BDSE-06

OpenCV 與樹莓派影像辨識實作

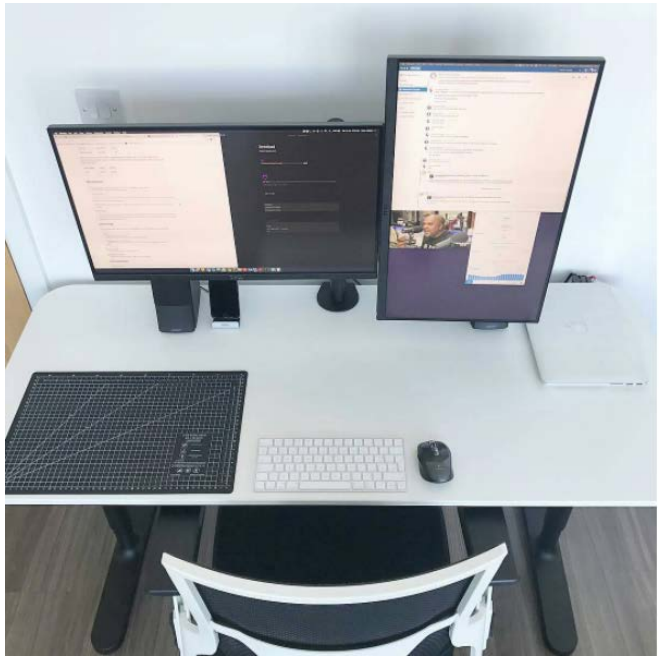
OpenCV & Raspberry Pi Image Recognition Implementation

Presenter : Tim 邢博竣

2017/11/13

Outline

- OpenCV
  - What is OpenCV ?
  - Haar Feature Cascade
- RaspBerry Pi
  - What is Raspberry Pi ?
  - Specification
- Cloud Camera Implementation
  - Imagga API
  - CloudCamera.py
- Reference



BDSE-06


OpenCV

OpenCV

- What is OpenCV ?
  - Open source Computer Vision library
  - The library is written in C (v1.x) and C++(v2.x)
  - Runs under Linux, Windows and Mac OS X
  - Active development on interfaces for:
    - Python, Ruby, Matlab, and other languages.



python™

Ruby PROGRAMMING Language

MATLAB

OpenCV

- What is OpenCV ?
  - 從Intel 1999年發布OpenCV以來，功能仍正在持續更新與增加中
  - 目前最新的版本為 3.3.1 (Oct. 2017)
  - 讓電腦視覺有更低的門檻，並充分利用 Intel 處理器的運算效能
  - Timeline:

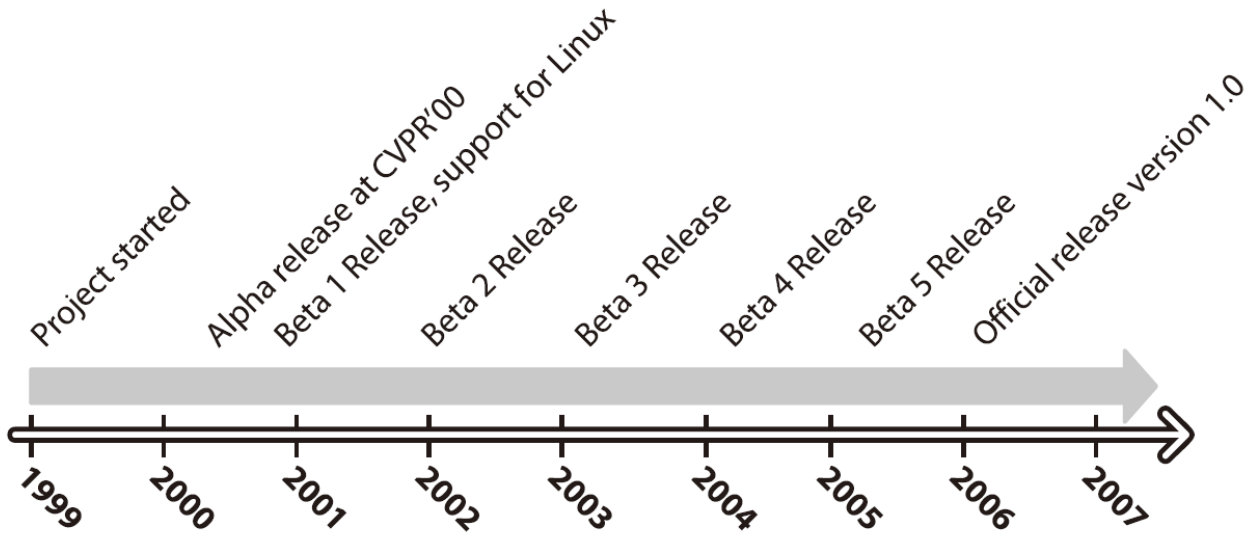


Figure 1. OpenCV timeline

Image Processing

- Filters
- Edge Maps
- Transformation
- Segmentation

Feature

- Detection
- Tracking
- Matching

And More

- Machine Learning
- Image Pyramids
- Matrix Math
- Stereoscopic 3D

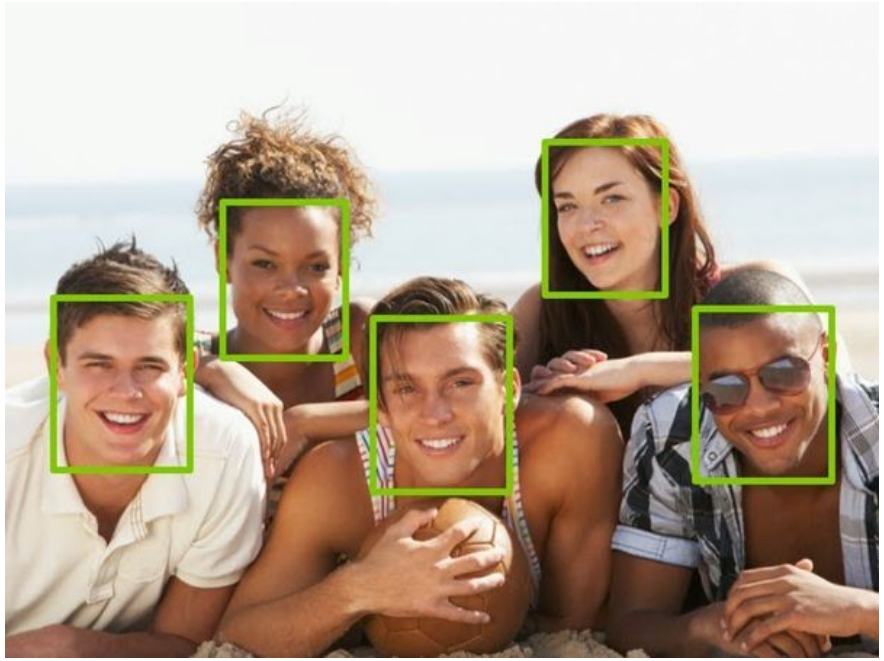
OpenCV

- 特徵提取
  - 對於電腦來說，圖像只是一堆數值的集合
  - 為了使電腦能夠像人類的視覺一樣，透過觀察理解圖像，我們研究如何從圖像的像素值，抽取有用的數據或訊息，來描述這個圖像或物體
  - 利用這些特徵通過訓練來讓電腦如何理解這些特徵

OpenCV

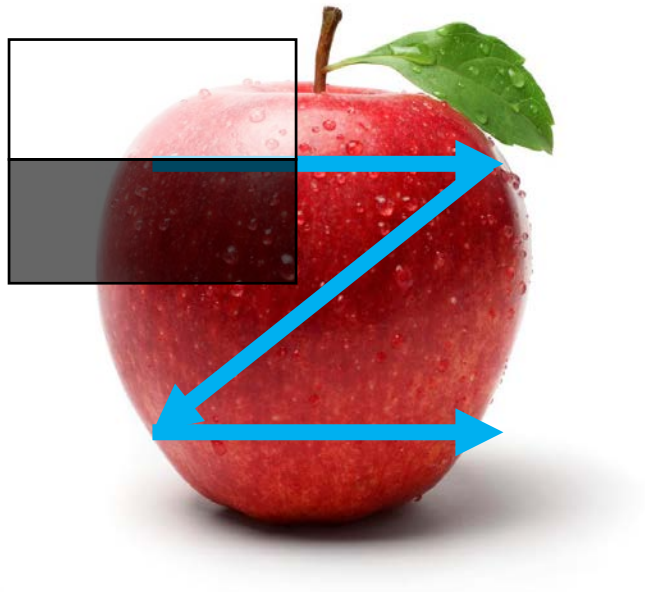
Haar Feature Cascade

- 採監督式學習的類神經網路演算法, 並有以下特色:
  - 特徵比對(Haar features)
  - 積分影像計算(Integral Image)
  - 串接分類器(Cascade)
  - 學習機制(AdaBoost)



OpenCV

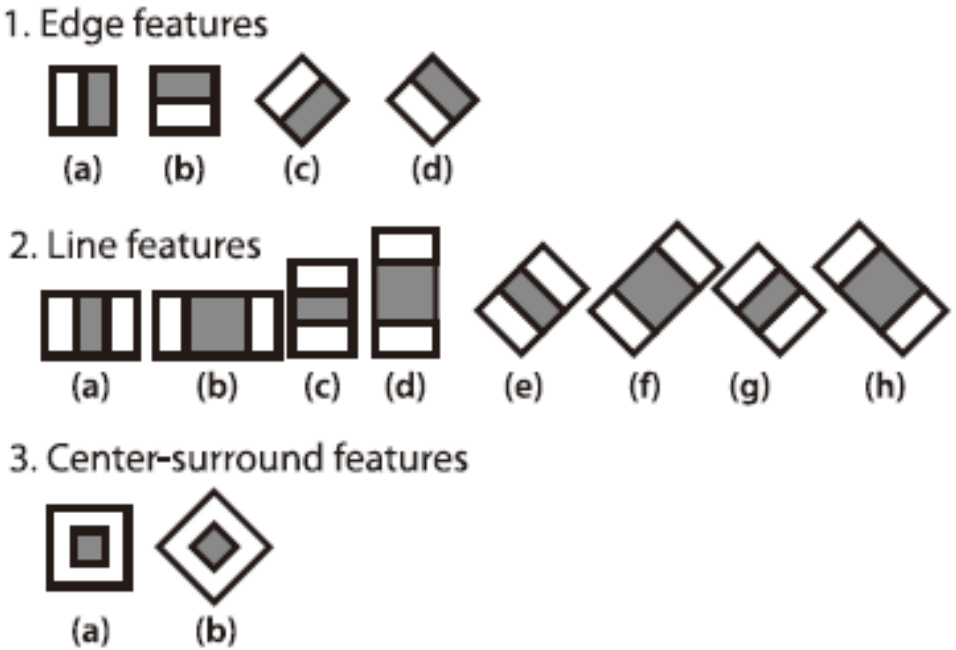
- 特徵比對
  - Pick a scale (ex: 24x24 pixels) for the feature
  - Slide it across the image
  - Compute the average pixel values under the white area and the black area
  - If the difference between the areas is above some threshold, the feature matches



OpenCV

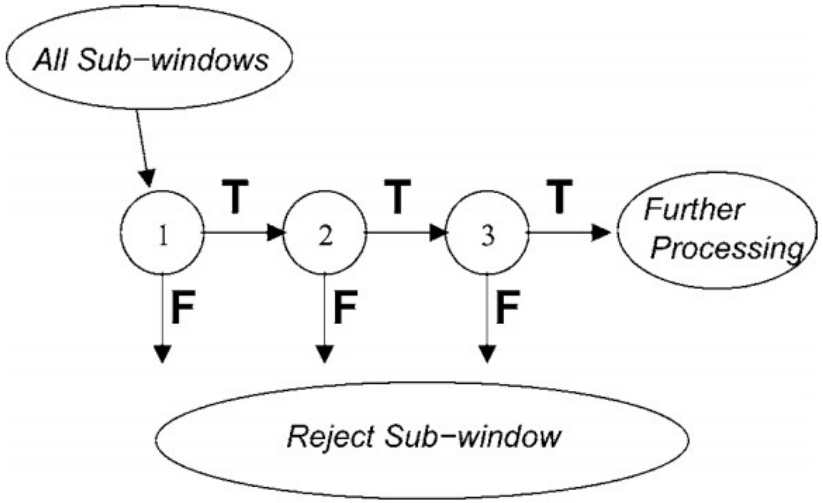
- 弱分類器
  - A single classifier isn't accurate enough
    - It's called a "weak classifier "

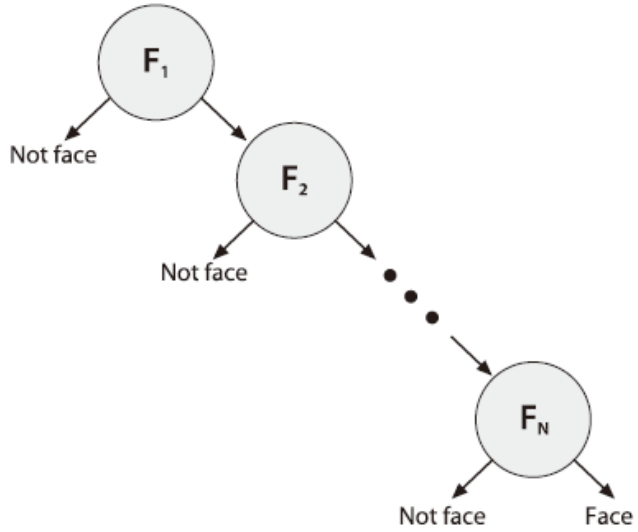
- Feature:  and 
- Classifier:  

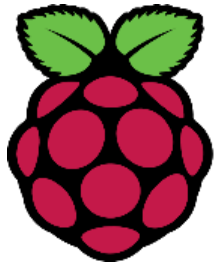


OpenCV

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







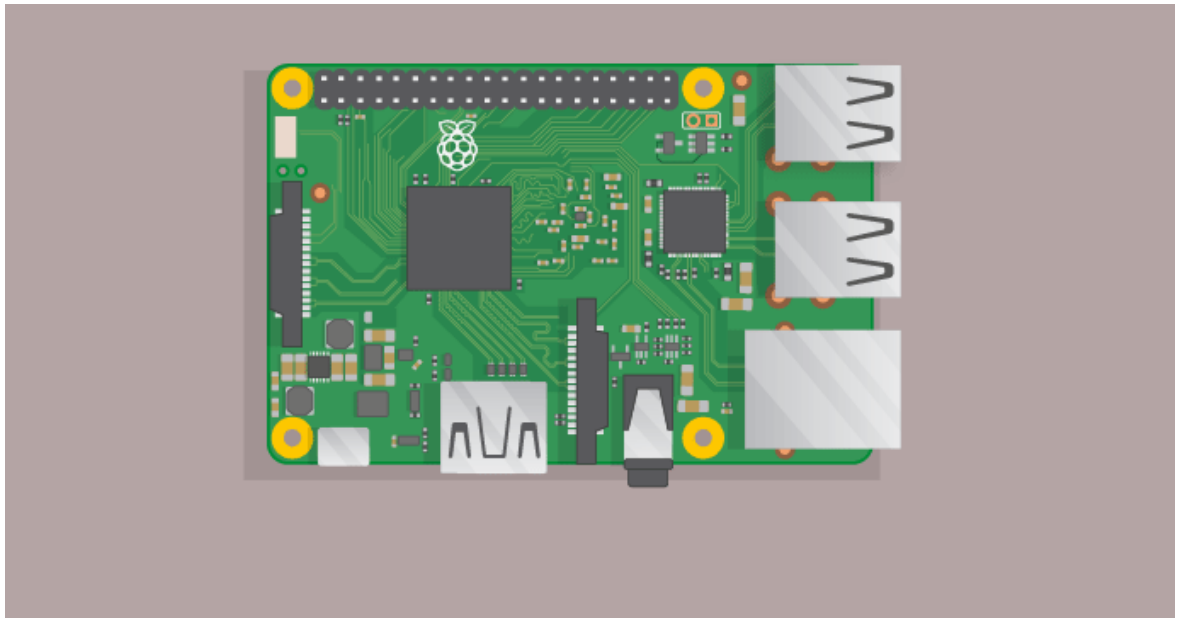
Raspberry Pi

Raspberry Pi

- What is Raspberry Pi ?
  - A single-board computers
  - Add a computer with a OS to practically anything
  - Support for a vast array of peripherals (thanks to the Linux kernel)
  - An IOT terminal device
  - Programing education
  - Powerful community resources

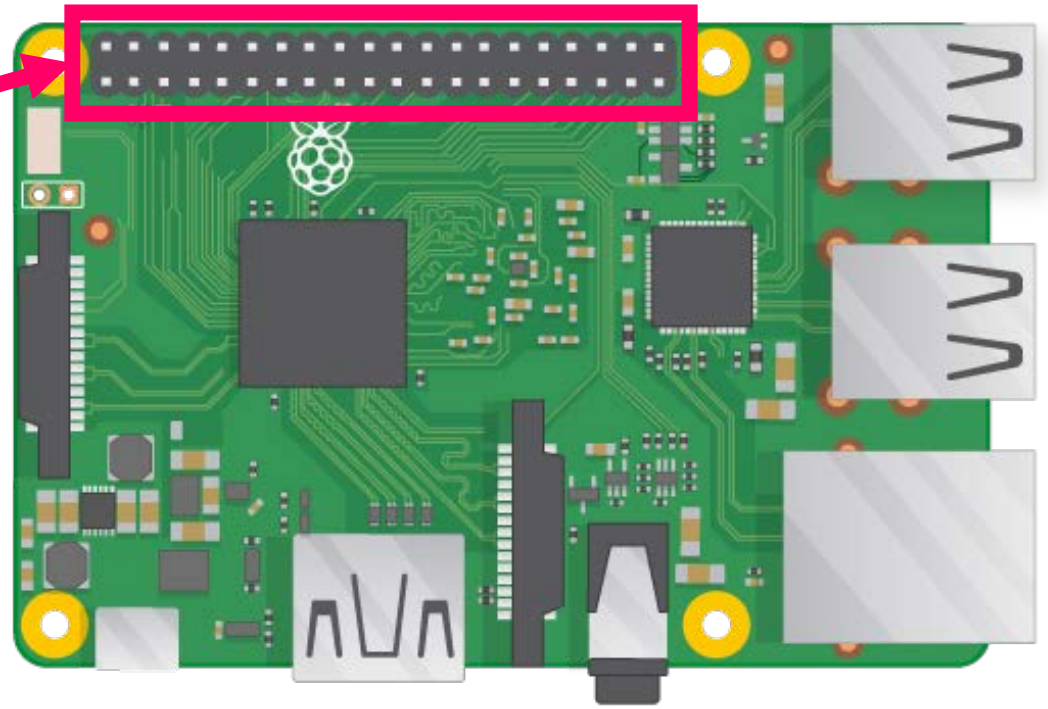
Raspberry Pi

- What we need:



Raspberry Pi

- Specification:
  - A 1.2GHz 64-bit quad-core ARMv8 CPU
  - 802.11n Wireless LAN
  - Bluetooth 4.1
  - 4 USB ports
  - 40 GPIO pins
    - General-purpose input/output
  - Full HDMI port
  - Ethernet port



Raspberry Pi

- Raspberry official supported Operating System:

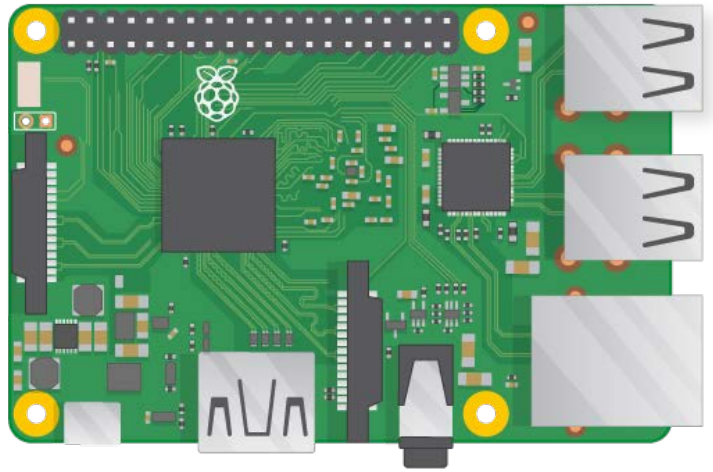
Raspbian

- Third Party Operating System







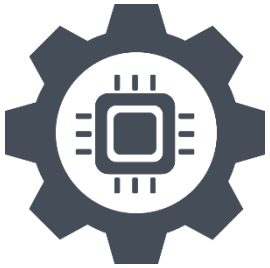
BDSE-06

Cloud Camera Implementation

Cloud Camera Implementation

- imagga
  - An Image Recognition Platform-as-a-Service providing Image Tagging APIs for developers & businesses to build scalable, image intensive cloud apps.

Save
time & money



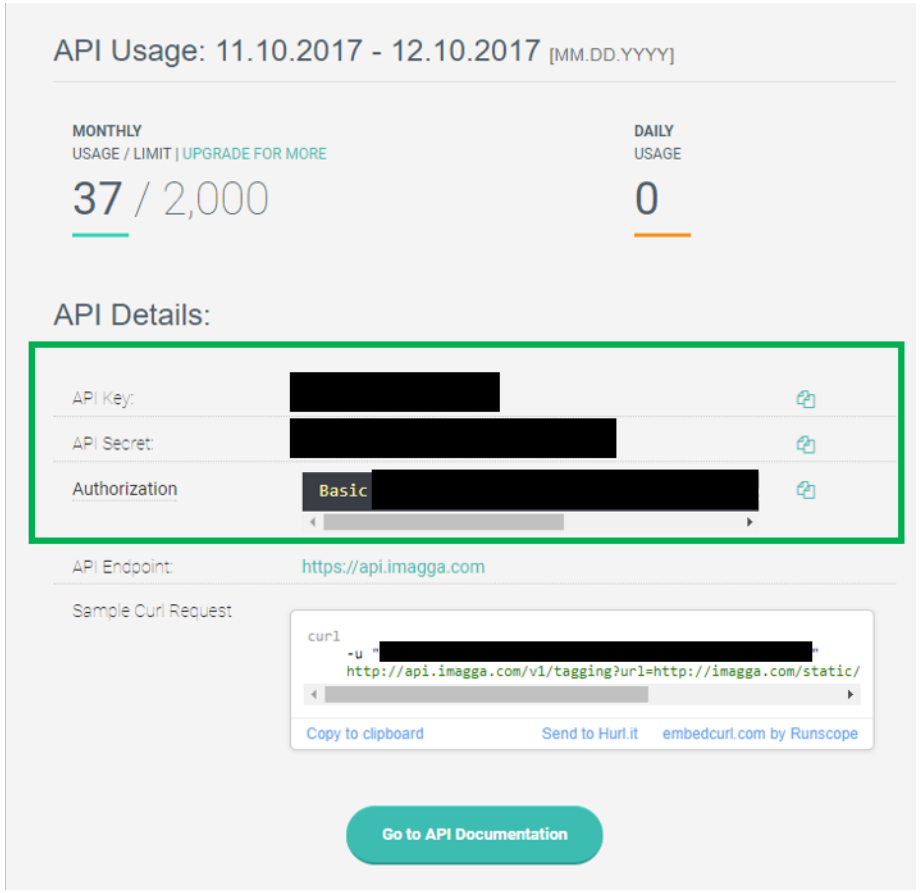
Powerful
image recognition

Self-learning
solution

Cloud Camera Implementation

- Imagga API
  - /tagging
  - /categorizers
  - /croppings
  - /content(/<content_id>)

API Key
API Secret
Authorization



Cloud Camera Implementation

- Imagga API
  - /tagging

- Request:

```
import requests

api_key = ''
api_secret = ''
image_url = 'https://imagga.com/static/images/tagging/wind-farm-
538576_640.jpg'

response = requests.get('https://api.imagga.com/v1/tagging?url=%s' %
image_url,
                        auth=(api_key, api_secret))

print response.json()
```

- Response:


```
{
    "results": [
        {
            "tagging_id": null,
            "image": "https://imagga.com/static/images/tagging/wind-farm-
538576_640.jpg",
            "tags": [
                {
                    "confidence": 38.51164403389053,
                    "tag": "sky"
                },
                {
                    "confidence": 25.775729223221116,
                    "tag": "turbine"
                },
                {
                    "confidence": 20.417771132010046,
                    "tag": "landscape"
                },
```

Cloud Camera Implementation

- Imagga API
  - /content(/<content_id>)
    - Request:

```
import requests

api_key = ''
api_secret = ''
image_path = '/path/to/your/image.jpg'

response = requests.post('https://api.imagga.com/v1/content',
auth=(api_key, api_secret),
files={'image': open(image_path, 'r')})
print response.json()
```

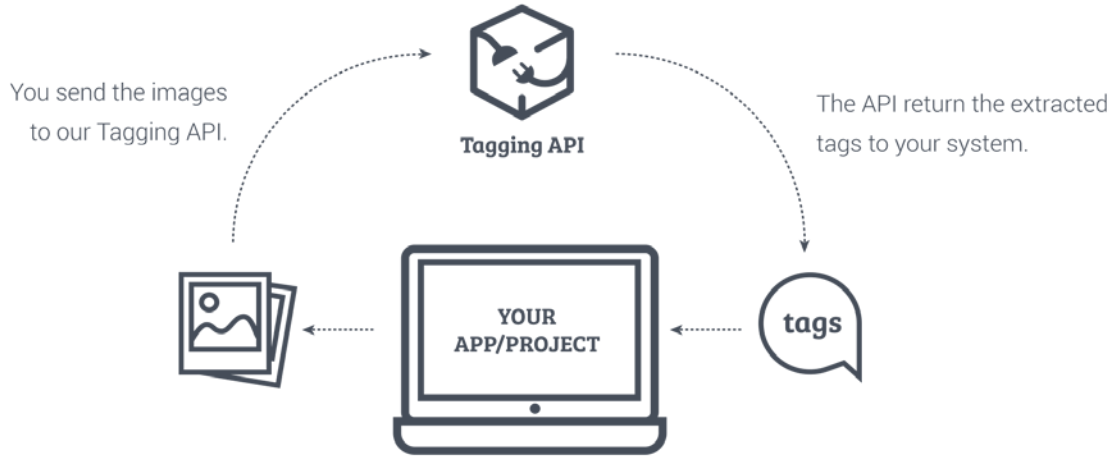
    - Response:

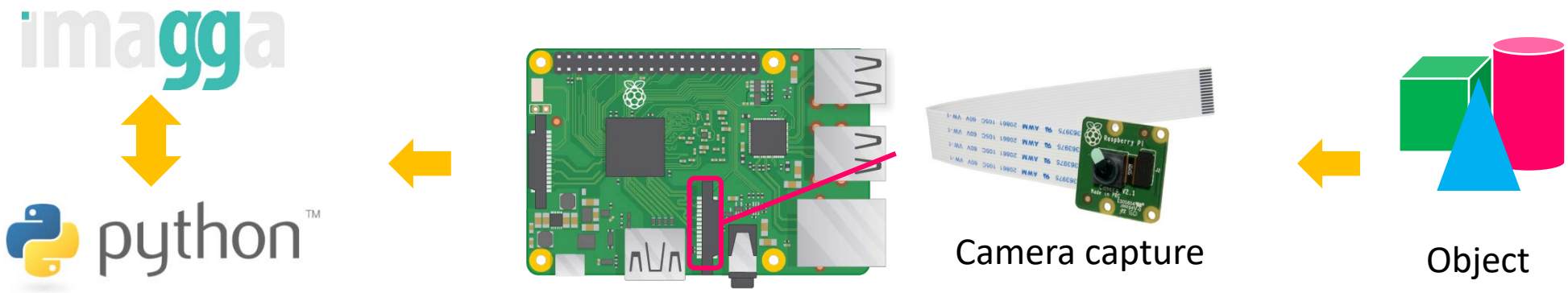
```
{
    "status": "success",
    "uploaded": [
        {
            "id": "f86af4d8a6f1abec39d828579a6c4004",
            "filename": "image.jpg"
        }
    ]
}
```

Content ID

Cloud Camera Implementation

- How to implement ?
  - Capture object by camera
  - Upload image to imagga
  - Requests imagga tagging
  - Show the top 5 confidence tags
  - Use TTS to transfer text (the highest confidence) to speech





Cloud Camera Implementation

- CloudCamera.py
  - Upload file

```
def imaggaUpload(filename):
      import requests
      img = filename
      api_key = [redacted]
      api_secret = [redacted]

      responseContentID = requests.post('https://api.imagga.com/v1/content',
                        auth=(api_key, api_secret),
                        files = {"file": open(img,"r")})

      contentID=responseContentID.json()["uploaded"][0]["id"]
      uploadStatus = responseContentID.json()["status"]
      return contentID
      print("Upload status: "+uploadStatus)
      print("ContentID: "+contentID)
```

status: success

ContentID: c4b1c6aeed2384217473cbc61d7da3a3

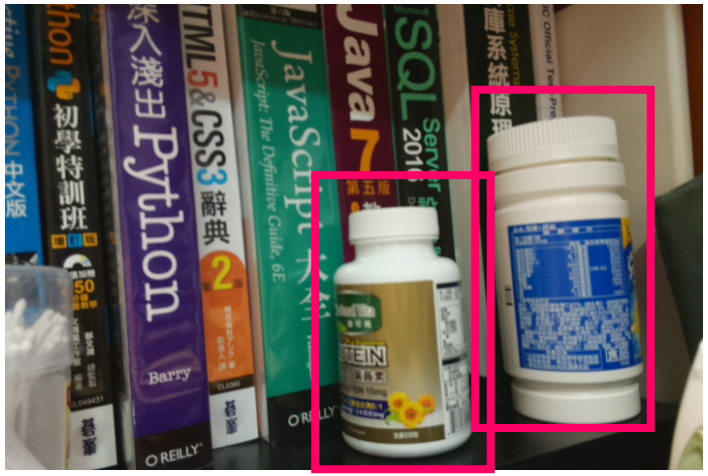
Cloud Camera Implementation

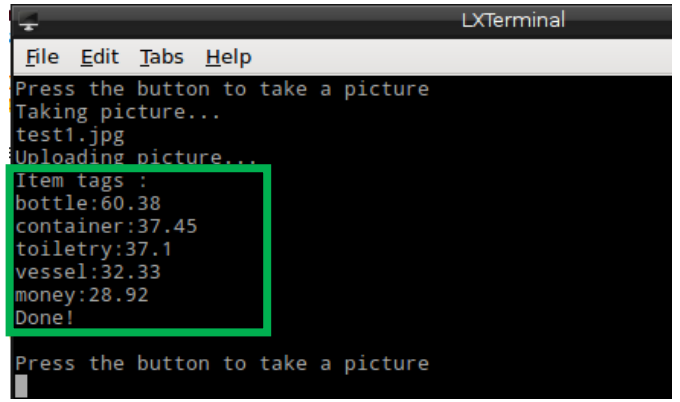
- CloudCamera.py
  - Tagging

```
def imaggaTag(contentID):
      import requests
      cID = contentID
      api_key = [redacted]
      api_secret = [redacted]

      responseTag = requests.get('https://api.imagga.com/v1/tagging?content=%s' % cID,
                                                        auth=(api_key, api_secret))
      tags = responseTag.json()["results"][0]["tags"]
      print"Item tags : "
      for i in range(0,5):
            print
      tags[i]["tag"].encode("ascii")+":"+str(round(tags[i]["confidence"],2));

      result = tags[0]["tag"].encode("ascii")
      cofidence = str(round(tags[0]["confidence"],2)) +"%";
      t2 = "Item is " +result+", which confidence is "+cofidence+"."
      os.system("sudo speech.sh "+t2+" >/dev/null 2>&1")
```





Bottle:60.38%
Container:37.45%
Toiletry:37.1%

Cloud Camera Implementation

- CloudCamera.py
  - Camera & GPIO Input init

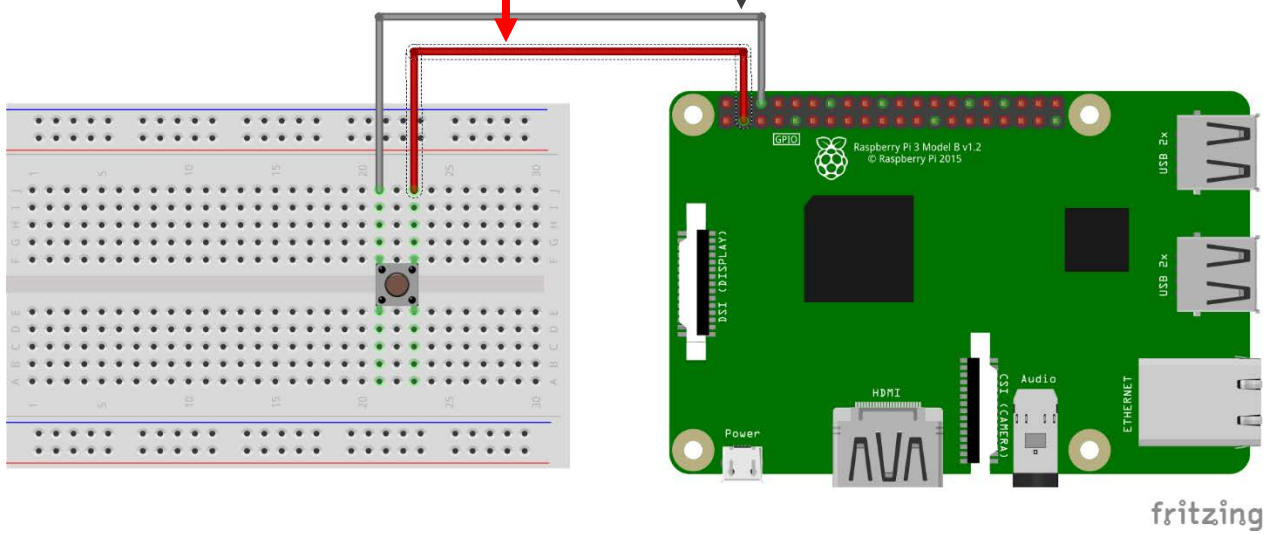
```
import time, os
import RPi.GPIO as GPIO

#camera init
camera = cameraInit()
#GPIO init
GPIO.setmode(GPIO.BOARD)
buttomPin=3
GPIO.setup(buttomPin,GPIO.IN)
prev_input=1
i=0
#set file name
filename="test"
pNum=1
```

Set GPIO Pin3 as input pin

Button to GPIO Pin6 GND

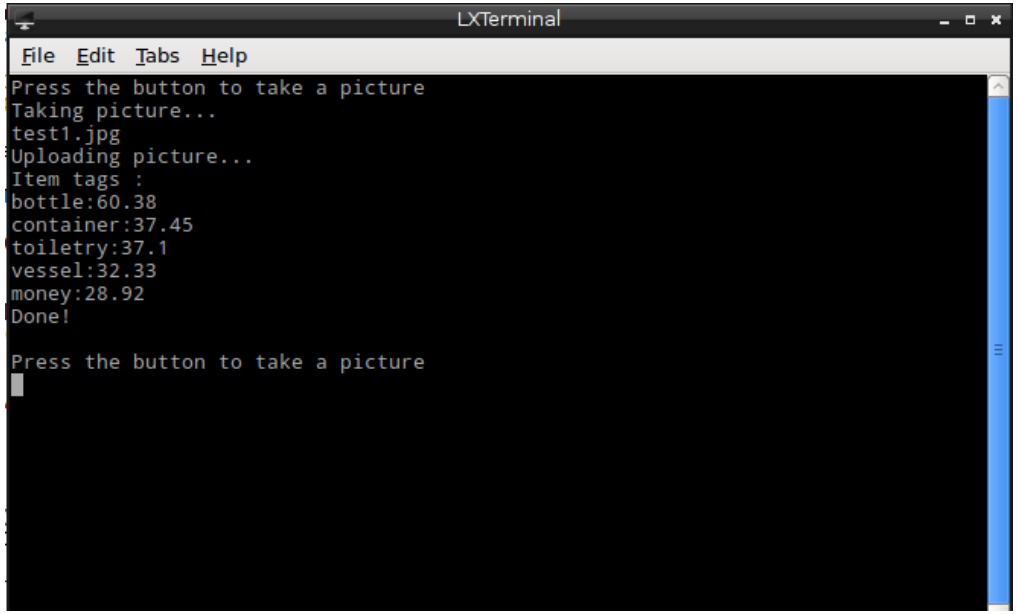
GPIO Pin3 to button

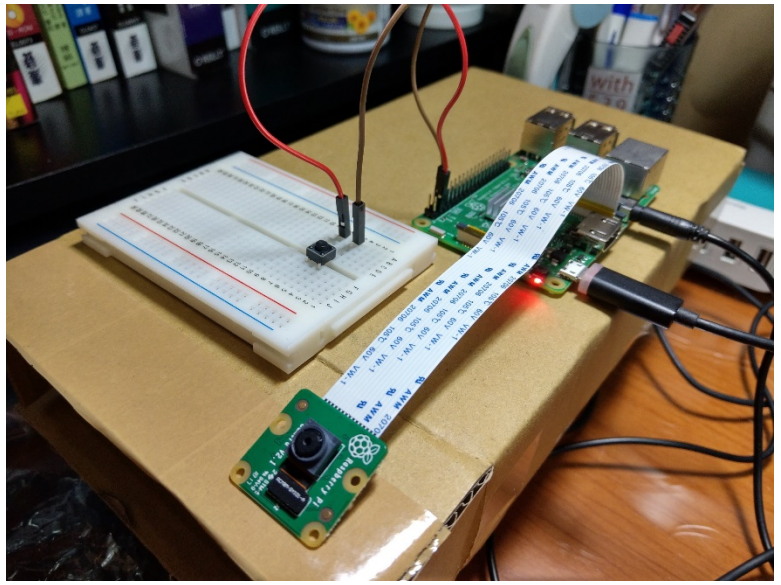


Cloud Camera Implementation

- CloudCamera.py
  - Main

```
print "Press the button to take a picture"
while True:
     input=GPIO.input(buttomPin)
     if ((not prev_input) and input):
          i += 1
          print "Taking picture..."
          image = filename+str(pNum)+".jpg"
          takePicture(camera,image)
          print image
          print "Uploading picture..."
          imageID = imaggaUpload(image)
          imaggaTag(imageID)
          print "Done!\n"
          print "Press the button to take a picture"
          pNum=pNum+1
     prev_input=input
     time.sleep(0.1)
```





Reference

- OpenCV中文網站
  http://wiki.opencv.org.cn/index.php/%E9%A6%96%E9%A1%B5
- OpenCV https://opencv.org
- Imagga https://imagga.com
- Raspberry Pi 臺灣樹梅派 https://www.raspberrypi.com.tw/tag/opencv/
- Raspberry Pi https://www.raspberrypi.org/

BDSE-06

DEMO

The end.
Thanks for your attention!!