Progress report

進度報告

林晉德

2021年7月5日

To do list

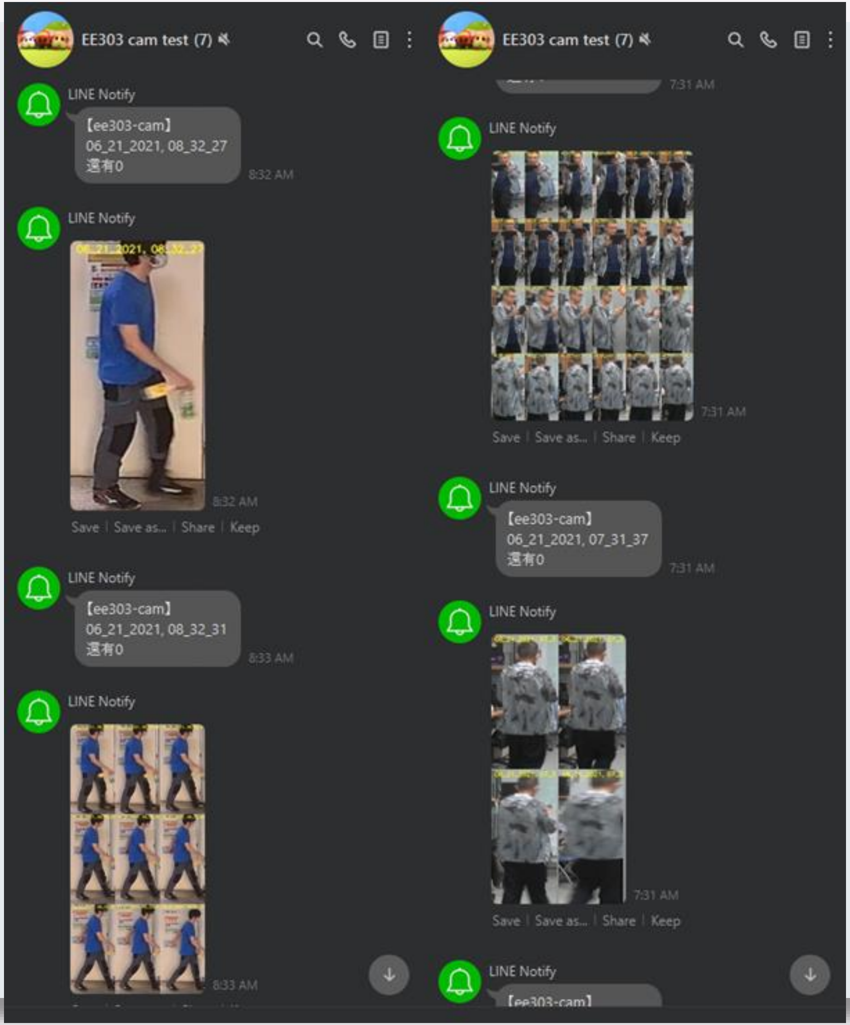
Person Re-Identification system		
1. Person detection	Object detection (MobileNetSSD) Openpose	done cont.
2. 特徵擷取網路架構	Torchreid (A Library for Deep Learning Person Re-Identification in Pytorch) backbone: ResNet50 backbone: OSNet Tensorflow -> Pytorch 世超-DSPF (backbone: SE-ResNeXt)	 done cont. cont.
3. 動態每天分群給編號 4. 使用3的分群編號，去做Query識別	Feature Extractor Compute distance matrix Real-time system	done done cont.
5. 確認編號的確實身份	結合Line bod應用	cont.

What I have done

Person detection

Object detection (MobileNetSSD)	done
Openpose	cont.

1	aeroplane
2	bicycle
3	bird
4	boat
5	bottle
6	bus
7	car
8	cat
9	chair
10	cow
11	dining table
12	dog
13	horse
14	motorbike
15	**person**
16	pottedplant
17	sheep
18	sofa
19	train
20	TV monitor



EE303 cam test (7)

LINE Notify

【ee303-cam】
06_21_2021, 08_32_27
還有0

8:32 AM

LINE Notify

8:32 AM

LINE Notify

【ee303-cam】
06_21_2021, 08_32_31
還有0

8:33 AM

LINE Notify

8:33 AM

EE303 cam test (7)

7:31 AM

LINE Notify

7:31 AM

LINE Notify

【ee303-cam】
06_21_2021, 07_31_37
還有0

7:31 AM

LINE Notify

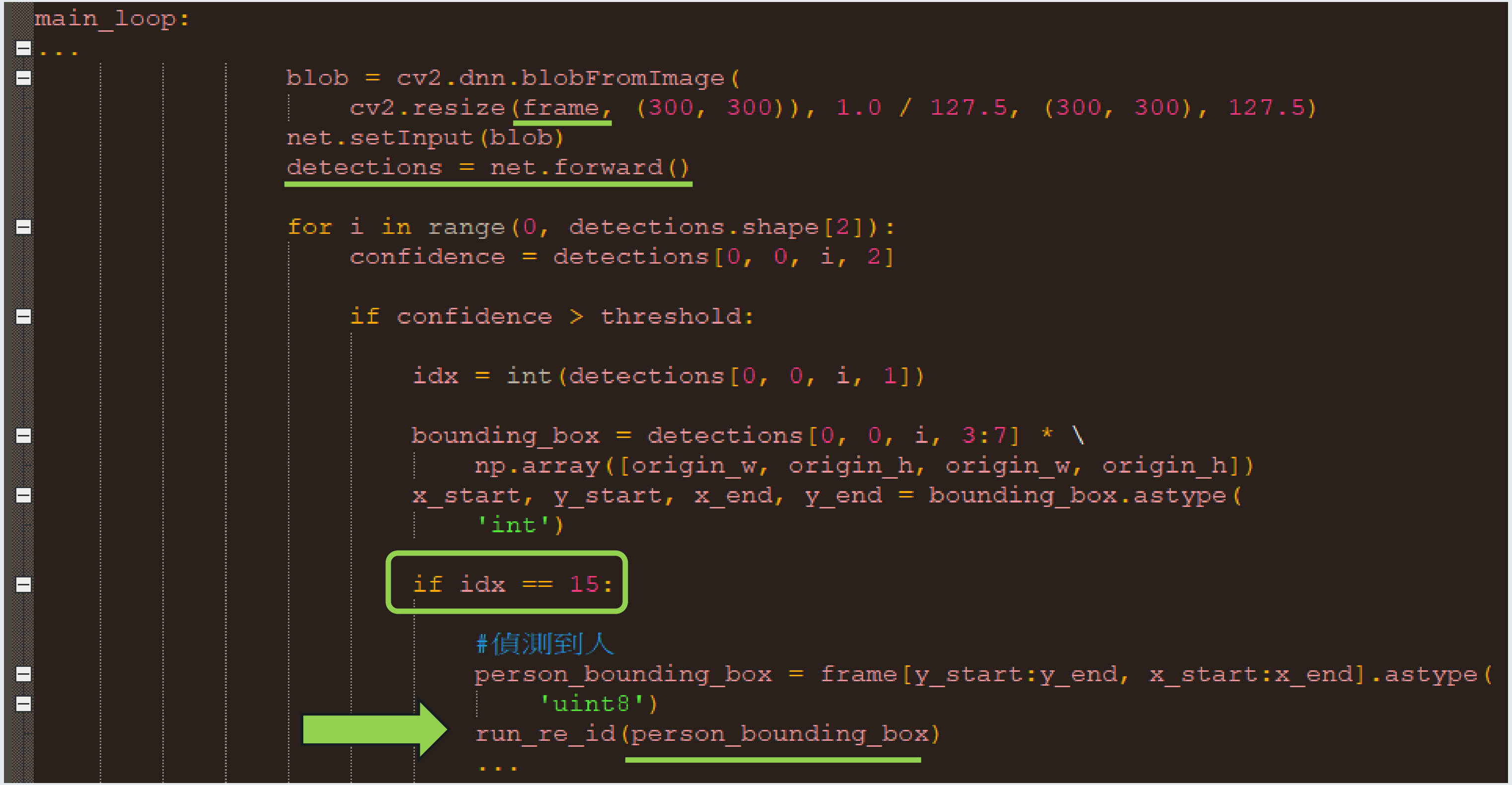
7:31 AM

LINE Notify

【ee303-cam】

Save | Save as... | Share | Keep
Save | Save as... | Share | Keep
Save | Save as... | Share | Keep

Use person bounding box



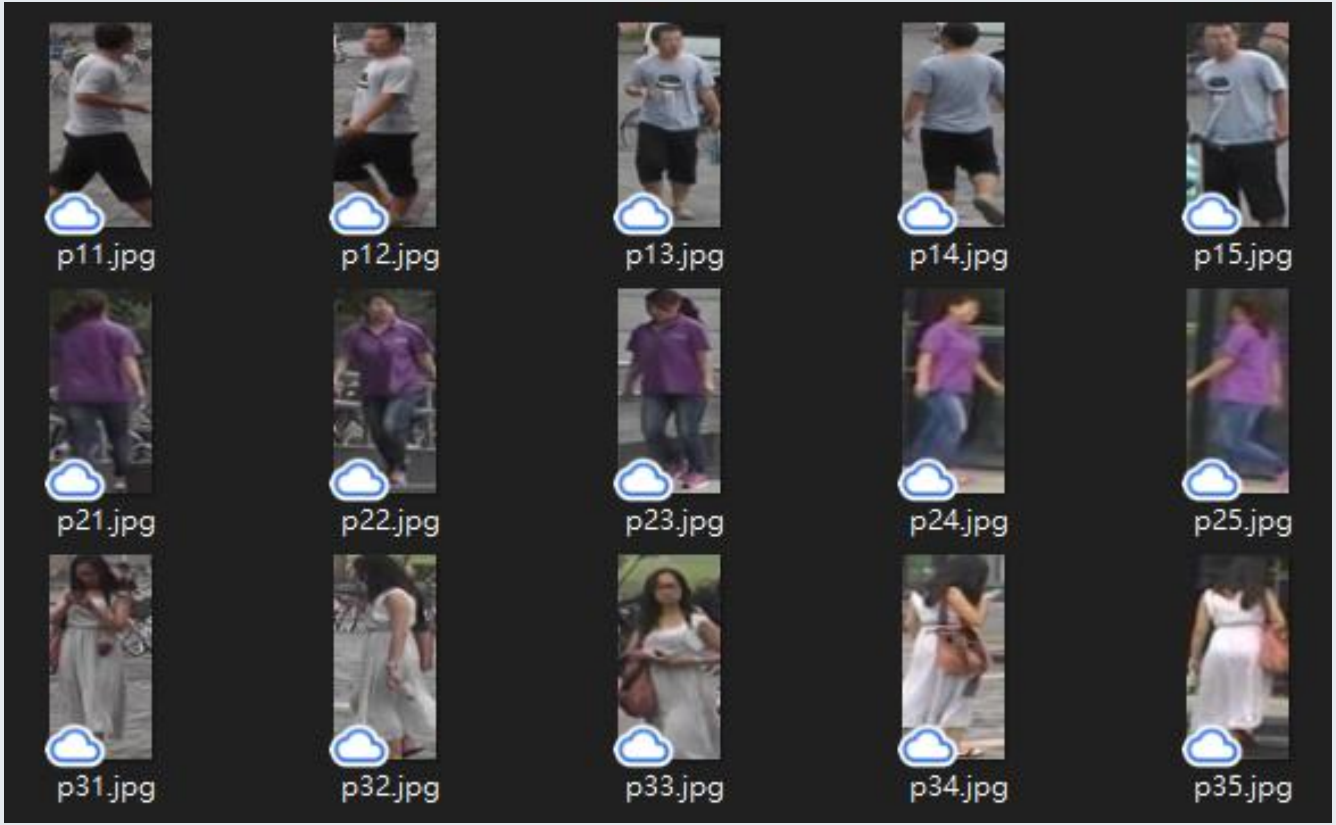
```
main_loop:
...
        blob = cv2.dnn.blobFromImage(
          cv2.resize(frame, (300, 300)), 1.0 / 127.5, (300, 300), 127.5)
        net.setInput(blob)
        detections = net.forward()

        for i in range(0, detections.shape[2]):
          confidence = detections[0, 0, i, 2]

          if confidence > threshold:

            idx = int(detections[0, 0, i, 1])

            bounding_box = detections[0, 0, i, 3:7] * \
              np.array([origin_w, origin_h, origin_w, origin_h])
            x_start, y_start, x_end, y_end = bounding_box.astype(
              'int')

            if idx == 15:

              #偵測到人
              person_bounding_box = frame[y_start:y_end, x_start:x_end].astype(
                'uint8')
              run_re_id(person_bounding_box)
              ...
```

Torchreid training ResNet50

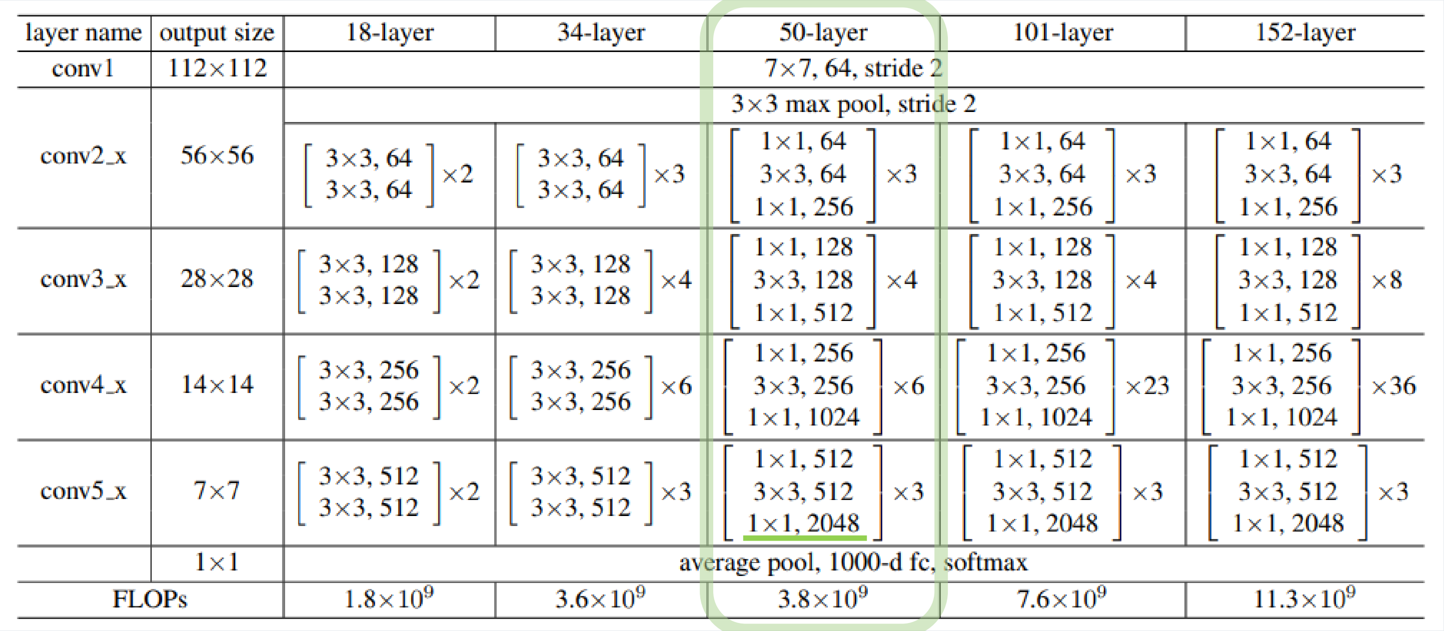
Hyperparamters	
Data set	Market1501
batch_size_train	32
model name	resnet50
optimizer	adam
lr	0.0003
lr_scheduler	single_step
stepsize	20
max_epoch	60

Results	
mAP	67.40%
Rank-1	85.00%
Elapsed	01:15:48

Test image


p11.jpg
p12.jpg
p13.jpg
p14.jpg
p15.jpg
p21.jpg
p22.jpg
p23.jpg
p24.jpg
p25.jpg
p31.jpg
p32.jpg
p33.jpg
p34.jpg
p35.jpg

Torchreid ResNet50 Feature Extractor



layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times23$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Input image size (256, 128)

ResNet50

Output feature (1, 2048)

Feature Extractor

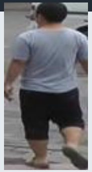
(ResNet50)

image_list_p1

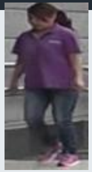
image_list_p2

p21

p14

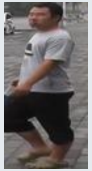


p23



p11

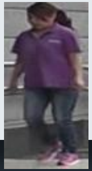
p12



p22

p31

p23



```
features_1 = extractor(image_list_p1)
features_2 = extractor(image_list_p2)
print("features_1 ",features_1)
print("features_2 ",features_2)

=>>>
features_1
torch.Size([5, 2048])
        [0.0785, 0.3583, 0.3029,  ..., 0.0520, 0.0000, 1.0073],
        [0.0664, 0.0527, 0.3985,  ..., 0.0327, 0.0454, 1.1085],
        [0.5219, 1.2649, 0.0334,  ..., 0.0906, 0.1035, 0.2110],
        [0.0283, 0.2683, 0.5947,  ..., 0.3594, 0.0563, 0.2664],
        [0.4860, 1.0593, 0.0162,  ..., 0.0346, 0.0486, 0.1544]

features_2
torch.Size([3, 2048])
        [0.3463, 1.4591, 0.0021,  ..., 0.1234, 0.0718, 0.1940],
        [0.0032, 0.0800, 0.2054,  ..., 0.1168, 0.0088, 0.6943],
        [0.4860, 1.0593, 0.0162,  ..., 0.0346, 0.0486, 0.1544]
```

```
image_list_p1 = [
    'test_image/p11.jpg',
    'test_image/p12.jpg',
    'test_image/p22.jpg',
    'test_image/p31.jpg',
    'test_image/p23.jpg',
]
```

```
image_list_p2 = [
    'test_image/p21.jpg',
    'test_image/p14.jpg',
    'test_image/p23.jpg'
]
```

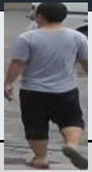
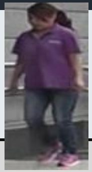
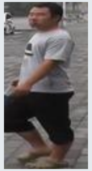
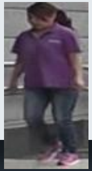
Compute distance matrix

use **euclidean**

$$d(x,y) := \sqrt{(x_1-y_1)^2 + (x_2-y_2)^2 + \cdots + (x_n-y_n)^2} = \sqrt{\sum_{i=1}^{n}(x_i-y_i)^2}$$

image_list_p1

image_list_p2

		p21	p14	p23
				
p11		689	71	666
p12		679	82	663
p22		54	652	45
p31		662	602	615
p23		69	625	0

```
image_list_p1 = [
    'test_image/p11.jpg',
    'test_image/p12.jpg',
    'test_image/p22.jpg',
    'test_image/p31.jpg',
    'test_image/p23.jpg',
]
```

```
image_list_p2 = [
    'test_image/p21.jpg',
    'test_image/p14.jpg',
    'test_image/p23.jpg'
]
```

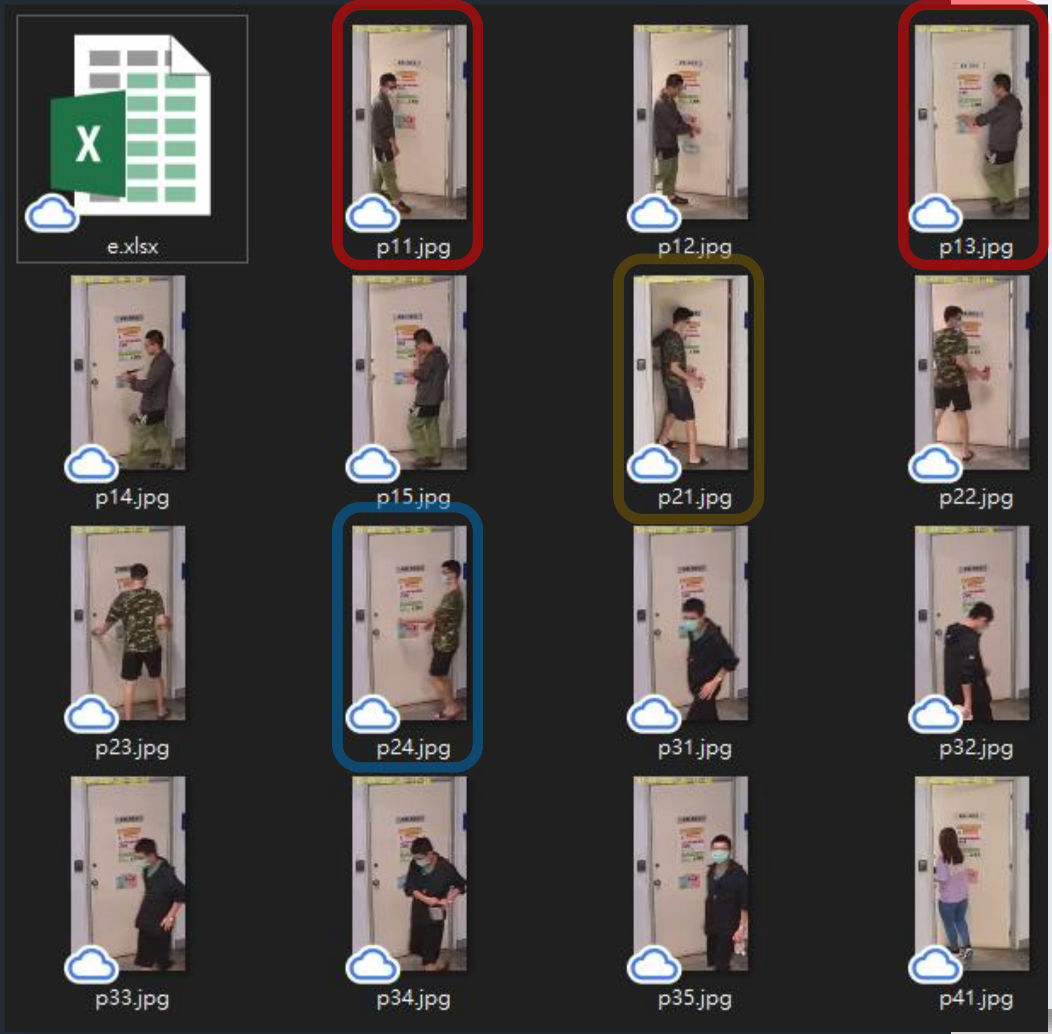
```
features_1 = extractor(image_list_p1)
features_2 = extractor(image_list_p2)

distmat = metrics.compute_distance_matrix(features_1, features_2)
print(distmat)
```

```
=>>>
[689.1064,     71.9318,      666.6927],
[679.5190,     82.5314,      663.0724],
[ 54.5557,     652.5553,     45.2095],
[662.2687,     602.6262,     615.7032],
[ 69.6683,      625.9954,    0.0000]
```

Compute distance matrix
use **cosine**

```
#cosine
distmat2 = metrics.compute_distance_matrix(features_1, features_2, metric='cosine')
print(distmat2)

=>>>
[0.5796531438827515,      0.0591344833374023,       0.5568877458572388],
[0.5758468508720398,      0.0683749318122863,       0.5579483509063721],
[0.04654204845428467,     0.545458018779754,        0.03835523128509521 5],
[0.5825185775756836,      0.517381191253662,        0.5374826192855835],
[0.06009882688522339,     0.527764081954956,        1.1920928955078125e-07]
```

```
features_1 = extractor(image_list_p1)
features_2 = extractor(image_list_p2)

distmat = metrics.compute_distance_matrix(features_1, features_2)
print(distmat)

=>>>
[689.1064,      71.9318,        666.6927],
[679.5190,      82.5314,        663.0724],
[ 54.5557,      652.5553,       45.2095],
[662.2687,      602.6262,       615.7032],
[ 69.6683,       625.9954,      0.0000]
```



EE3F test

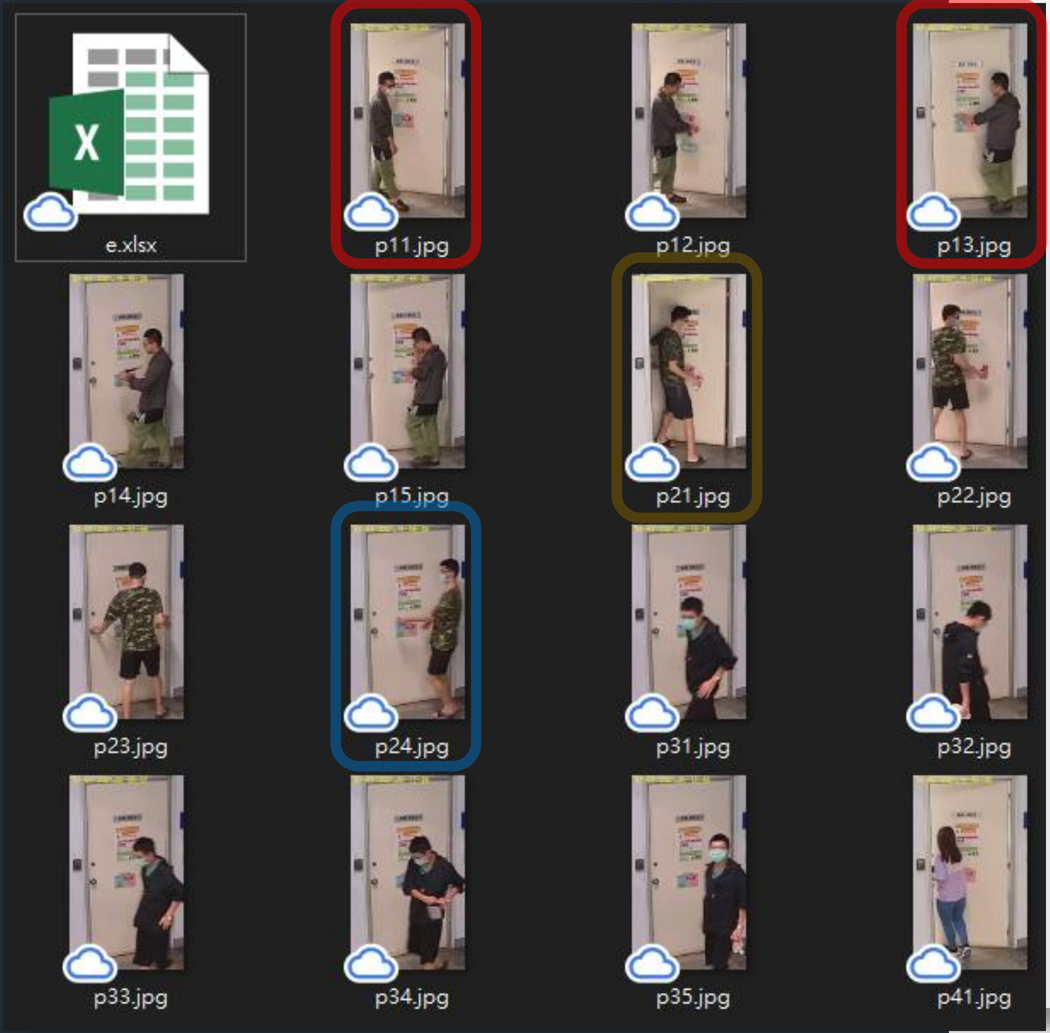
False Positive
(一樣)

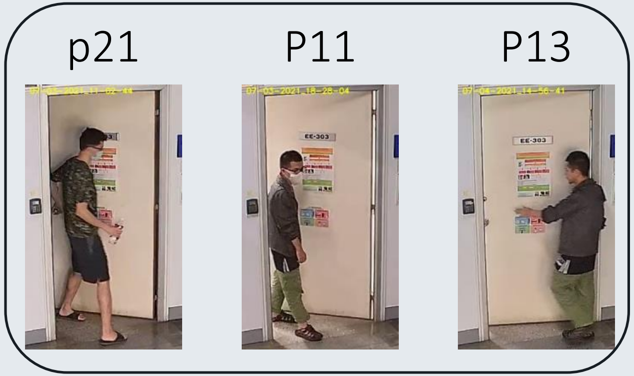
False Negative
(不一樣)

image total = 6+7=13

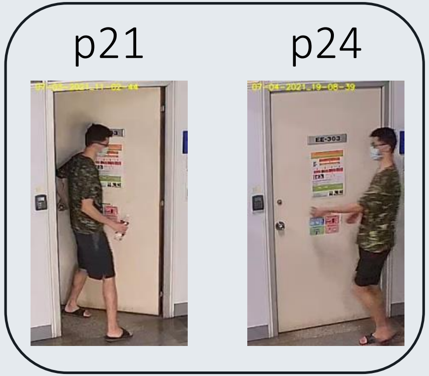
if V>110	Prediction					
	p21	p12	p23	p33	p14	p35
p11	50.9017	20.3506	247.2152	381.1587	77.9178	370.6215
p22	54.5635	127.8018	102.7195	340.7387	140.1973	341.6727
p13	75.845	35.5033	277.5178	385.7975	48.2986	380.8727
p34	452.6635	440.9745	373.1768	89.053	358.6364	84.5405
p24	188.0116	220.797	78.1254	227.1243	132.516	252.0209
p15	142.2623	76.3605	228.548	268.5231	36.2512	279.7692
p41	199.574	179.9592	343.4623	363.6335	186.4541	368.7164

	Ground truth					
	p21	p12	p23	p33	p14	p35
p11		1			1	
p22	1		1			
p13		1			1	
p34				1		1
p24	1		1			
p15		1			1	
p41						





False Positive
(一樣)



False Negative
(不一樣)

Next to do

Next to do

Person Re-Identification system		
1. Person detection	Object detection (MobileNetSSD) Openpose	done cont.
2. 特徵擷取網路架構	Torchreid (A Library for Deep Learning Person Re-Identification in Pytorch) backbone: ResNet50 backbone: OSNet Tensorflow -> Pytorch 世超-DSPF (backbone: SE-ResNeXt)	 done cont. cont.
3. 動態每天分群給編號 4. 使用3的分群編號，去做Query識別	Feature Extractor Compute distance matrix Real-time system	done done cont.
5. 確認編號的確實身份	結合Line bod應用	cont.