熱像儀

CC (Creative Commons)

姓名標示 — 非商業性 — 相同方式分享

姓名標示 — 你必須給予 適當表彰、提供指向本授權條款的連結，以及 指出（本作品的原始版本）是否已被變更。你可以任何合理方式為前述表彰，但不得以任何方式暗示授權人為你或你的使用方式背書。

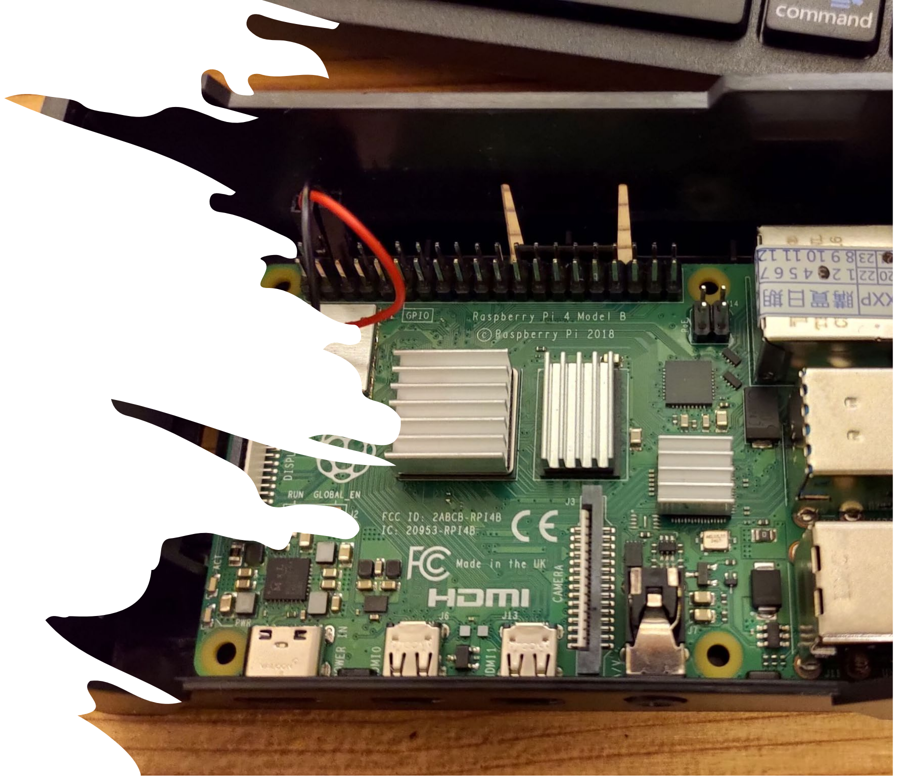
非商業性 — 你不得將本素材進行商業目的之使用。

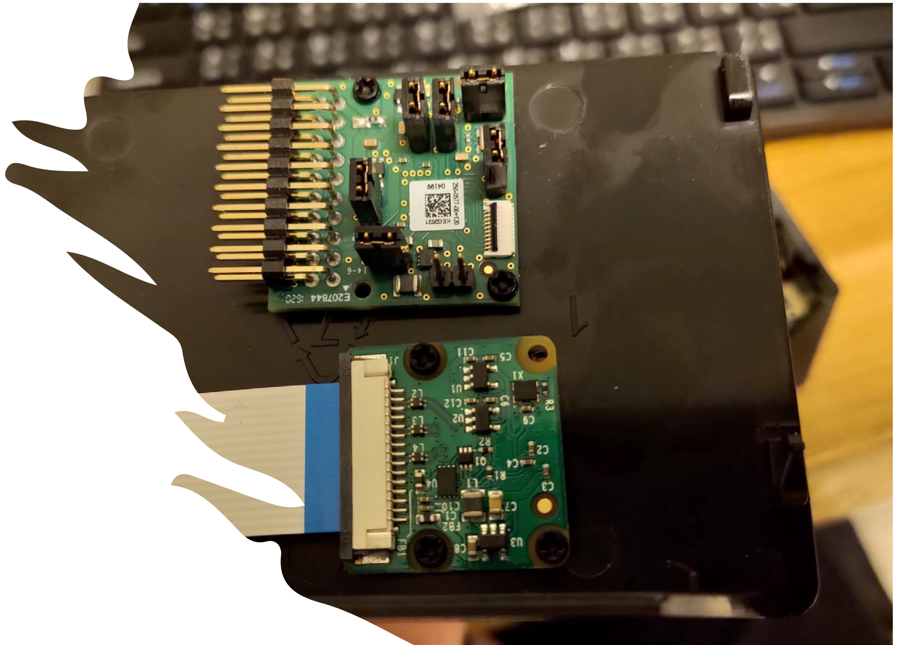
相同方式分享 — 若你重混、轉換本素材，或依本素材建立新素材，你必須依本素材的授權條款來散布你的貢獻物。

本教材改編自台灣樹莓派–用樹莓派自製防疫熱像

Outline

1. v4l2
   1. OpenCV 簡介
   2. v4l2 簡介
   3. v4l2 安裝
   4. v4l2 測試
2. pyLepton
3. Thermal pi
4. Fusion

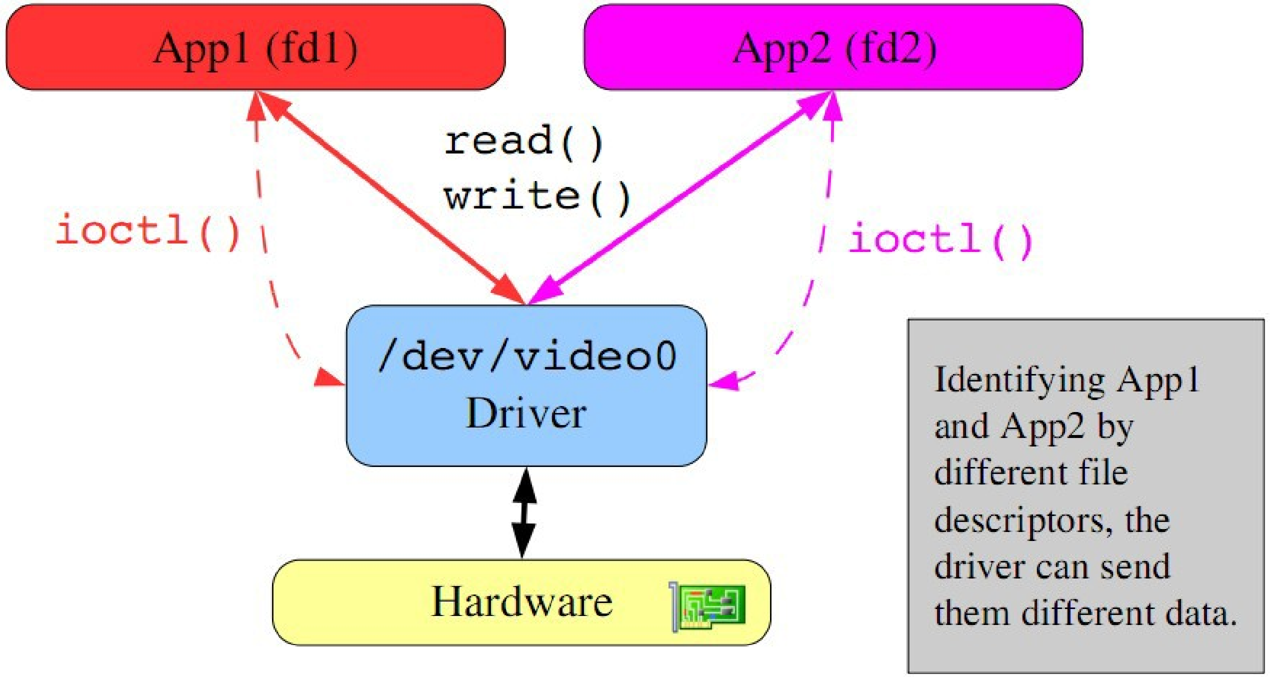




使用 V4L2 讀取影像

Video For Linux 2nd(V4L2)

- 是 Linux 對視訊設備 (如 Webcam) 的 Userspace API



Identifying App1 and App2 by different file descriptors, the driver can send them different data.

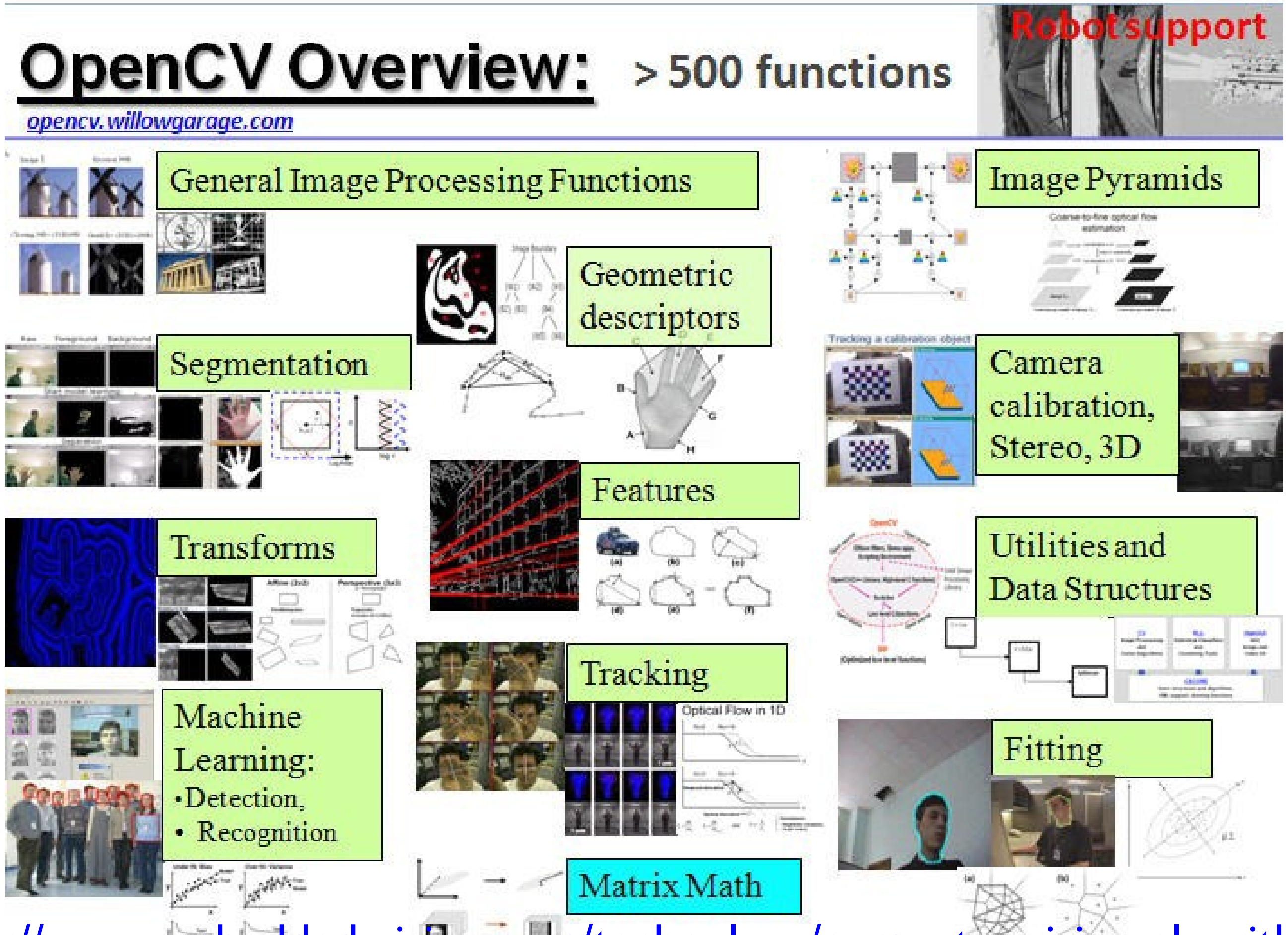
http://free-electrons.com/doc/embedded_linux_multimedia.pdf
https://www.linuxtv.org/downloads/legacy/video4linux/API/V4L2_API/spec-single/v4l2.html

使用 OpenCV 讀取，處理影像

OpenCV

- Open Source Computer Vision Library

- 跨平台的計算機函式庫，主要由 C/C++ 撰寫



OpenCV Overview: > 500 functions

opencv.willowgarage.com

Robot support

General Image Processing Functions

Image Pyramids

Geometric descriptors

Segmentation

Camera calibration, Stereo, 3D

Features

Transforms

Utilities and Data Structures

Tracking

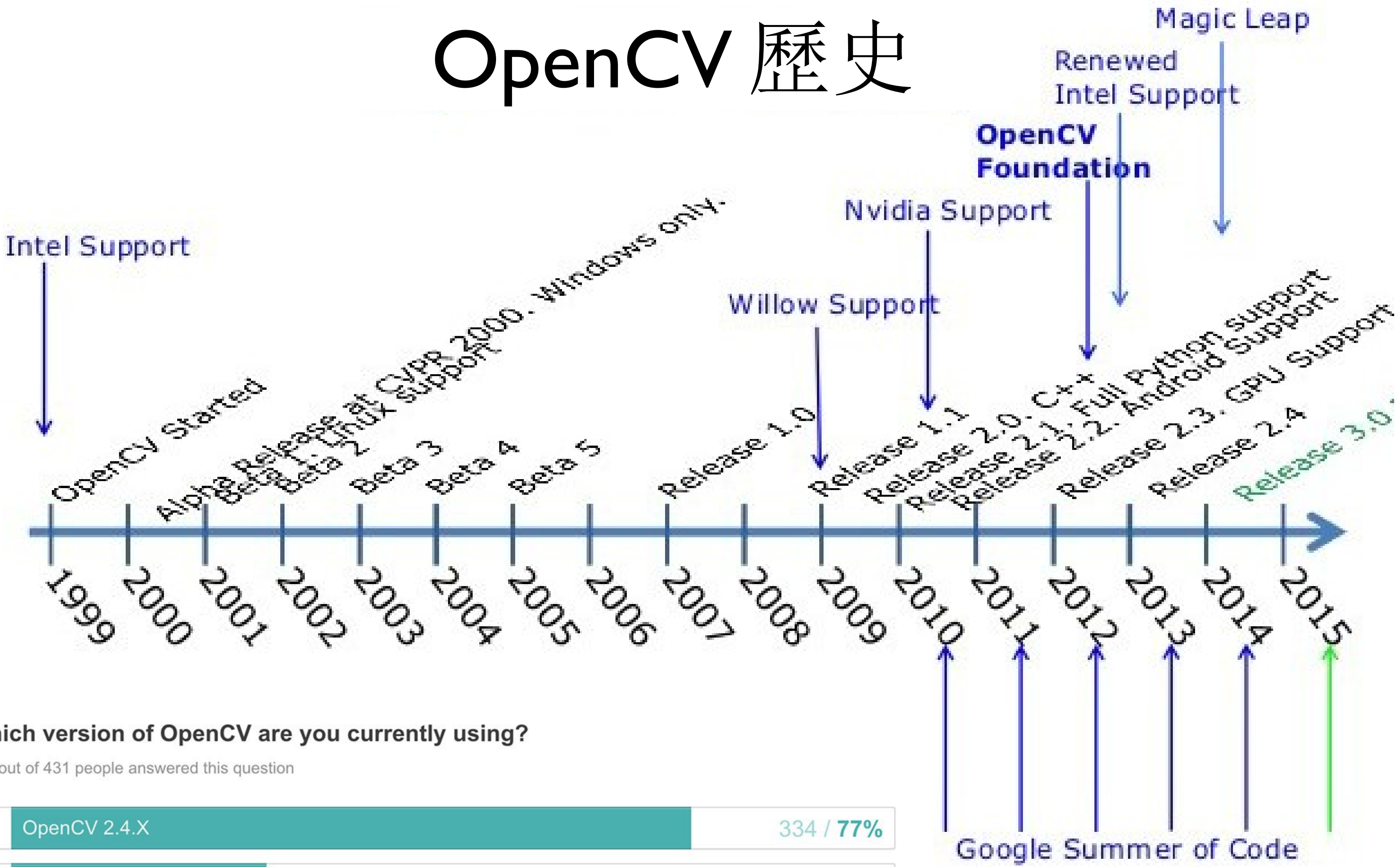
Machine Learning:
·Detection,
• Recognition

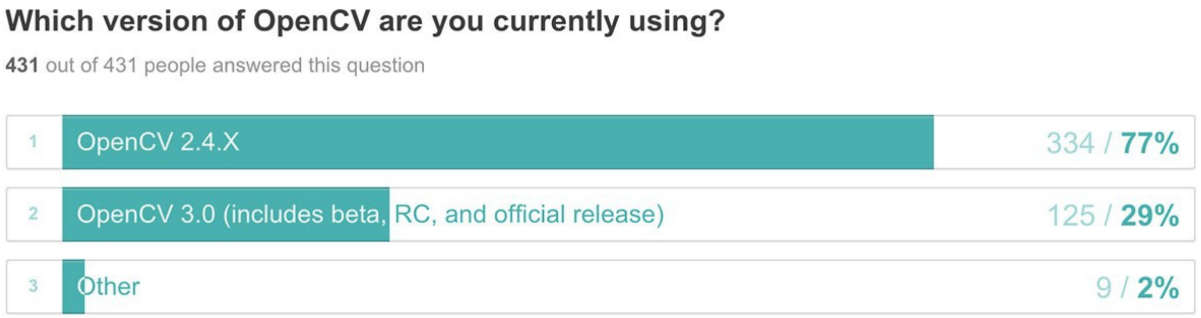
Fitting

Matrix Math

http://www.embedded-vision.com/technology/computer-vision-algorithms

OpenCV 歴史





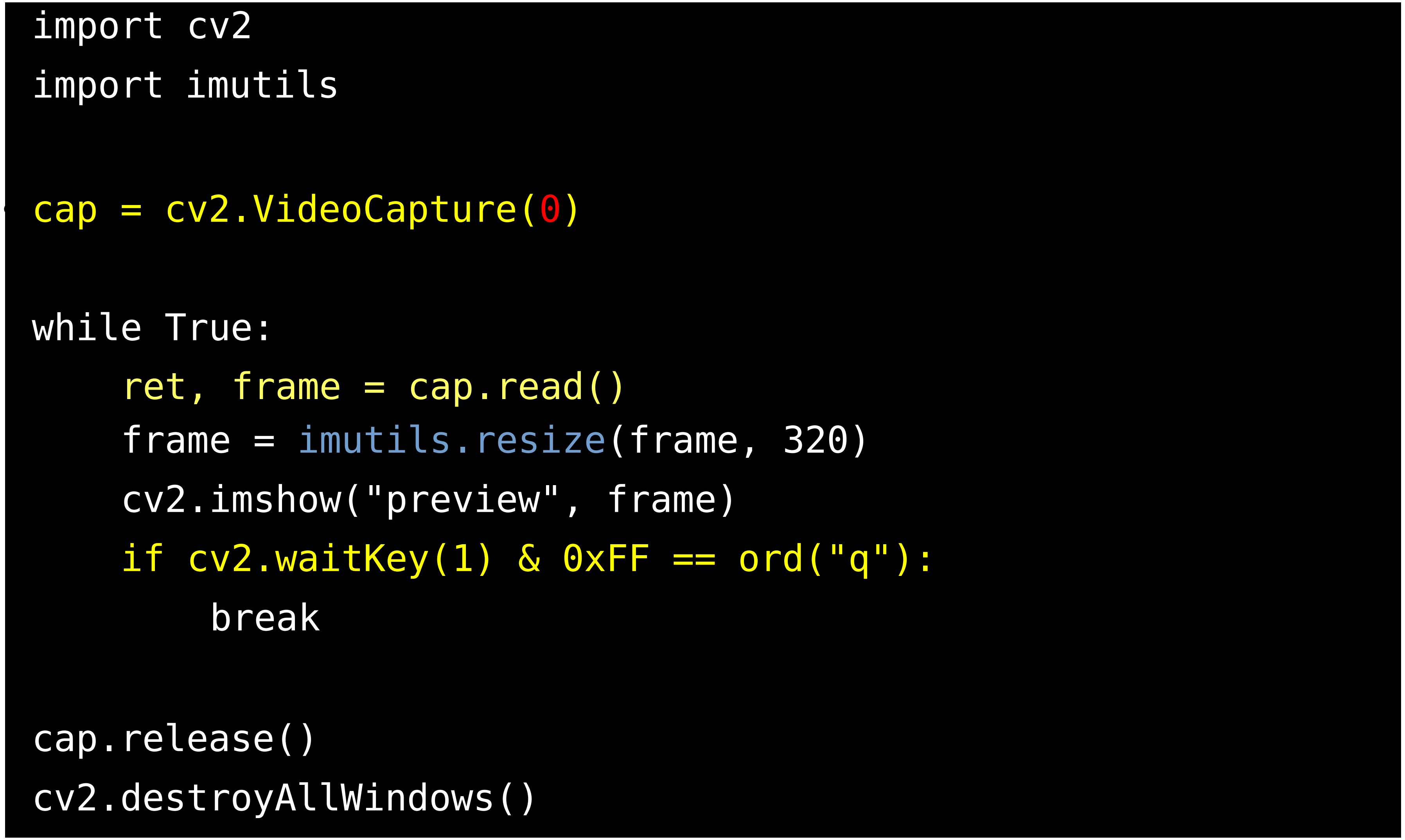
Which version of OpenCV are you currently using?

431 out of 431 people answered this question

1	OpenCV 2.4.X	334 / **77%**
2	OpenCV 3.0 (includes beta, RC, and official release)	125 / **29%**
3	Other	9 / **2%**

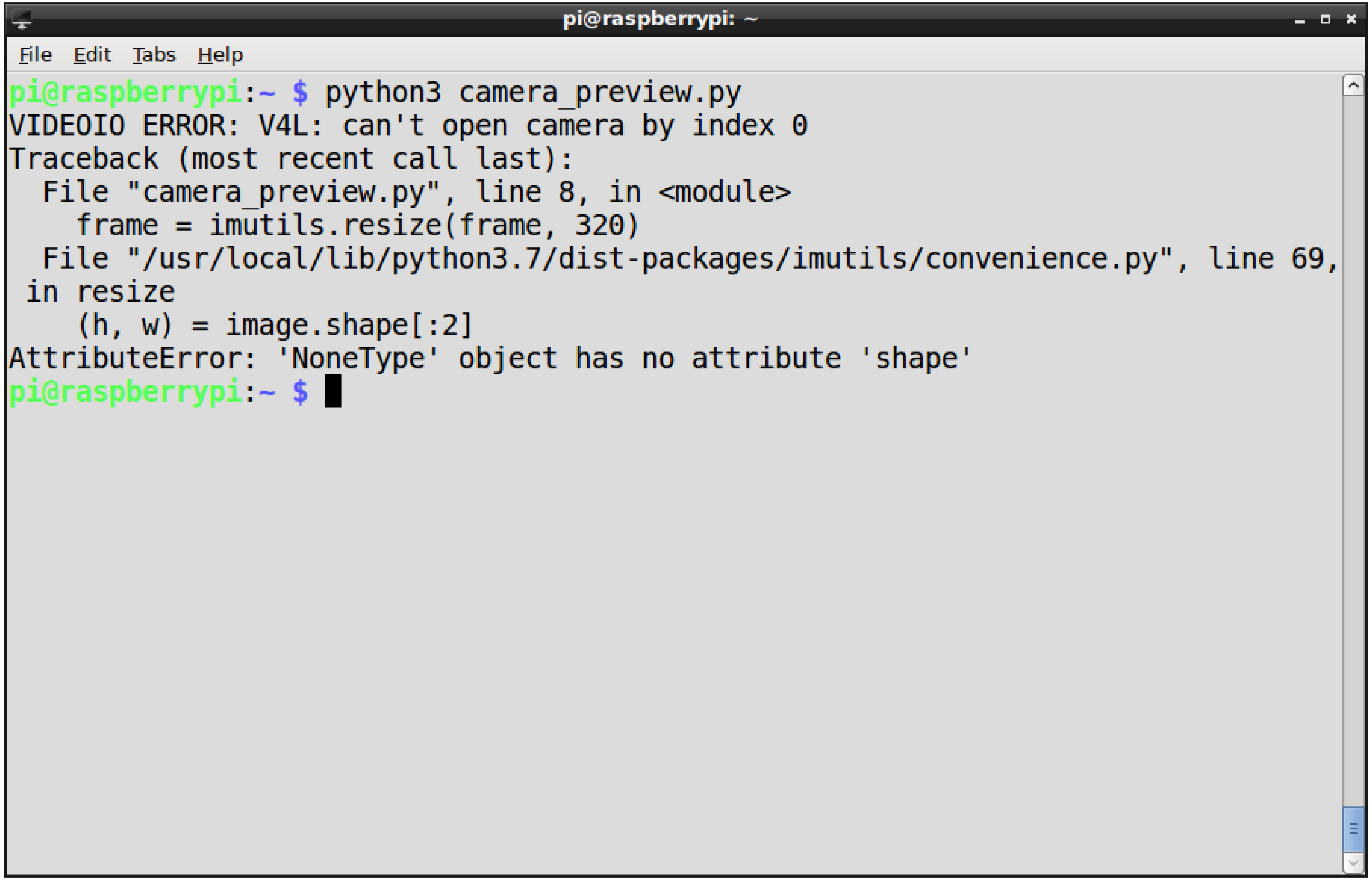
https://www.slideshare.net/embeddedvision/e04-open-cvbradsk

讀取 Camera 並顯示



```python
import cv2
import imutils

cap = cv2.VideoCapture(0)

while True:
    ret, frame = cap.read()
    frame = imutils.resize(frame, 320)
    cv2.imshow("preview", frame)
    if cv2.waitKey(1) & 0xFF == ord("q"):
        break

cap.release()
cv2.destroyAllWindows()
```

執行結果



```
pi@raspberrypi:~ $ python3 camera_preview.py
VIDEOIO ERROR: V4L: can't open camera by index 0
Traceback (most recent call last):
  File "camera_preview.py", line 8, in <module>
    frame = imutils.resize(frame, 320)
  File "/usr/local/lib/python3.7/dist-packages/imutils/convenience.py", line 69,
 in resize
    (h, w) = image.shape[:2]
AttributeError: 'NoneType' object has no attribute 'shape'
pi@raspberrypi:~ $
```

用 V4L2 讀取SPI 影像 (v4l2loopback)

- 步驟：

1. 安裝編譯 `v4l2loopback` 必要軟體
2. 編譯 `v4l2loopback` 虛擬裝置節點
3. 安裝 `V4L2 Kernel Module`
4. 使用 `V4L2` 讀取 `SPI` 影像

I. 安裝必要軟體(已安裝)

- `$ sudo apt-get update`
- `$ sudo apt-get install -y bc flex bison libncurses5-dev`
- `$ sudo wget https://raw.githubusercontent.com/notro/rpi-source/master/rpi-source -O /usr/bin/rpi-source && sudo chmod +x /usr/bin/rpi-source  && /usr/bin/rpi-source -q --tag-update`
- `$ rpi-source`

詢問問題時，都按 Enter(預設答案)

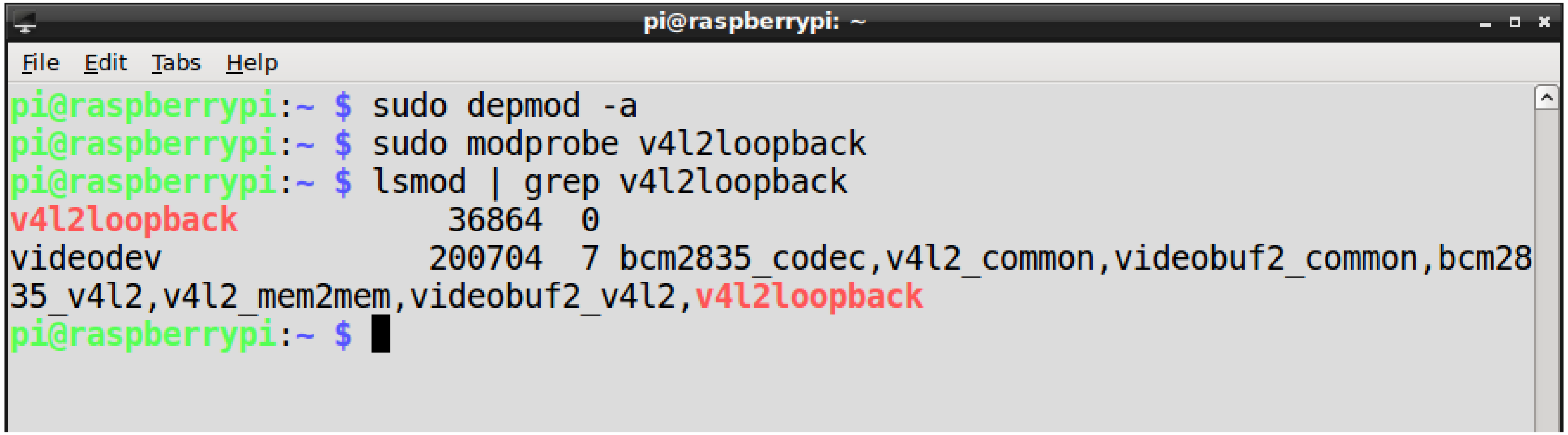
2. 編譯 v4l2loopback 虛擬裝置節點

```
$ cd ~
$ git clone https://github.com/umlaeute/v4l2loopback
$ cd ~/v4l2loopback
$ sudo make
$ sudo make install
```

3. 安裝 V4L2 Kernel Module

- `$ sudo depmod -a`
- `$ sudo modprobe v4l2loopback`
- `$ lsmod | grep v4l2loopback`

這三行每次開機都需要重新啟用



```
pi@raspberrypi:~ $ sudo depmod -a
pi@raspberrypi:~ $ sudo modprobe v4l2loopback
pi@raspberrypi:~ $ lsmod | grep v4l2loopback
v4l2loopback           36864  0
videodev              200704  7 bcm2835_codec,v4l2_common,videobuf2_common,bcm28
35_v4l2,v4l2_mem2mem,videobuf2_v4l2,v4l2loopback
pi@raspberrypi:~ $ 
```

https://github.com/raspberrypi-tw/meetup19

4. 將 SPI 讀取到的影像導到 V4L2 裝置

- `$ cd ~/LeptonModule/software/v4l2lepton`

  `==>  可能需要先修改v4l2lepton.cpp`

- `$ make`
- `$ sudo ./v4l2lepton -d /dev/spidev0.0 /dev/video1`

  ```
  Waiting for sink
  done reading, resets:
  ```

要看 v4l2lepton 裡的 /dev/video 決定X
已經有相機了，故選1

再次執行 camera_preview.py

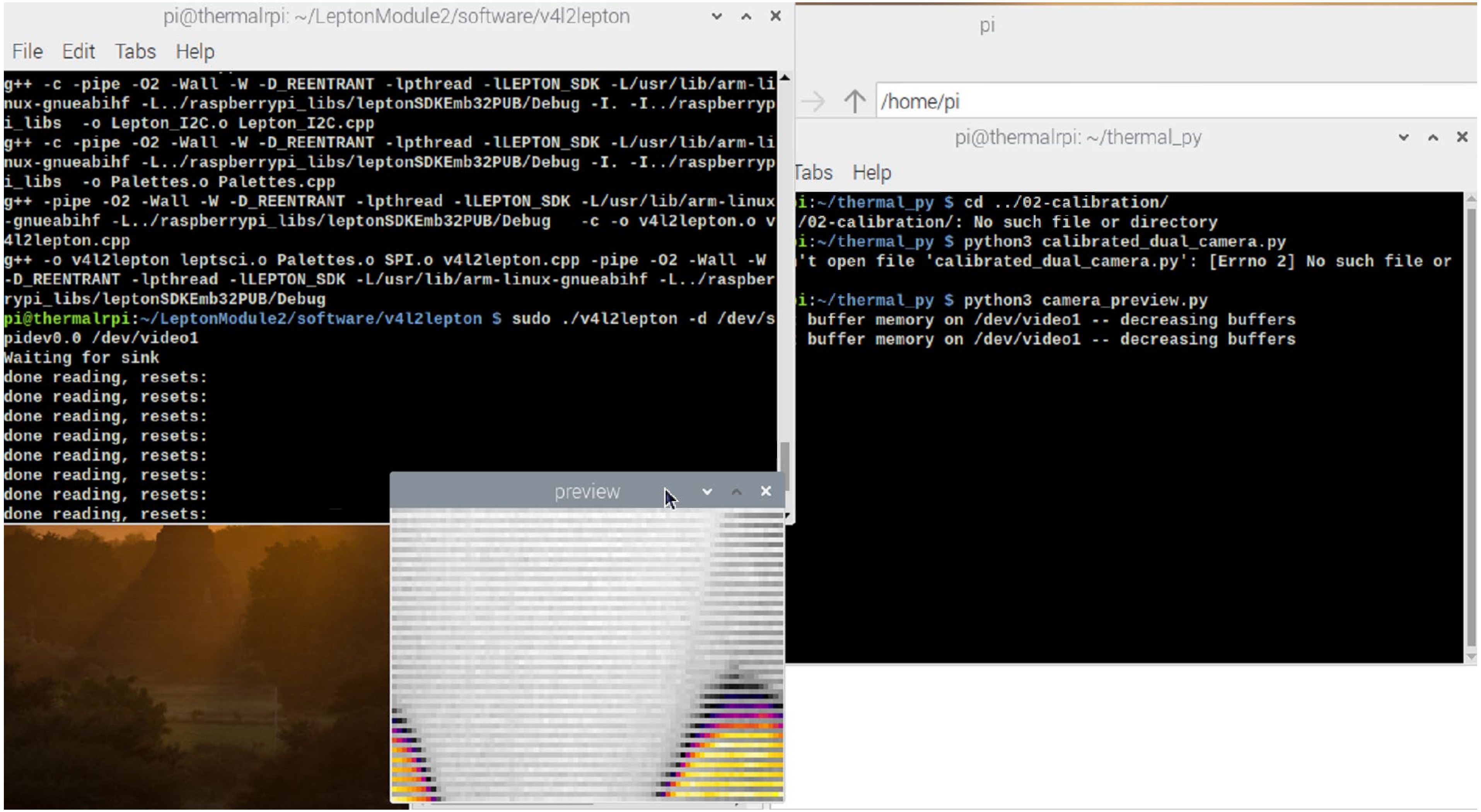
```
import cv2
import imutils

cap = cv2.VideoCapture(0)

while True:
    ret, frame = cap.read()
    frame = imutils.resize(frame, 320)
    cv2.imshow("preview", frame)
    if cv2.waitKey(1) & 0xFF == ord("q"):
        break

cap.release()
cv2.destroyAllWindows()
```

DEMO

camera_preview.py

```
$ cd ~/FLIR/thermal-pi/01-flir
$ python3 camera_preview.py
```

執行結果(開兩個視窗)



pi@thermalrpi: ~/LeptonModule2/software/v4l2lepton
File Edit Tabs Help
pi
/home/pi
pi@thermalrpi: ~/thermal_py
Tabs Help
preview

```
g++ -c -pipe -O2 -Wall -W -D_REENTRANT -lpthread -lLEPTON_SDK -L/usr/lib/arm-li
nux-gnueabihf -L../raspberrypi_libs/leptonSDKEmb32PUB/Debug -I. -I../raspberryp
i_libs  -o Lepton_I2C.o Lepton_I2C.cpp
g++ -c -pipe -O2 -Wall -W -D_REENTRANT -lpthread -lLEPTON_SDK -L/usr/lib/arm-li
nux-gnueabihf -L../raspberrypi_libs/leptonSDKEmb32PUB/Debug -I. -I../raspberryp
i_libs  -o Palettes.o Palettes.cpp
g++ -pipe -O2 -Wall -W -D_REENTRANT -lpthread -lLEPTON_SDK -L/usr/lib/arm-linux
-gnueabihf -L../raspberrypi_libs/leptonSDKEmb32PUB/Debug   -c -o v4l2lepton.o v
4l2lepton.cpp
g++ -o v4l2lepton leptsci.o Palettes.o SPI.o v4l2lepton.cpp -pipe -O2 -Wall -W
-D_REENTRANT -lpthread -lLEPTON_SDK -L/usr/lib/arm-linux-gnueabihf -L../raspber
rypi_libs/leptonSDKEmb32PUB/Debug
pi@thermalrpi:~/LeptonModule2/software/v4l2lepton $ sudo ./v4l2lepton -d /dev/s
pidev0.0 /dev/video1
Waiting for sink
done reading, resets:
done reading, resets:
done reading, resets:
done reading, resets:
done reading, resets:
done reading, resets:
done reading, resets:
done reading, resets:
```

```
i:~/thermal_py $ cd ../02-calibration/
/02-calibration/: No such file or directory
i:~/thermal_py $ python3 calibrated_dual_camera.py
't open file 'calibrated_dual_camera.py': [Errno 2] No such file or

i:~/thermal_py $ python3 camera_preview.py
 buffer memory on /dev/video1 -- decreasing buffers
 buffer memory on /dev/video1 -- decreasing buffers
```

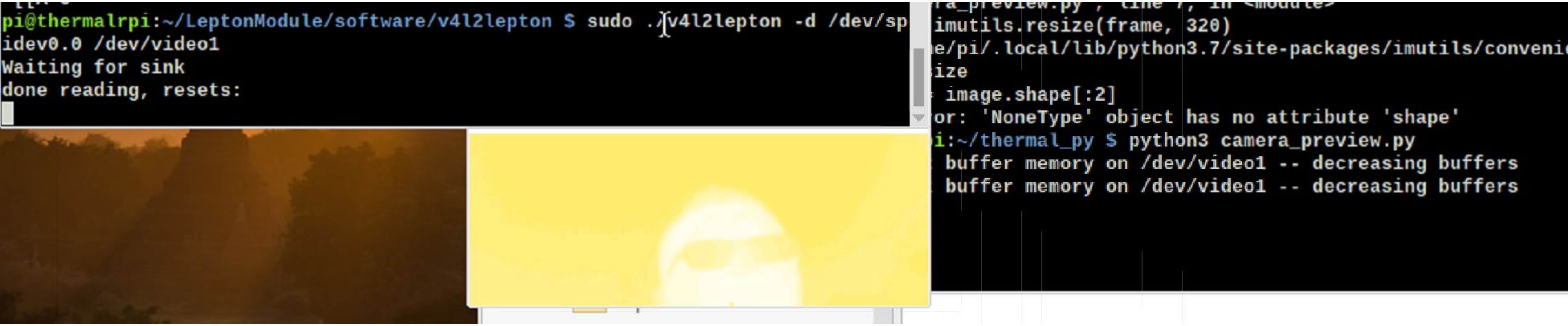
對應 lepton3 的版本

```
$ cd ~
$ git clone https://github.com/acqxi/FLIR.git
```

將裡面的 v4l2lepton.cpp 拖到

/home/pi/LeptonModule/software/v4l2lepton

覆蓋後重新 $ make 與 之前的步驟



```
pi@thermalrpi:~/LeptonModule/software/v4l2lepton $ sudo ./v4l2lepton -d /dev/sp
idev0.0 /dev/video1
Waiting for sink
done reading, resets:
```

```
ra_preview.py", line 7, in <module>
 imutils.resize(frame, 320)
e/pi/.local/lib/python3.7/site-packages/imutils/convenie
ize
  image.shape[:2]
or: 'NoneType' object has no attribute 'shape'
i:~/thermal_py $ python3 camera_preview.py
  buffer memory on /dev/video1 -- decreasing buffers
  buffer memory on /dev/video1 -- decreasing buffers
```

使用Python串接FLIR Lepton

pylepton

- Pure python library for capturing images from the Lepton over SPI

https://github.com/groupgets/pylepton

安裝 pylepton

- `$ cd ~`
- `$ git clone https://github.com/groupgets/pylepton -b lepton3-dev`
- `$ cd ~/pylepton`
- `$ sudo python3 setup.py install`

測試pylepton_capture 範例程式

- `$ cd ~/pylepton`
- `$ ./pylepton_capture output.jpg`
- `$ gpicview output.jpg`

pylepton_capture

```
import numpy as np
import cv2
from pylepton import Lepton

with Lepton() as l:
    a,_ = l.capture()
cv2.normalize(a, a, 0, 65535, cv2.NORM_MINMAX)
np.right_shift(a, 8, a) # fit data into 8 bits
cv2.imwrite("output.jpg", np.uint8(a))
```

raw sensor data

將 12-bit 的輸出正規化到 OpenCV uint16

DEMO
pylepton_capture

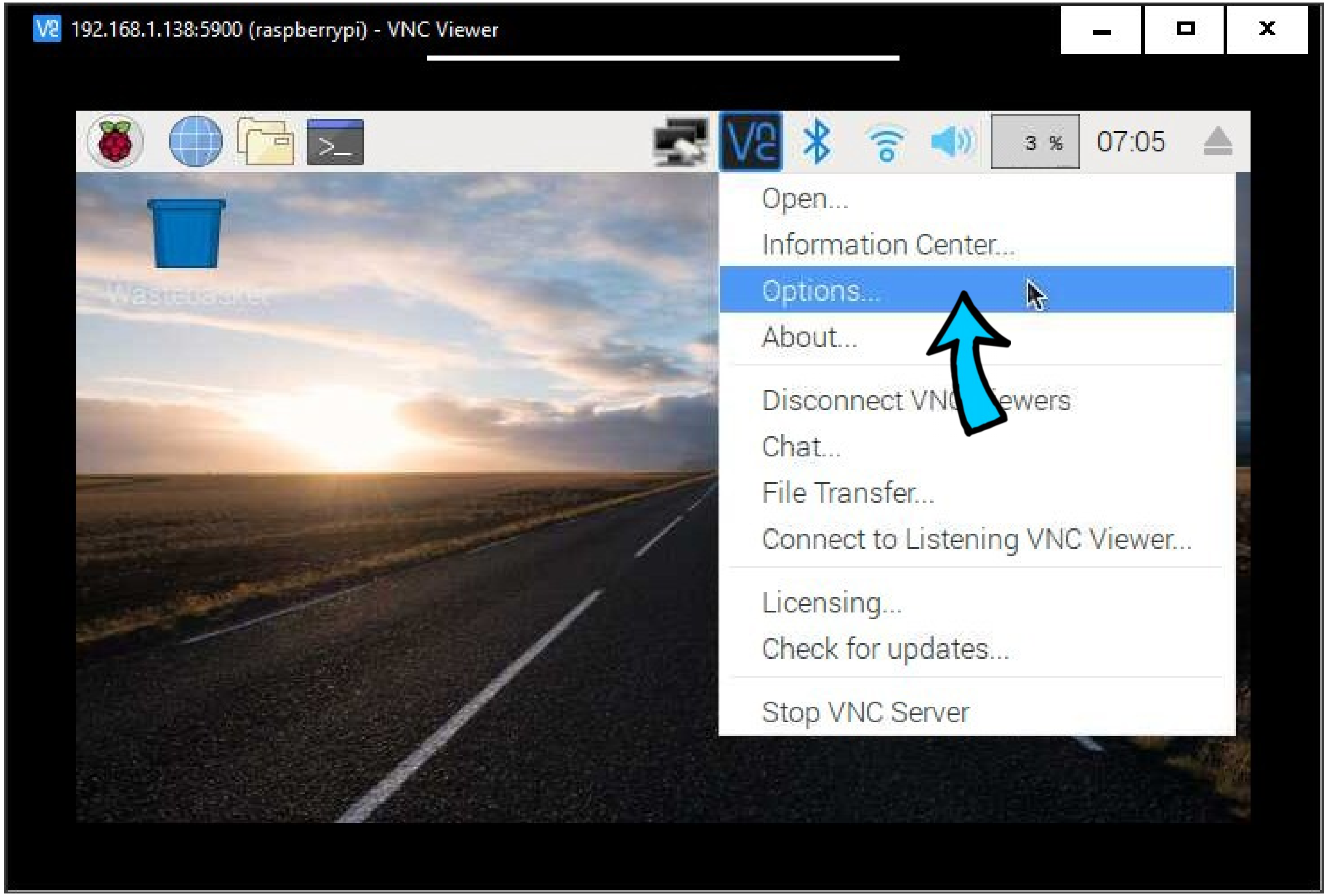
```
$ cd ~/pylepton
$ ./pylepton_capture output.jpg
```

測試 pylepton_overlay 範例程式

- `$ cd ~/pylepton`
- `$ ./pylepton_overlay`

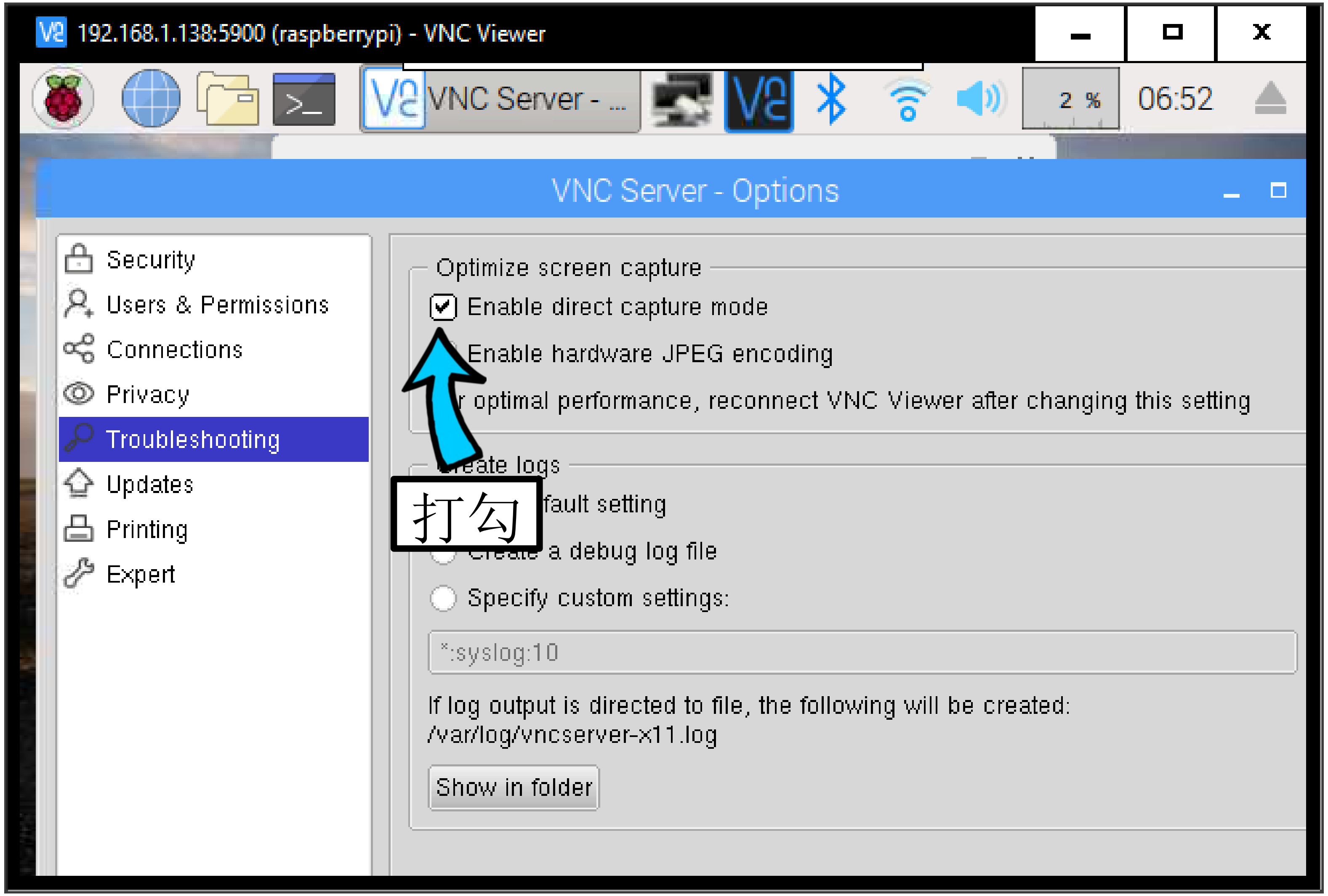
  (需要搭配螢幕或是 VNC Direct Capture Mode 觀察結果)

修改 VNC 設定



192.168.1.138:5900 (raspberrypi) - VNC Viewer

VNC Direct Capture Mode



192.168.1.138:5900 (raspberrypi) - VNC Viewer

VNC Server - Options

Optimize screen capture

☑ Enable direct capture mode

Enable hardware JPEG encoding

r optimal performance, reconnect VNC Viewer after changing this setting

reate logs

fault setting

○ Create a debug log file

○ Specify custom settings:

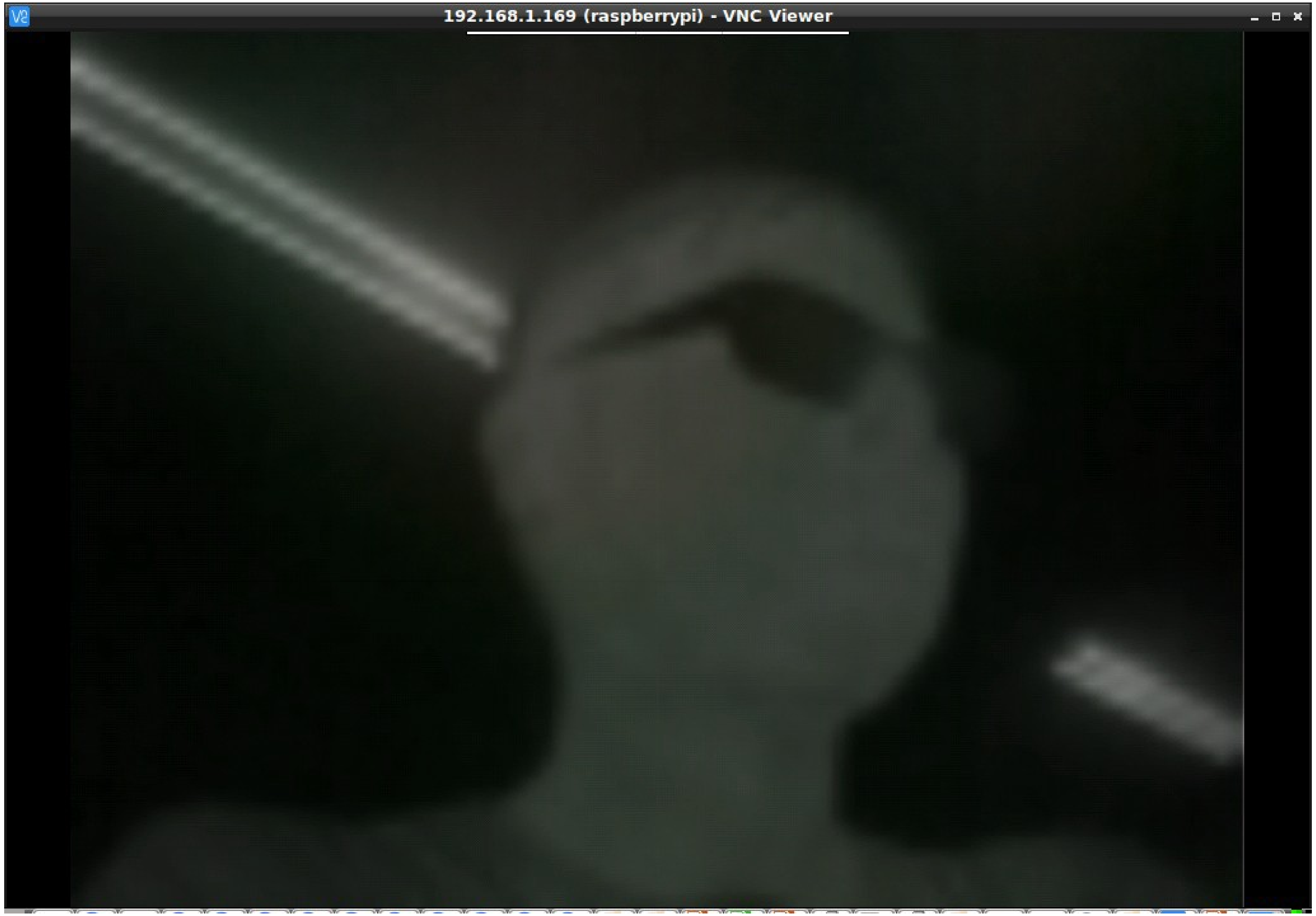
*:syslog:10

If log output is directed to file, the following will be created:
/var/log/vncserver-x11.log

Show in folder

打勾

執行./pylepton_overlay 就可以看到畫面



pylepton_overlay

```
lepton_buf = np.zeros((60, 80, 1), dtype=np.uint16)

with Lepton(device) as l:
    last_nr = 0
    while True:
        _,nr = l.capture(lepton_buf)
        if nr == last_nr:
            # no need to redo this frame
            continue
        last_nr = nr
        cv2.normalize(lepton_buf, lepton_buf, 0, 65535, cv2.NORM_MINMAX)
        np.right_shift(lepton_buf, 8, lepton_buf)
```

將 Lepton 讀到的資料轉換後給 OpenCV

```
from pylepton import Lepton

with Lepton() as l:
    while True:
        a, _ = l.capture()
        cv2.normalize(a, a, 0, 65535, cv2.NORM_MINMAX)
        np.right_shift(a, 8, a)

        cv2.imshow("preview", np.uint8(a))
        if cv2.waitKey(1) & 0xFF == ord("q"):
            break

        time.sleep(0.01)

cv2.destroyAllWindows()
```

DEMO
pylepton_preview.py

```
$ cd ~/FLIR/thermal-pi/01-flir
$ python3 pylepton_preview.py
```

輻射溫度換算

https://www.flir.com/discover/security/radiometric/the-benefits-and-challenges-of-radiometric-thermal-technology/

印出[3][3] 的溫度

```
from pylepton import Lepton

with Lepton() as l:
    while True:
        lepton_buf, nr = l.capture()
        lepton_temp = np.copy(lepton_buf)
        a = np.copy(lepton_buf)
        cv2.normalize(a, a, 0, 65535, cv2.NORM_MINMAX)
        np.right_shift(a, 8, a)
        ...
        t = lepton_temp[3][3]
        print(t, int((t - 27315)/100))

        cv2.imshow("temperature", _lepton_gray)
        time.sleep(0.01)

cv2.destroyAllWindows()
```

轉換 ColorMap

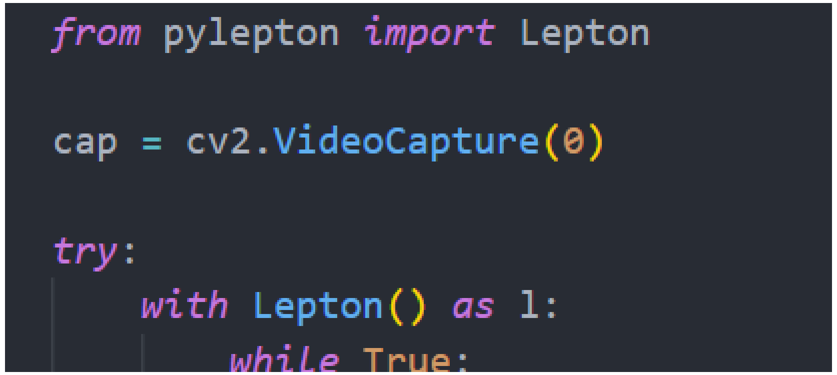
```
_lepton = np.asarray(lepton_buf, np.uint8)
_lepton_gray = cv2.cvtColor(_lepton, cv2.COLOR_GRAY2RGB)
_lepton_gray = cv2.resize(_lepton_gray, (w, h))
_lepton_gray = cv2.applyColorMap(_lepton_gray, cv2.COLORMAP_JET)
#_lepton_gray = cv2.applyColorMap(_lepton_gray, cv2.COLORMAP_RAINBOW)
```

DEMO
pylepton_get_temp.py

```
$ cd ~/FLIR/thermal-pi/01-flir
$ python3 pylepton_get_temp.py
```

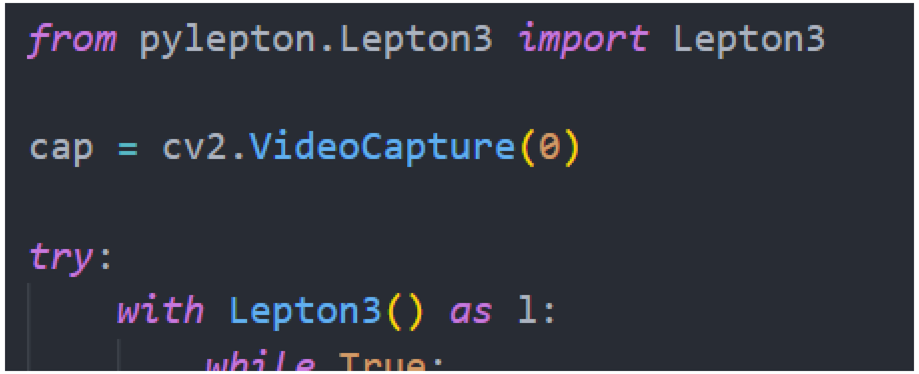
使用新版 pylepton

將程式中的 from pylepton import Lepton 與 with Lepton() as l:



```
from pylepton import Lepton

cap = cv2.VideoCapture(0)

try:
    with Lepton() as l:
        while True:
```

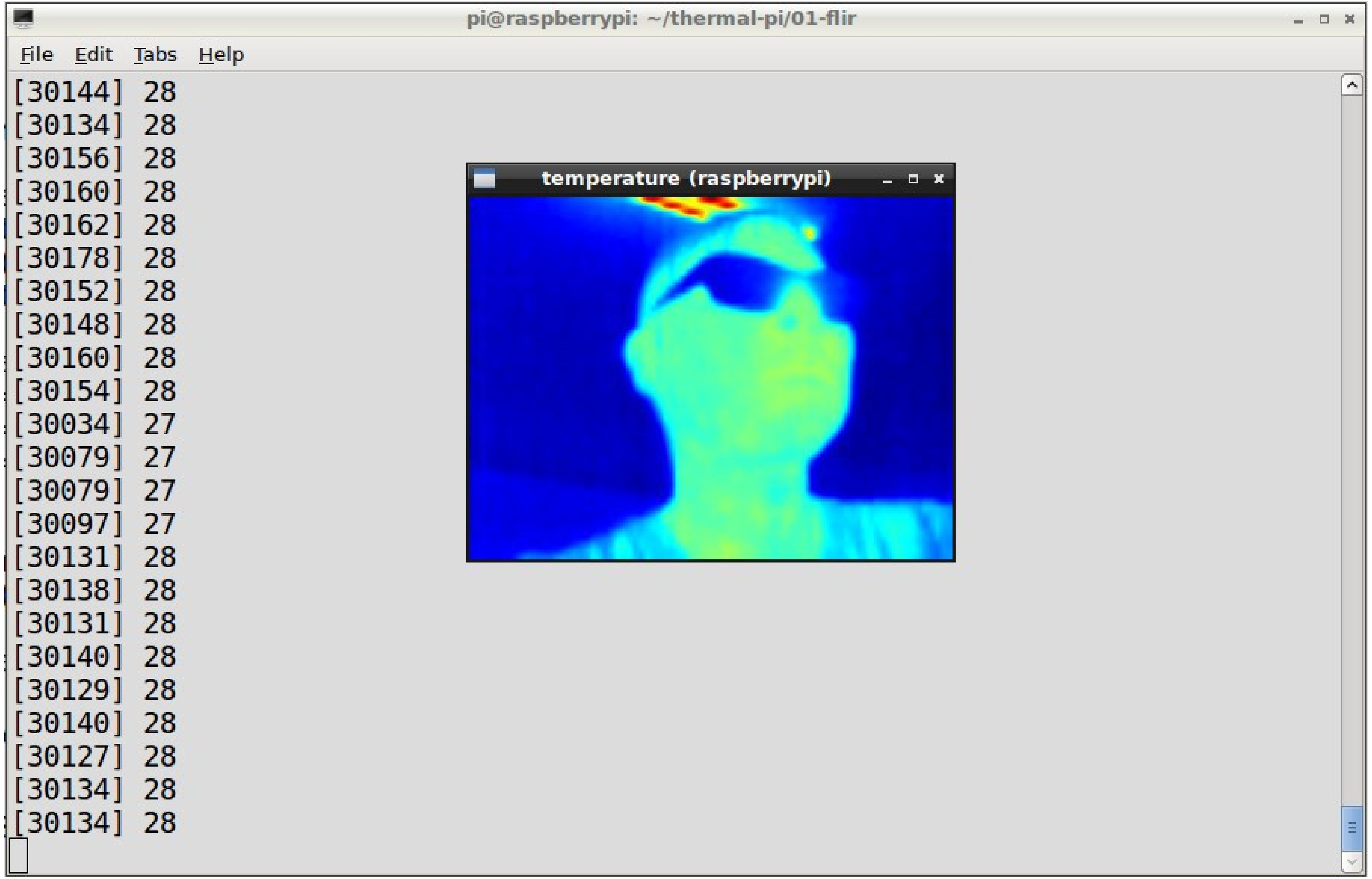
改成 from pylepton.Lepton3 import Lepton3 與 with Lepton3() as l:



```
from pylepton.Lepton3 import Lepton3

cap = cv2.VideoCapture(0)

try:
    with Lepton3() as l:
        while True:
```

之後使用的每份py檔案都需要改，
才能使用lepton3

執行結果

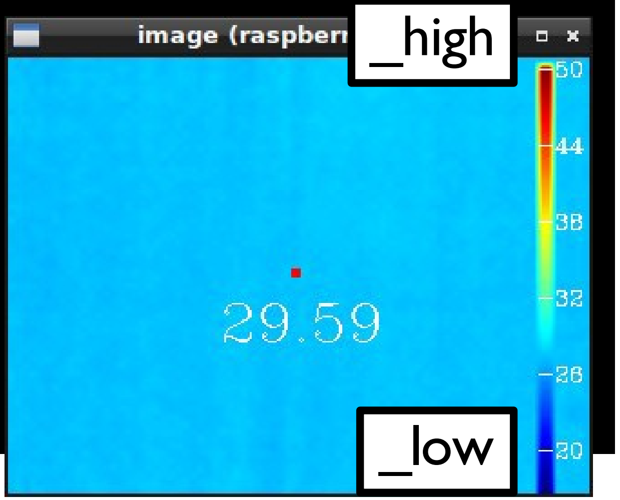


pi@raspberrypi: ~/thermal-pi/01-flir
File Edit Tabs Help
temperature (raspberrypi)

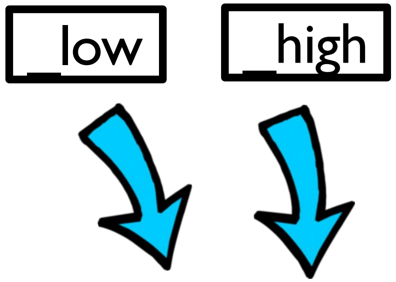
```
[30144] 28
[30134] 28
[30156] 28
[30160] 28
[30162] 28
[30178] 28
[30152] 28
[30148] 28
[30160] 28
[30154] 28
[30034] 27
[30079] 27
[30079] 27
[30097] 27
[30131] 28
[30138] 28
[30131] 28
[30140] 28
[30129] 28
[30140] 28
[30127] 28
[30134] 28
[30134] 28
```

畫出溫度條 (Temperature Bar)

```
def setColorBar(lepton_buf, _low, _high):
    d = (_high - _low) / 60.0
    i = 0

    for i in range(1, 60):
        _d = d * i
        lepton_buf[i][74] = _high - int(_d)
        lepton_buf[i][73] = _high - int(_d)
```



DEMO
pylepton_get_temp.py



_low

_high

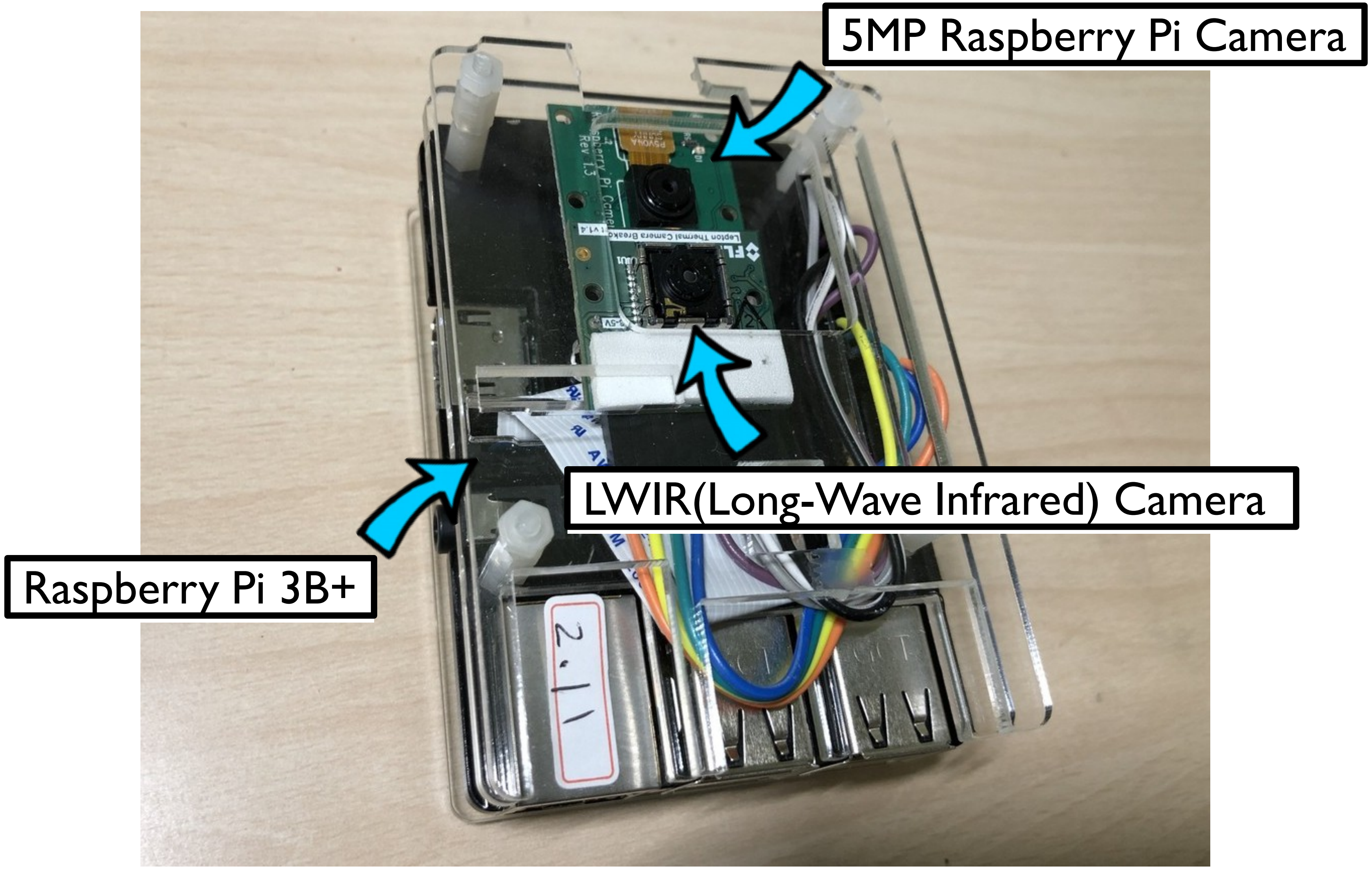
```
$ cd ~/FLIR/thermal-pi/01-flir
$ python3 pylepton_temp_colorbar.py 20 60
```

Camera 安裝

再次執行 camera_preview.py

```
cap = cv2.VideoCapture(0)

while True:
    ret, frame = cap.read()
    frame = imutils.resize(frame, 320)
    cv2.imshow("preview", frame)
    if cv2.waitKey(1) & 0xFF == ord("q"):
        break

cap.release()
cv2.destroyAllWindows()
```

雙相機雛型系統

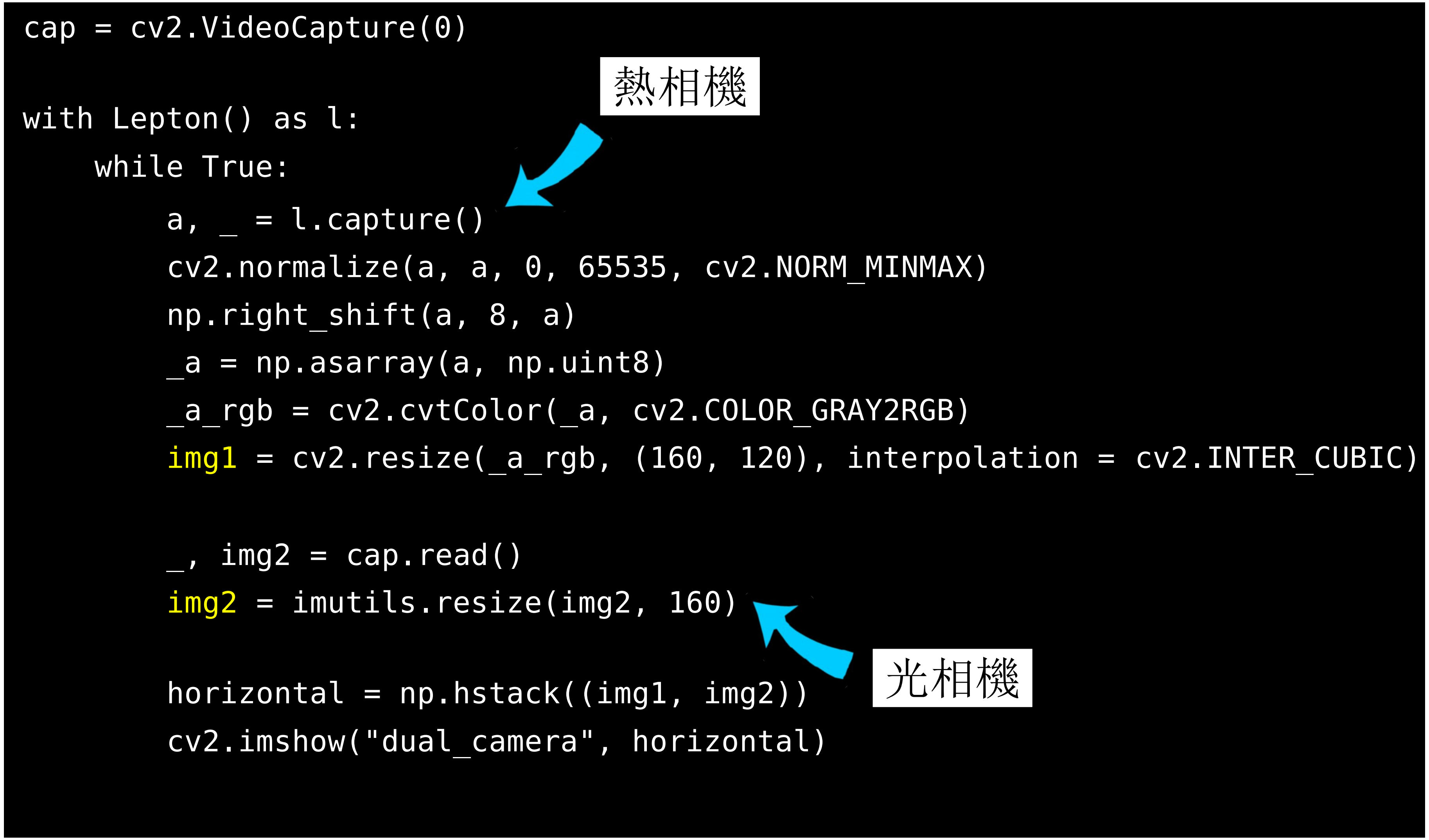


5MP Raspberry Pi Camera

LWIR(Long-Wave Infrared) Camera

Raspberry Pi 3B+

雙相機顯示



```
cap = cv2.VideoCapture(0)

with Lepton() as l:
    while True:
        a, _ = l.capture()
        cv2.normalize(a, a, 0, 65535, cv2.NORM_MINMAX)
        np.right_shift(a, 8, a)
        _a = np.asarray(a, np.uint8)
        _a_rgb = cv2.cvtColor(_a, cv2.COLOR_GRAY2RGB)
        img1 = cv2.resize(_a_rgb, (160, 120), interpolation = cv2.INTER_CUBIC)

        _, img2 = cap.read()
        img2 = imutils.resize(img2, 160)

        horizontal = np.hstack((img1, img2))
        cv2.imshow("dual_camera", horizontal)
```

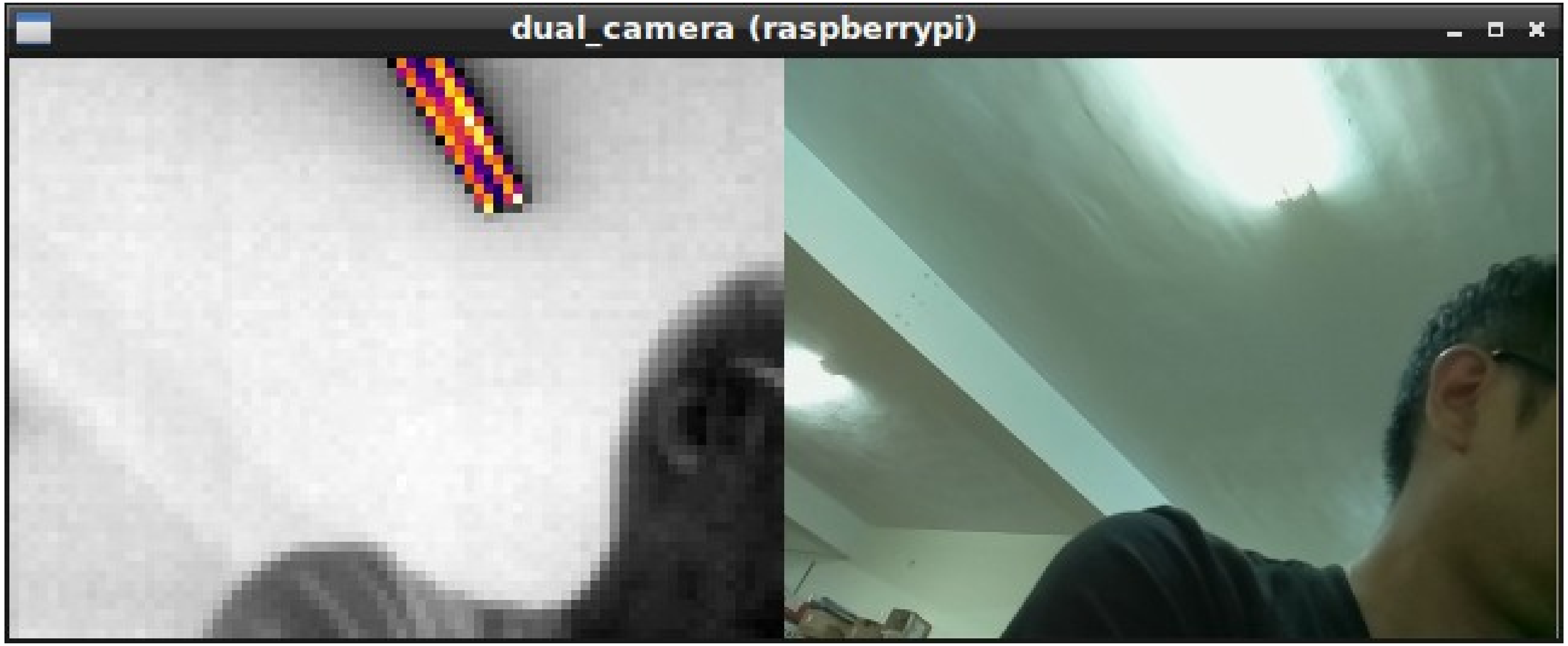
熱相機

光相機

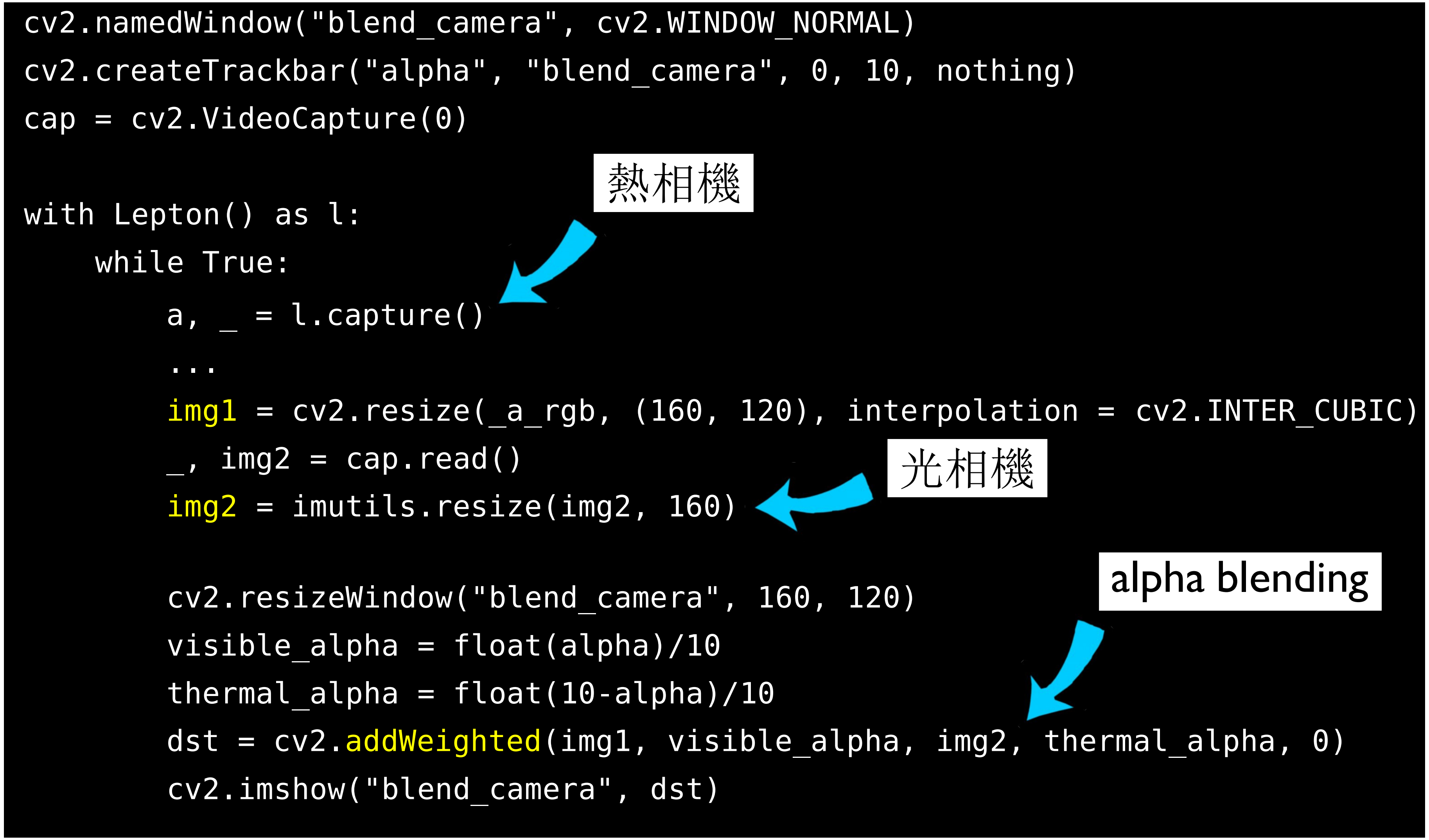
DEMO
dual_camera.py

```
$ cd ~/FLIR/thermal-pi/02-
calibration
$ python3 dual_camera.py
```

將雙相機並排顯示



雙相機 alpha blending 混合



```
cv2.namedWindow("blend_camera", cv2.WINDOW_NORMAL)
cv2.createTrackbar("alpha", "blend_camera", 0, 10, nothing)
cap = cv2.VideoCapture(0)

with Lepton() as l:
    while True:
        a, _ = l.capture()
        ...
        img1 = cv2.resize(_a_rgb, (160, 120), interpolation = cv2.INTER_CUBIC)
        _, img2 = cap.read()
        img2 = imutils.resize(img2, 160)

        cv2.resizeWindow("blend_camera", 160, 120)
        visible_alpha = float(alpha)/10
        thermal_alpha = float(10-alpha)/10
        dst = cv2.addWeighted(img1, visible_alpha, img2, thermal_alpha, 0)
        cv2.imshow("blend_camera", dst)
```

熱相機

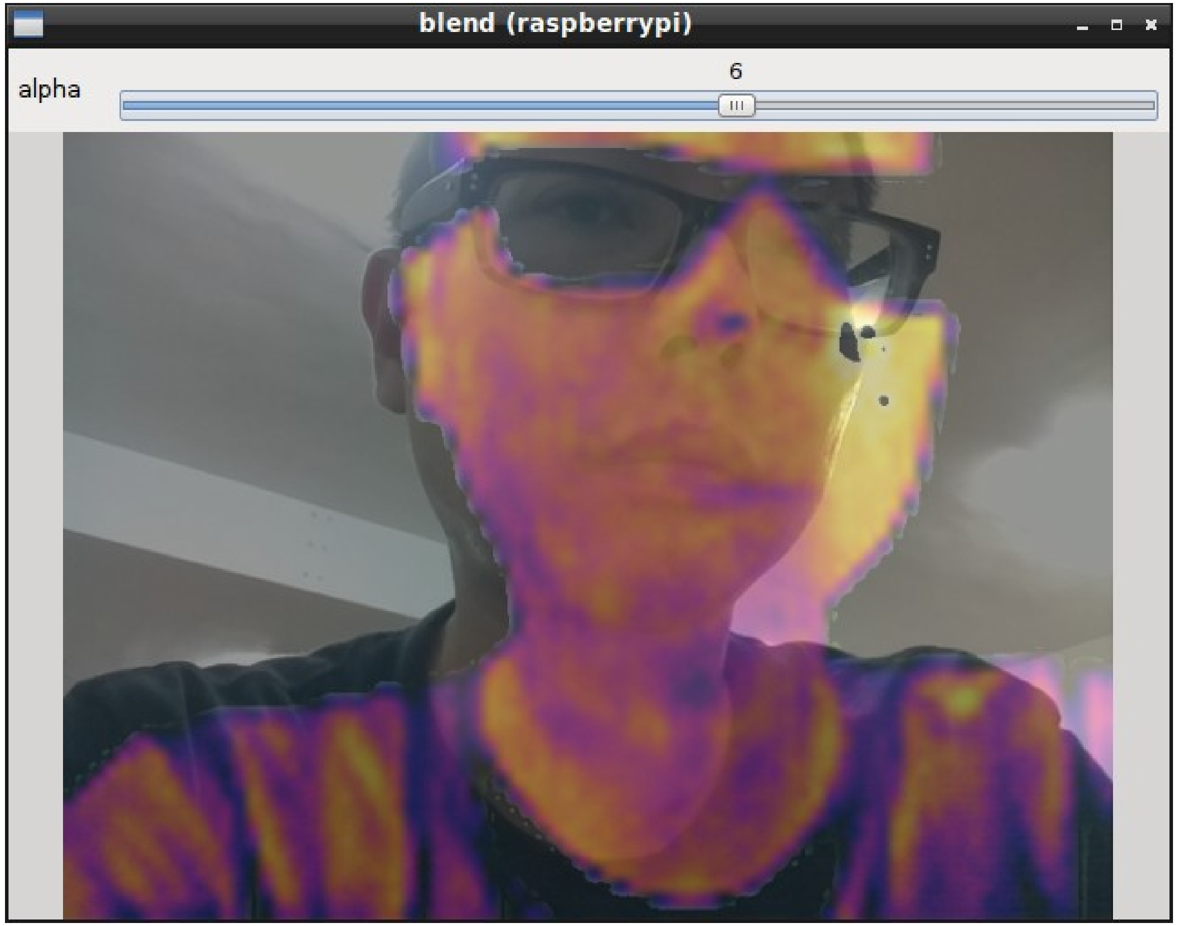
光相機

alpha blending

DEMO
blend_camera.py

```
$ cd ~/FLIR/thermal-pi/02-calibration
$ python3 blend_camera.py
```

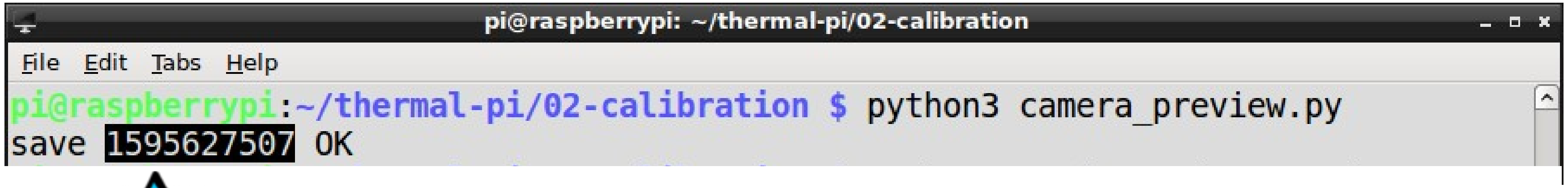
調整雙相機的 alpha 值



雙相機校正

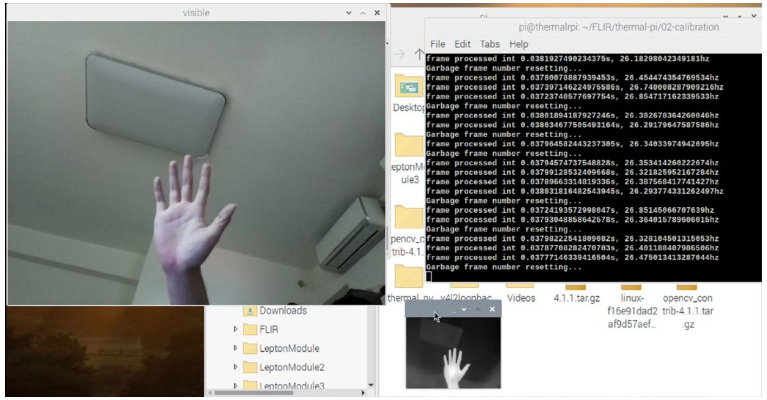
雙相機校正需要先拍照

- $ python3 camera_preview.py



```
pi@raspberrypi:~/thermal-pi/02-calibration $ python3 camera_preview.py
save 1595627507 OK
```

按 c 後自動產生的檔名



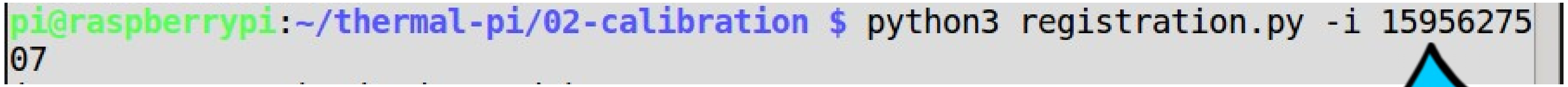
```
frame processed int 0.0381927490234375s, 26.18298042349181hz
Garbage frame number resetting...
frame processed int 0.03780078887939453s, 26.454474354769534hz
frame processed int 0.037397146224975586s, 26.740008287909216hz
frame processed int 0.03723740577697754s, 26.854717162339533hz
Garbage frame number resetting...
frame processed int 0.03801894187927246s, 26.302678364260046hz
frame processed int 0.038034677505493164s, 26.29179647587586hz
Garbage frame number resetting...
frame processed int 0.037964582443237305s, 26.34033974942695hz
Garbage frame number resetting...
frame processed int 0.03794574737548828s, 26.353414260222674hz
frame processed int 0.03799128532409668s, 26.321825952167284hz
frame processed int 0.03789663314819336s, 26.387568417741427hz
frame processed int 0.038031816482543945s, 26.293774331262497hz
Garbage frame number resetting...
frame processed int 0.03724193572998047s, 26.85145066707639hz
frame processed int 0.03793048858642578s, 26.364015789606015hz
Garbage frame number resetting...
frame processed int 0.03798222541809082s, 26.328104501315053hz
frame processed int 0.03787708282470703s, 26.401188407986506hz
frame processed int 0.03777146339416504s, 26.475013413287044hz
Garbage frame number resetting...
```

先點選視窗，
按 'c' 拍照
按 'q' 離開

校正重點：
光影像背景要單純
熱影像物體要完整

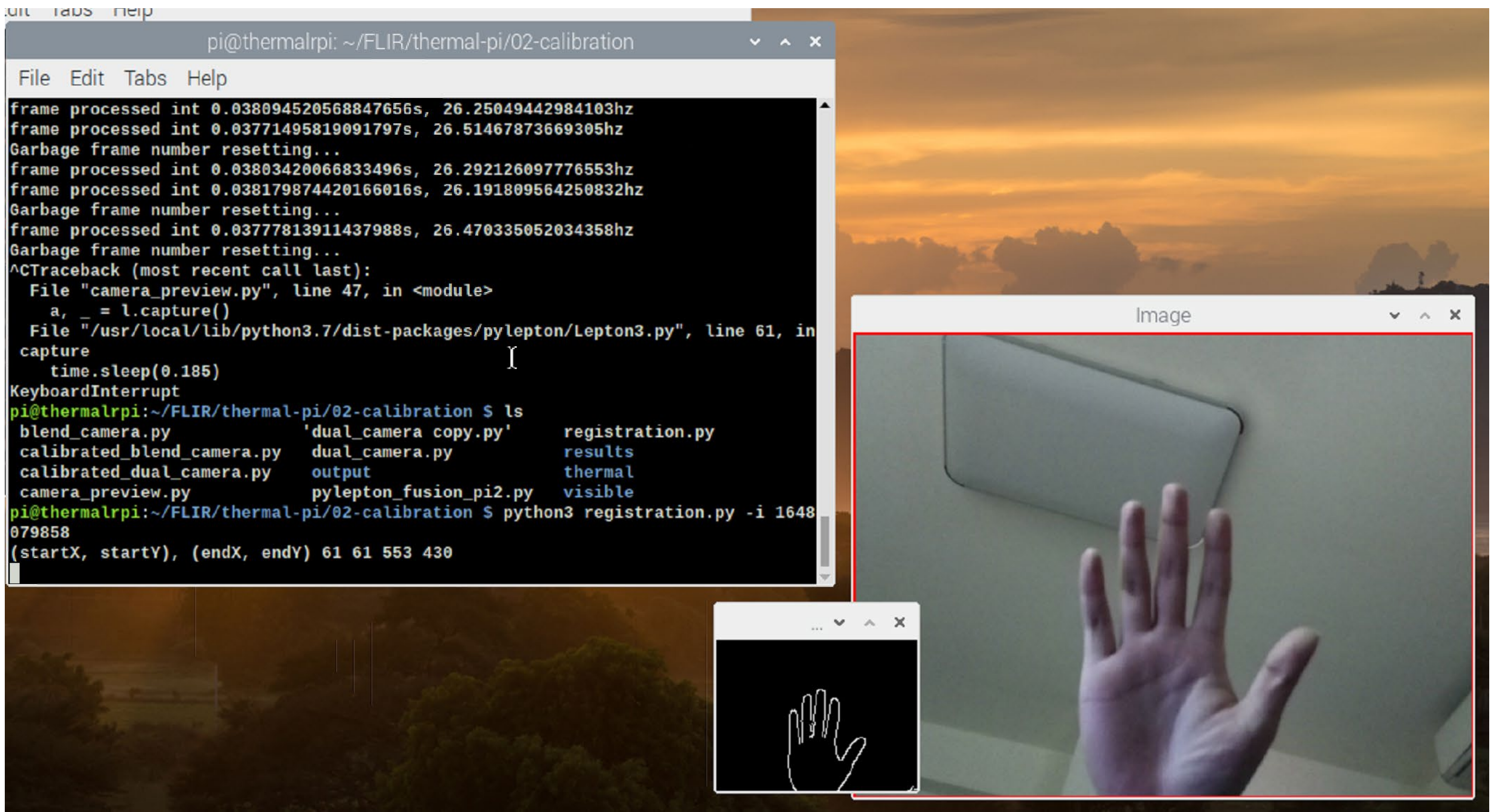
拍照後根據產生檔名做校正

- `$ python3 registration.py -i [FILE]`



```
pi@raspberrypi:~/thermal-pi/02-calibration $ python3 registration.py -i 15956275
07
```

自動產生的檔名

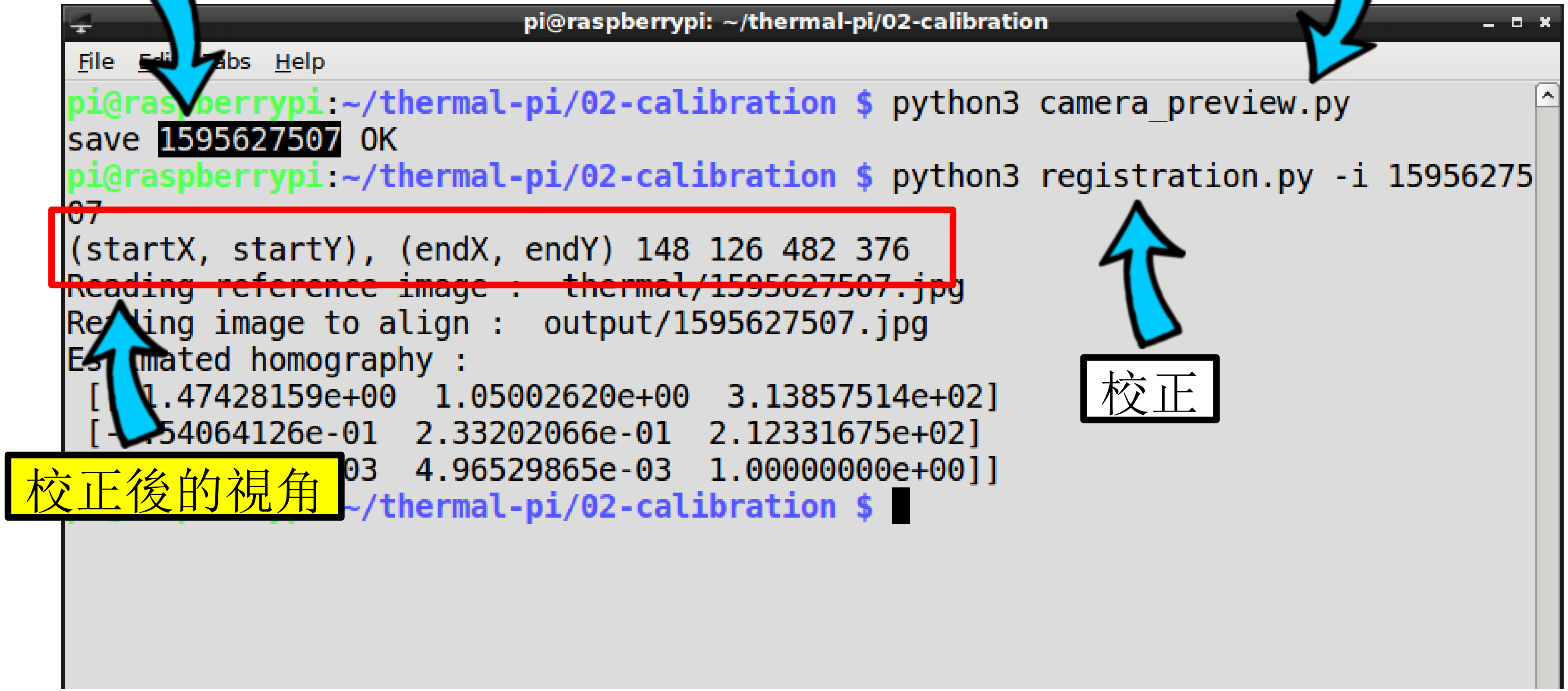


```
frame processed int 0.038094520568847656s, 26.25049442984103hz
frame processed int 0.03771495819091797s, 26.51467873669305hz
Garbage frame number resetting...
frame processed int 0.03803420066833496s, 26.292126097776553hz
frame processed int 0.038179874420166016s, 26.191809564250832hz
Garbage frame number resetting...
frame processed int 0.03777813911437988s, 26.470335052034358hz
Garbage frame number resetting...
^CTraceback (most recent call last):
  File "camera_preview.py", line 47, in <module>
    a, _ = l.capture()
  File "/usr/local/lib/python3.7/dist-packages/pylepton/Lepton3.py", line 61, in
 capture
    time.sleep(0.185)
KeyboardInterrupt
pi@thermalrpi:~/FLIR/thermal-pi/02-calibration $ ls
 blend_camera.py             'dual_camera copy.py'     registration.py
 calibrated_blend_camera.py   dual_camera.py            results
 calibrated_dual_camera.py    output                    thermal
 camera_preview.py            pylepton_fusion_pi2.py    visible
pi@thermalrpi:~/FLIR/thermal-pi/02-calibration $ python3 registration.py -i 1648
079858
(startX, startY), (endX, endY) 61 61 553 430
```

按任意鍵關閉

執行結果

- `$ python3 registration.py -i [FILE]`



```
pi@raspberrypi:~/thermal-pi/02-calibration $ python3 camera_preview.py
save 1595627507 OK
pi@raspberrypi:~/thermal-pi/02-calibration $ python3 registration.py -i 15956275
07
(startX, startY), (endX, endY) 148 126 482 376
Reading reference image :  thermal/1595627507.jpg
Reading image to align :  output/1595627507.jpg
Estimated homography :
 [[ 1.47428159e+00  1.05002620e+00  3.13857514e+02]
 [-[illegible].54064126e-01  2.33202066e-01  2.12331675e+02]
 [[illegible]03  4.96529865e-03  1.00000000e+00]]
pi@raspberrypi:~/thermal-pi/02-calibration $
```

自動產生的檔名

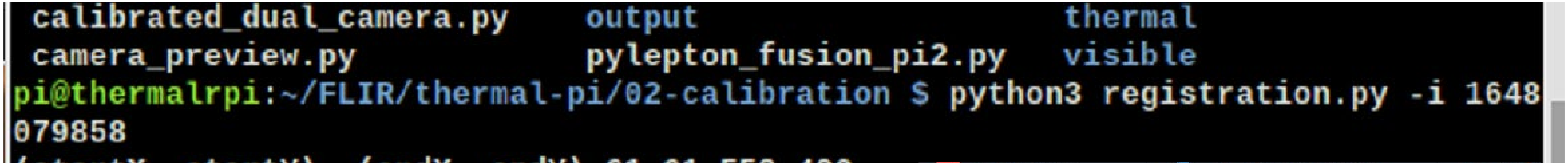
拍照

校正

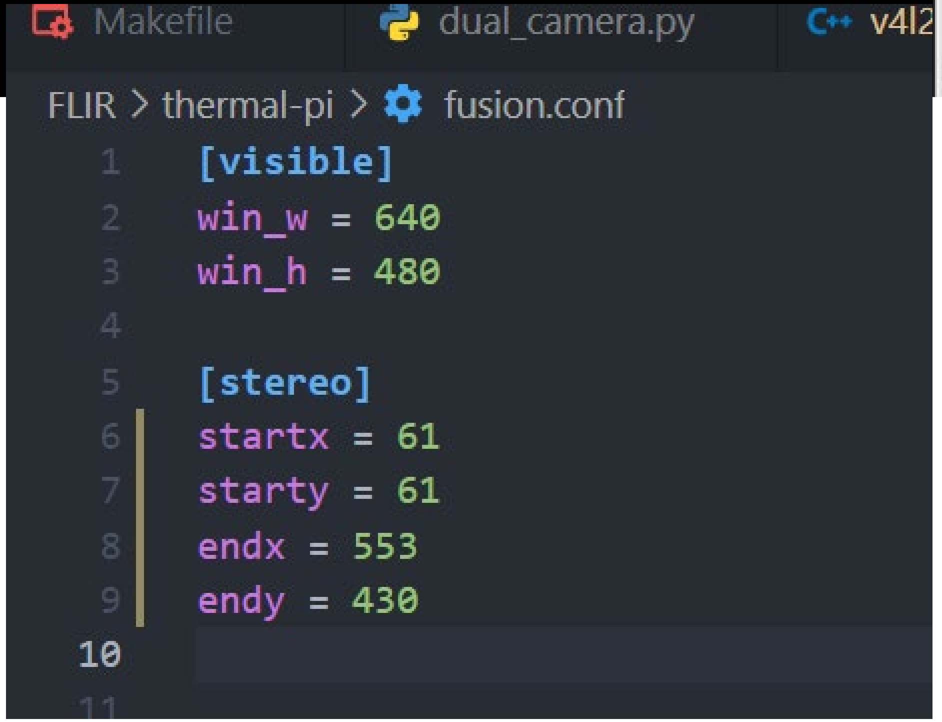
校正後的視角

假如沒有自動修正，需手動

- 將剛才的執行結果修正到 ../fusion.conf



```
 calibrated_dual_camera.py    output                  thermal
 camera_preview.py            pylepton_fusion_pi2.py  visible
pi@thermalrpi:~/FLIR/thermal-pi/02-calibration $ python3 registration.py -i 1648
079858
(startX, startY), (endX, endY) 61 61 553 430
```

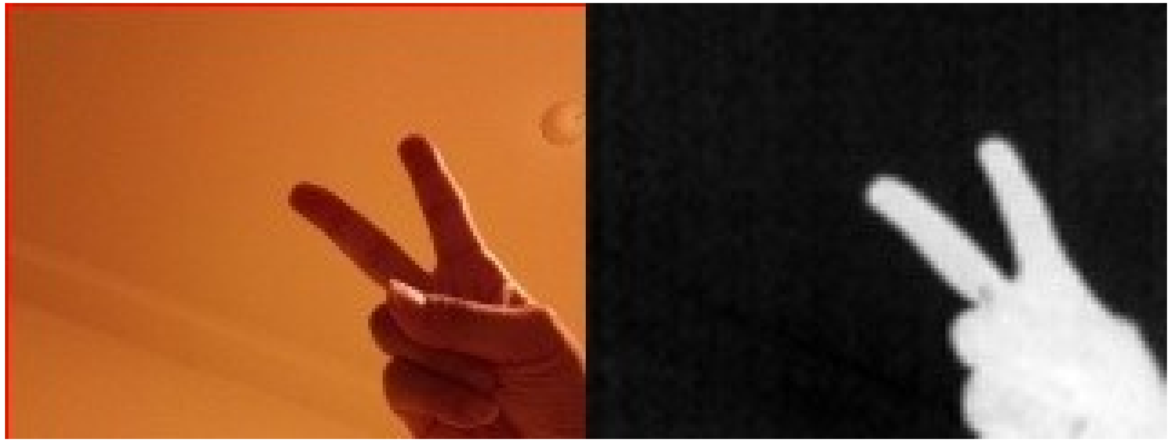


```
[visible]
win_w = 640
win_h = 480

[stereo]
startx = 61
starty = 61
endx = 553
endy = 430
```

檢查校正結果

```
$ gpicview results/1595627507.jpg
```

每個人檔名不同



自動讀取校正後的視角

讀取 fusion.conf 的視角

```
config = configparser.ConfigParser()
config.read('../fusion.conf')
startX = int(config.get('stereo', 'startX'))

with Lepton() as l:
    while True:
        a, _ = l.capture()
        cv2.normalize(a, a, 0, 65535, cv2.NORM_MINMAX)
        np.right_shift(a, 8, a)
        _a = np.asarray(a, np.uint8)
        _a_rgb = cv2.cvtColor(_a, cv2.COLOR_GRAY2RGB)
        img1 = cv2.resize(_a_rgb, (160, 120), interpolation = cv2.INTER_CUBIC)

        _, img2 = cap.read()
        img2 = imutils.resize(img2, 640)
        crop_img2 = img2[startY:endY, startX:endX]
        crop_img2 = cv2.resize(crop_img2, (160, 120))
```

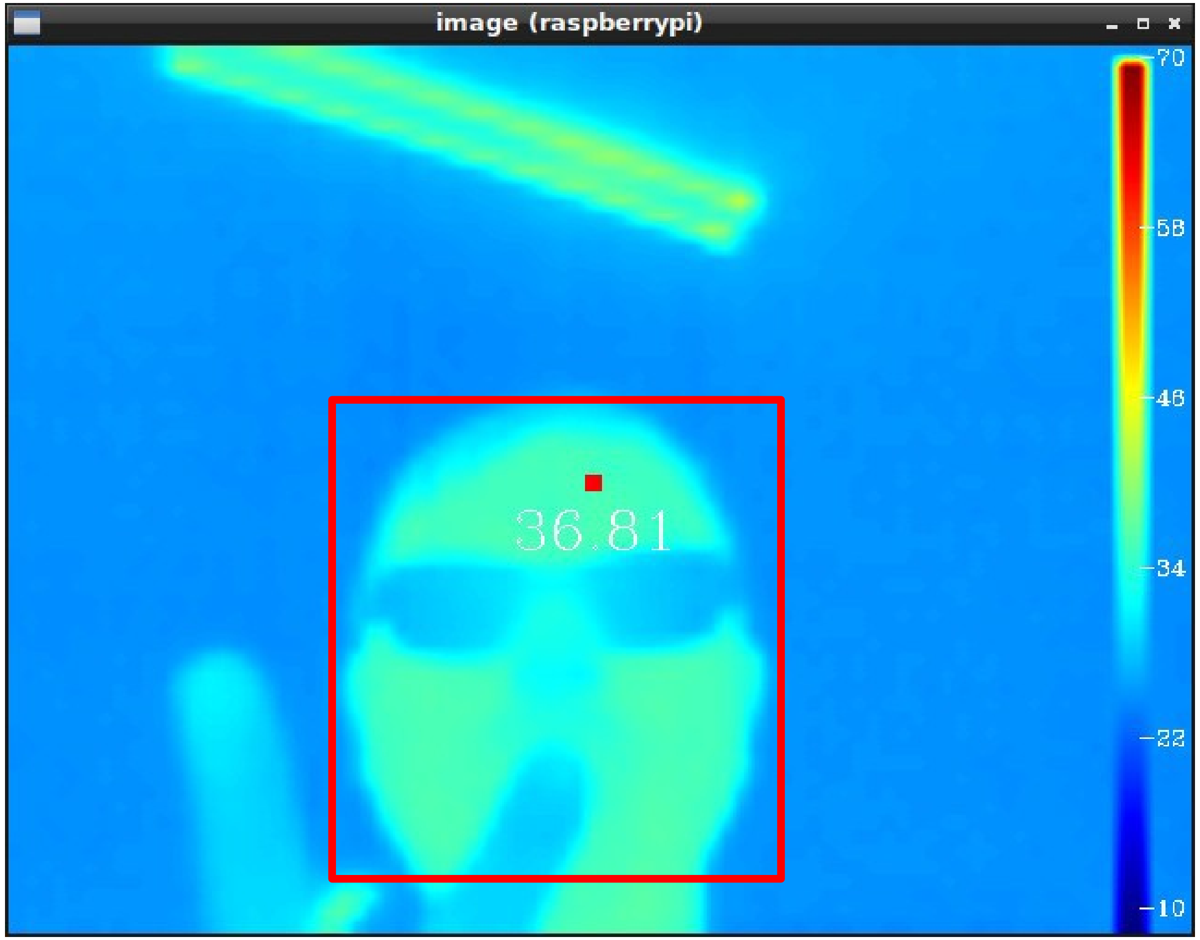
DEMO
calibrated_dual_camera.py

```
$ cd ~/FLIR/thermal-pi/02-calibration
$ python3 calibrated_dual_camera.py
```

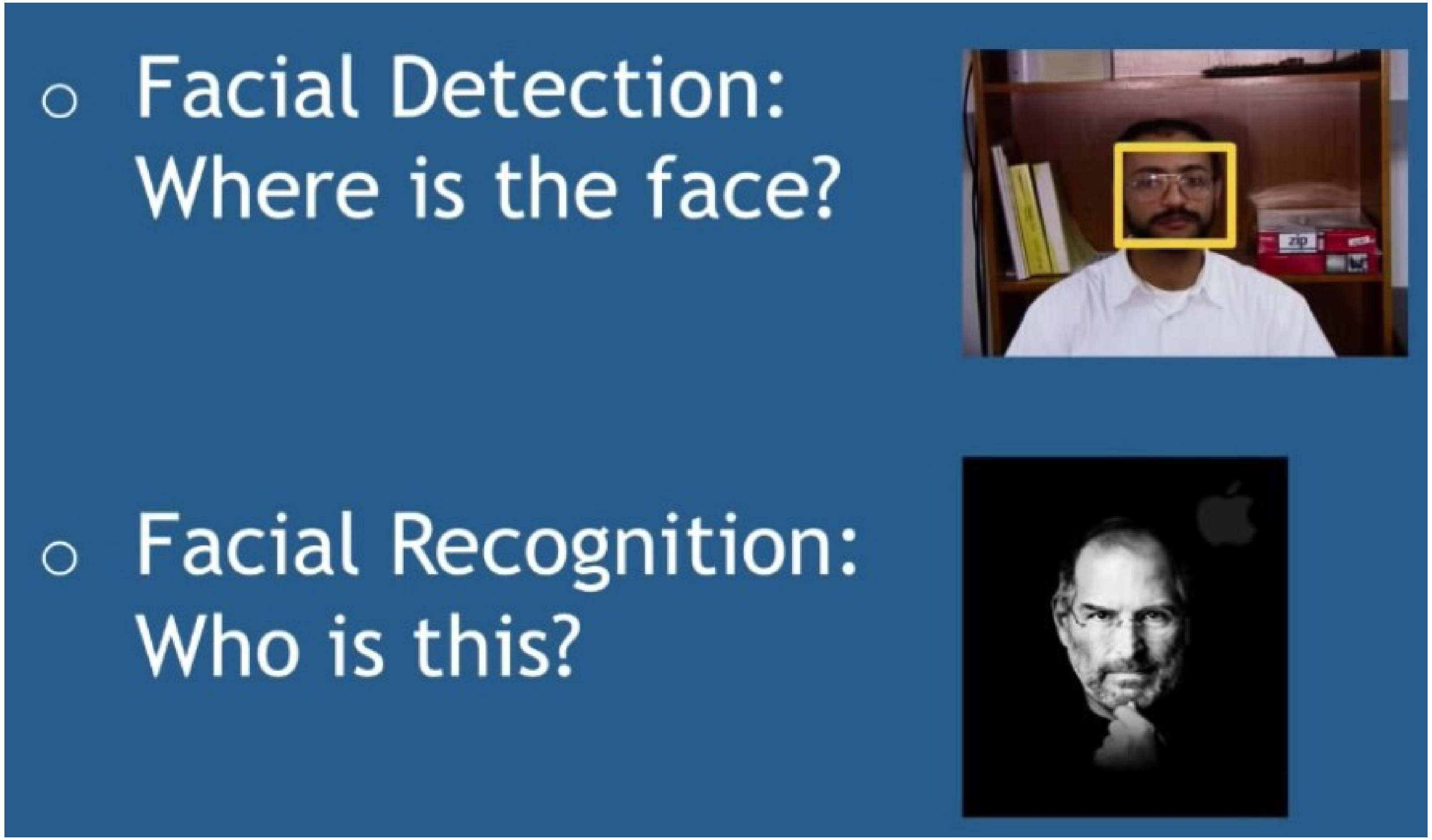
DEMO
calibrated_blend_camera.py

```
$ cd ~/FLIR/thermal-pi/02-calibration
$ python3 calibrated_blend_camera.py
```

先找出人臉，再計算出人臉最高溫度



人臉偵測與人臉識別

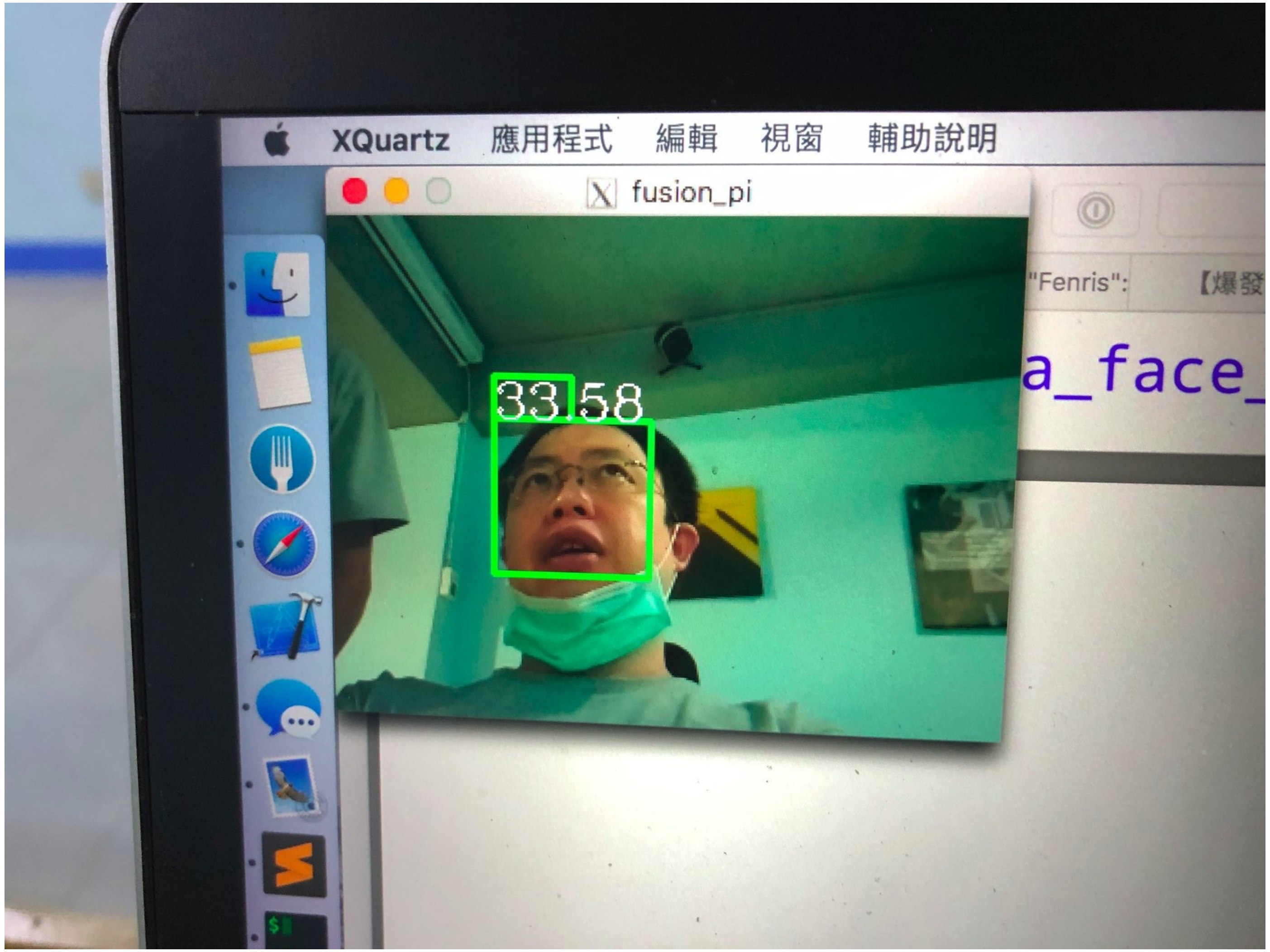


- Facial Detection: Where is the face?
- Facial Recognition: Who is this?

https://www.youtube.com/watch?v=sWTvK72-SPU

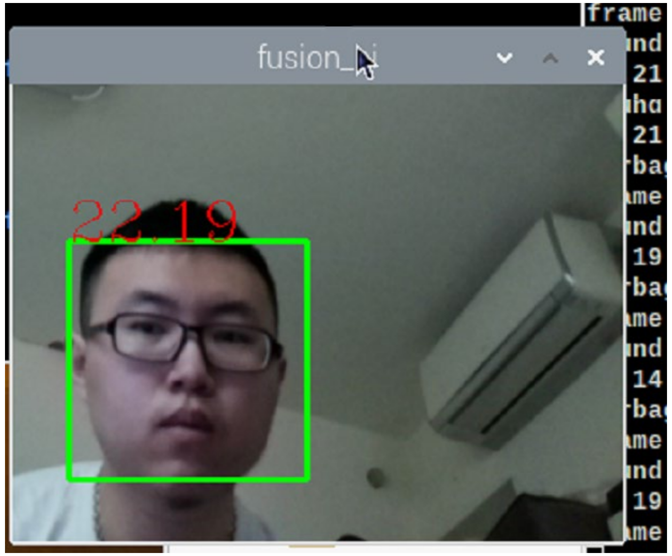
自製雙相機熱像儀

- 先找出人臉，再算出人臉平均溫度 (或最大溫度)



DEMO
pylepton_fusion_pi.py

```
$ cd ~/FLIR/thermal-pi/03-fusion
$ python3 pylepton_fusion_pi.py
```


fusion_pi
22.19
frame
nd
21
ha
21
ba
me
nd
19
ba
me
nd
14
ba
me
nd
19
me