iou`	`float`	`0.7`	Intersection Over Union (IoU) threshold for Non-Maximum Suppression (NMS). Lower values result in fewer detections by eliminating overlapping boxes, useful for reducing duplicates.
`imgsz`	`int or tuple`	`640`	Defines the image size for inference. Can be a single integer `640` for square resizing or a (height, width) tuple. Proper sizing can improve detection accuracy and processing speed.
`half`	`bool`	`False`	Enables half-precision (FP16) inference, which can speed up model inference on supported GPUs with minimal impact on accuracy.
`device`	`str`	`None`	Specifies the device for inference (e.g., `cpu`, `cuda:0` or `0`). Allows users to select between CPU, a specific GPU, or other compute devices for model execution.
`max_det`	`int`	`300`	Maximum number of detections allowed per image. Limits the total number of objects the model can detect in a single inference, preventing excessive outputs in dense scenes.

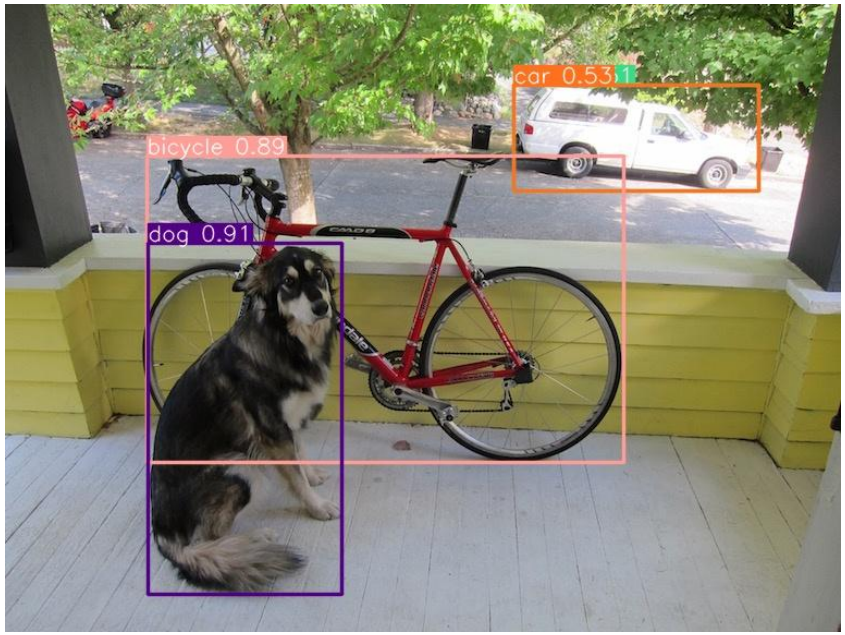
Visualization arguments:

Argument	Type	Default	Description
`show`	`bool`	`False`	If `True`, displays the annotated images or videos in a window. Useful for immediate visual feedback during development or testing.
`save`	`bool`	`False`	Enables saving of the annotated images or videos to file. Useful for documentation, further analysis, or sharing results.
`save_frames`	`bool`	`False`	When processing videos, saves individual frames as images. Useful for extracting specific frames or for detailed frame-by-frame analysis.
`save_txt`	`bool`	`False`	Saves detection results in a text file, following the format `[class] [x_center] [y_center] [width] [height] [confidence]`. Useful for integration with other analysis tools.
`save_conf`	`bool`	`False`	Includes confidence scores in the saved text files. Enhances the detail available for post-processing and analysis.
`save_crop`	`bool`	`False`	Saves cropped images of detections. Useful for dataset augmentation, analysis, or creating focused datasets for specific objects.
`show_labels`	`bool`	`True`	Displays labels for each detection in the visual output. Provides immediate understanding of detected objects.

https://docs.ultralytics.com/modes/predict/#inference-arguments

Quiz 2

- Try to use YOLO analyze your image, how to set the parameter to save annotated image?
- Please show your annotated image



Check with venv

- (以下是在虛擬環境測試用的指令)
- python -m venv yolo-test
- source yolo-test/bin/activate # 進入虛擬環境
- pip install --upgrade pip setuptools wheel
- pip install torch-1.8.1-cp39-cp39-linux_armv7l.whl
- pip install torchvision-0.9.1-cp39-cp39-linux_armv7l.whl
- pip install opencv-python
- pip install ultralytics
- python yolov8.py => ok
- deactivate # 離開虛擬環境

Virtualenv

- virtualenv is a tool to create isolated Python environments

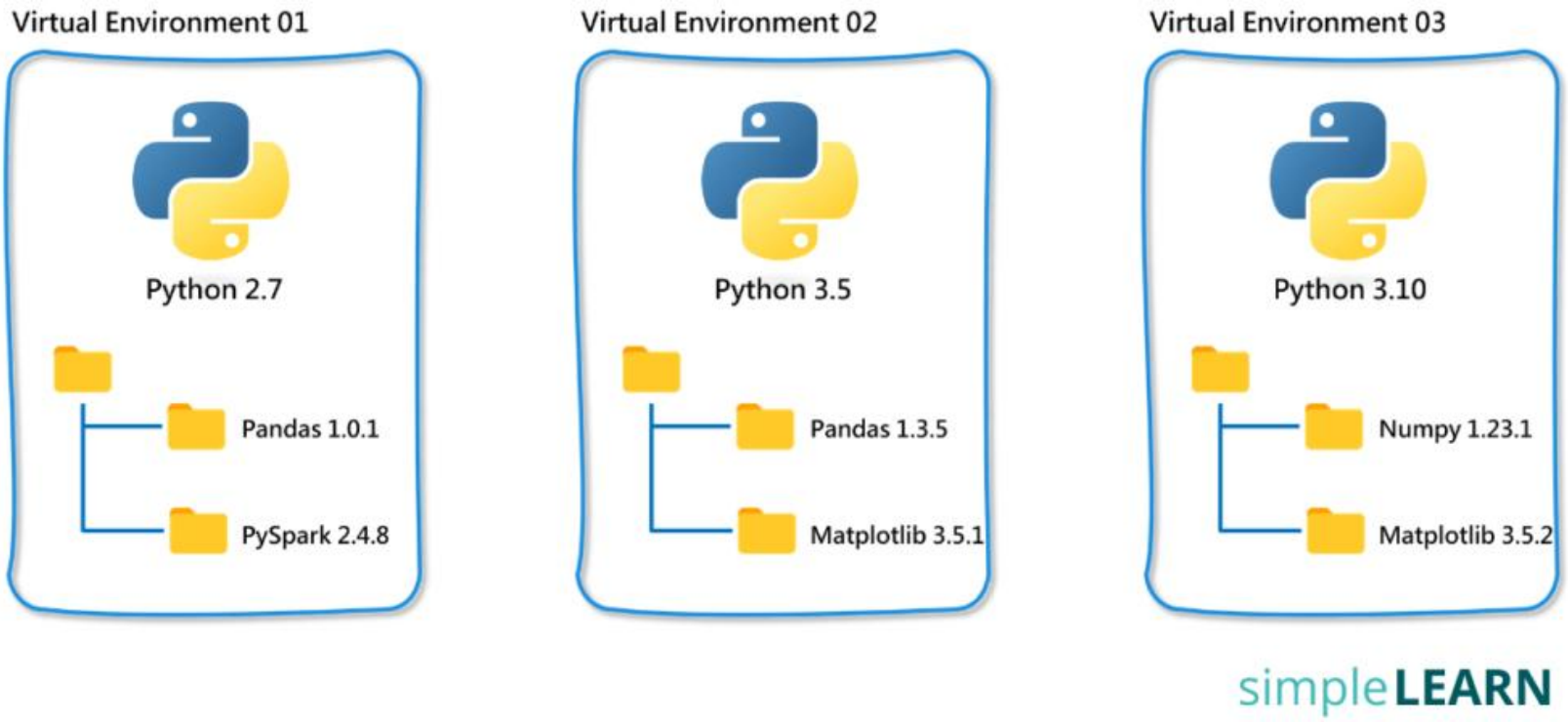


Fig source: https://simplelearn.tw/2022/08/08/ python-虛擬環境完整教學-1｜virtualenv-虛擬環境安裝及使用/

Command: pip list

- List installed packages, including editables.
- Packages are listed in a case-insensitive sorted order.

```
pi@raspberrypi:~ $ pip3 list
Package                 Version
----------------------  -----------
absl-py                 1.4.0
anyio                   3.6.2
argon2-cffi             21.3.0
argon2-cffi-bindings    21.2.0
…
…
wheel                   0.40.0
widgetsnbextension      4.0.7
wrapt                   1.10.11
zipp                    3.15.0
(# end)
```

Virtualenv

- List installed packages under a new virtualenv

```
pi@raspberrypi:~ $ source tflite-env/bin/activate
(tflite-env) pi@raspberrypi:~ $ pip3 list
Package            Version
-------------      -------
pip                18.1
pkg-resources      0.0.0
setuptools         40.8.0
```

- Leave command: deactivate

```
(tflite-env) pi@raspberrypi:~ $ deactivate
pi@raspberrypi:~ $
```

Summary

- Practice Lab: 整合視覺資料 (攝影機, 影像辨識)
- Write down the answer for discussion 1-3
  - D1: How to calculate the area summation by integral image?
  - D2: How to modify the sample code to show class id? (TF lite)
  - D3: Use YOLO analyze your image
- Quiz1:
  - Design drowsiness detection based on facial landmark
- Next week (4/18):
  - 期中考Midterm
  - Project分組 (開始找組員)