Masked Face Detection?

B10615040 陳洛翔 Axot
B10530301 許書豪 Jack
M10815822 Hoan

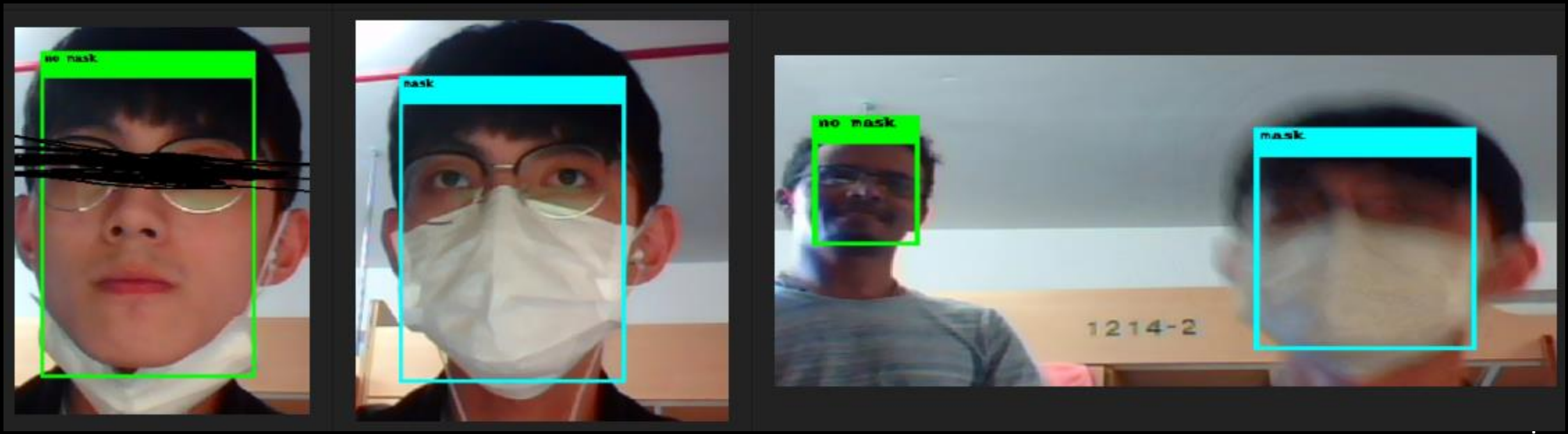
Table of contents

Introduction

- Objective
- GUI
- Flow chart
- Model we used

Objective

Our idea was to do something for this COVID-19 Era, creating a safe environment as we prepare for open facilities again, help people and their families free form harm , so then we want to do something that have more value and can have more extension.

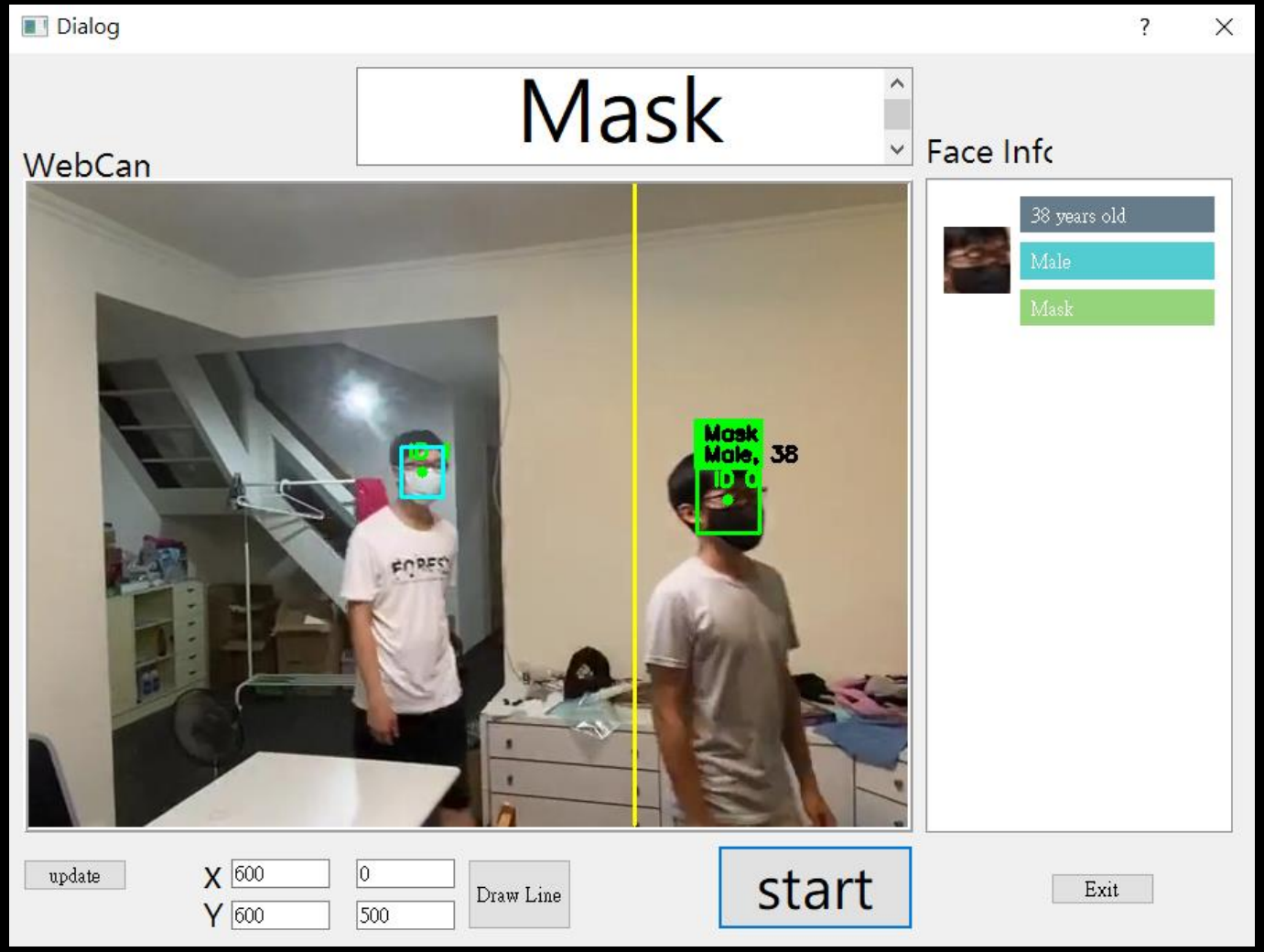


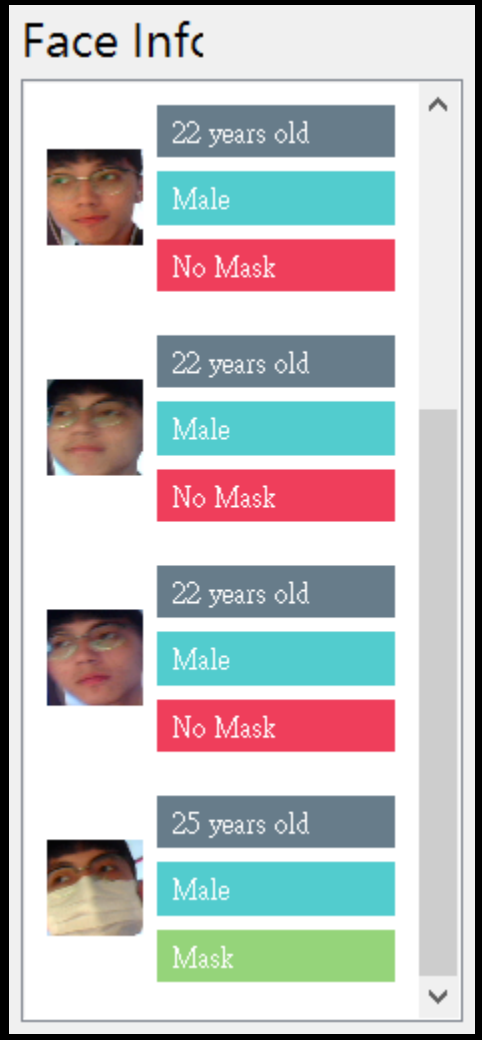
Objective

So we did….

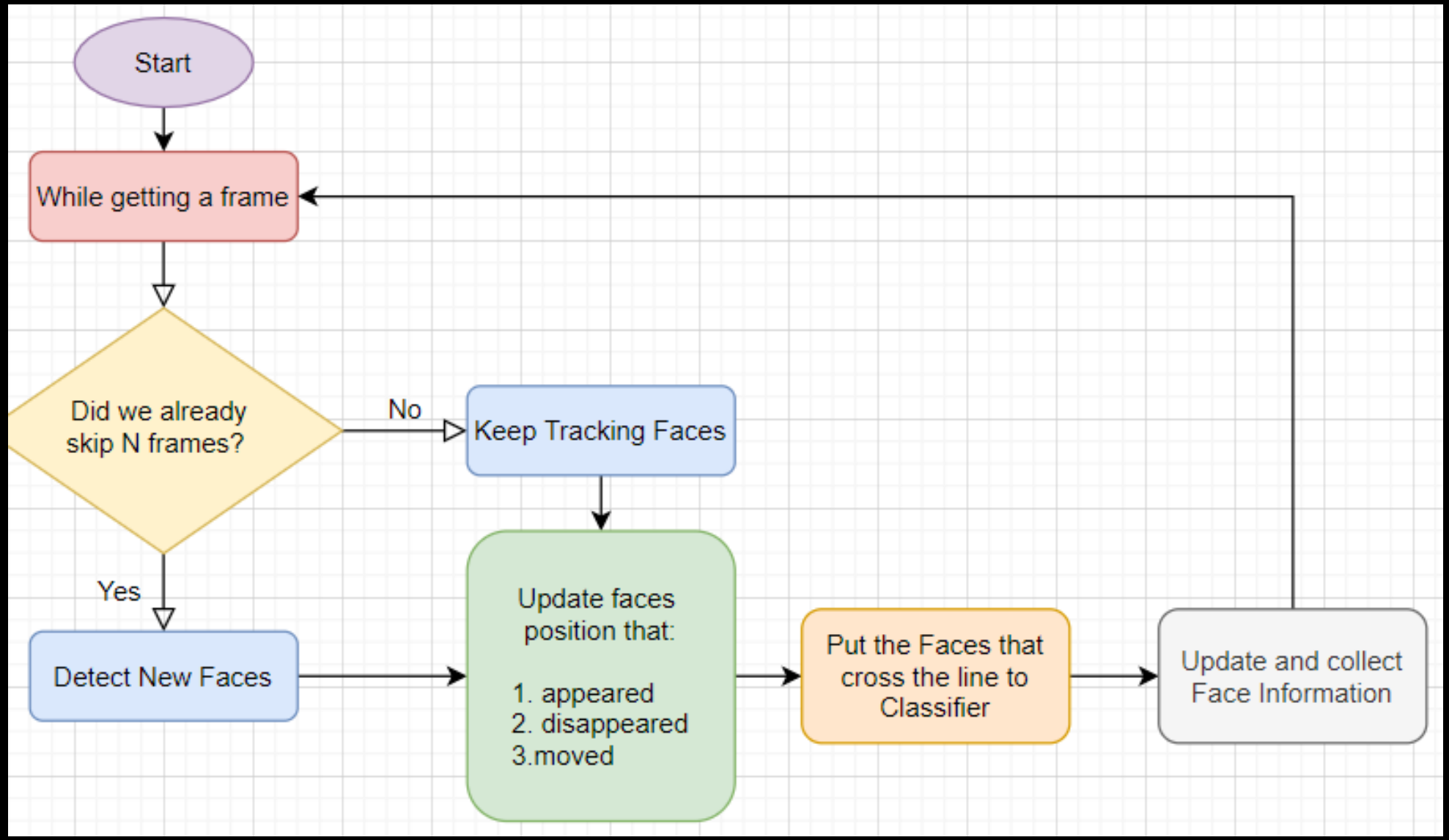
- Collect the people's' information while they get into/out a place e.g. convenience store, school, etc. And then we can do some analysis.
- With this analytic enables facility owners to know who enters their building with or without a required mask.
- People's' information contains
  - Wearing mask or not
  - Age
  - Gender
  - Timestamp

GUI


Dialog
Mask
WebCan
Face Infc
38 years old
Male
Mask
Mask
Male, 38
update
X
600
0
Y
600
500
Draw Line
start
Exit


Face Infc
22 years old
Male
No Mask
22 years old
Male
No Mask
22 years old
Male
No Mask
25 years old
Male
Mask

Flow Chart


Start
While getting a frame
Did we already skip N frames?
No
Keep Tracking Faces
Yes
Detect New Faces
Update faces position that:
1. appeared
2. disappeared
3.moved
Put the Faces that cross the line to Classifier
Update and collect Face Information

Model we used

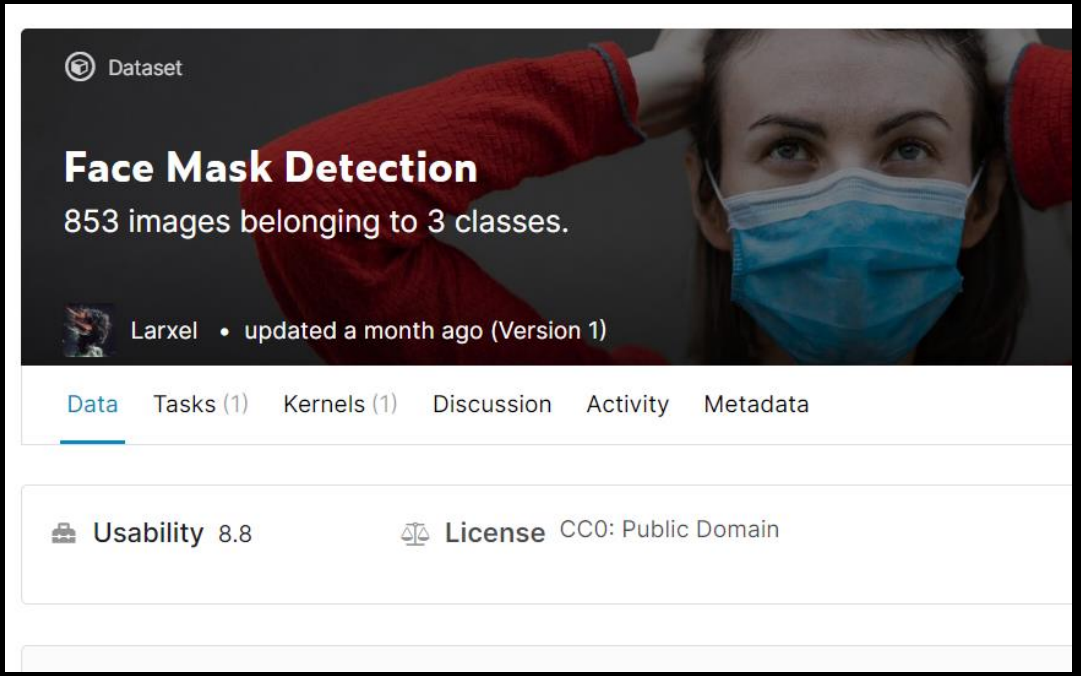
- Detection
  - Openvino pretrained model: face-detection-adas-0001
- Classifier
  - Openvino pretrained model: age-gender-recognition-retail-0013
  - Model we trained: MaskFaceClassifier

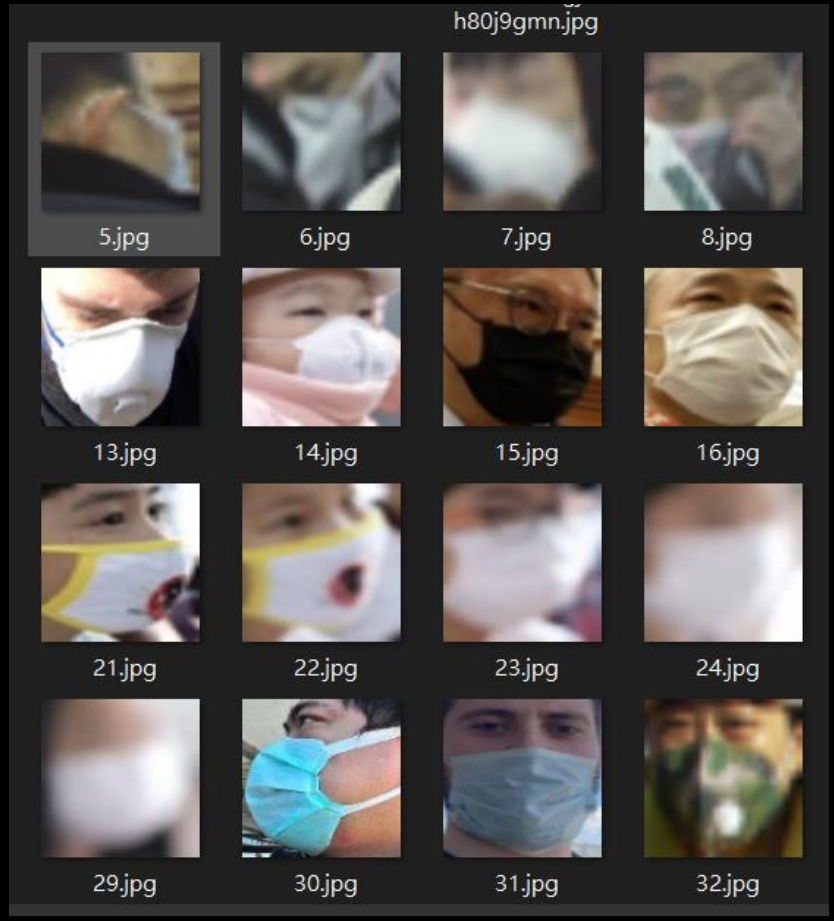
Mask Face Classifier Training

- Data source
- Model Structure
- Training Result
- Feature Map

Data source

https://www.kaggle.com/andrewmvd/face-mask-detection





Then we can get 716 with and without mask face

Model Structure

Total params:110,721

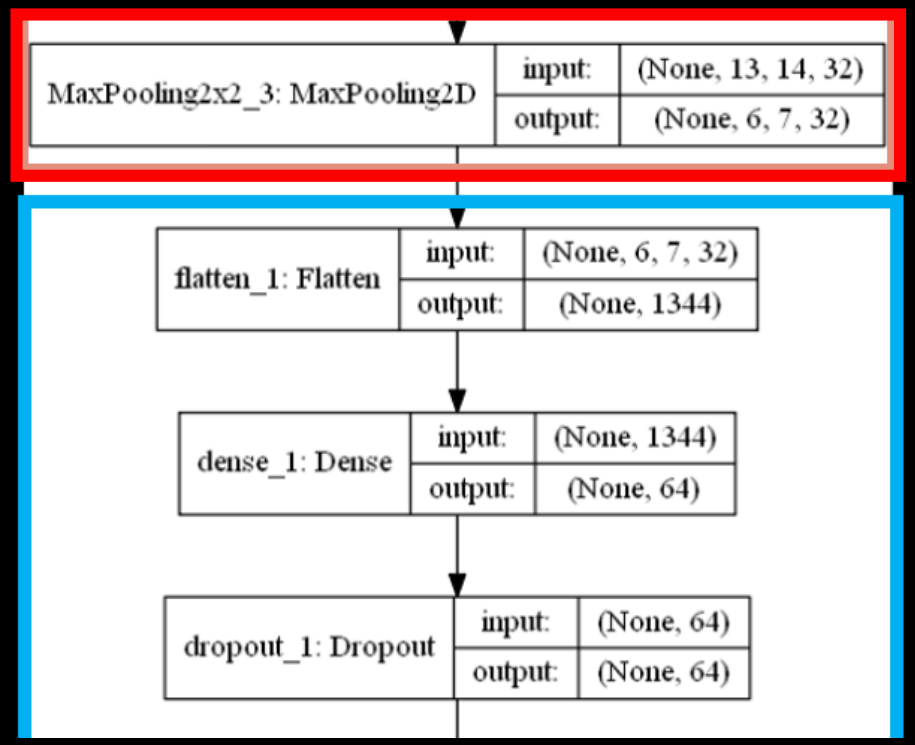
conv3x3_1: Conv2D	input:	(None, 62, 62, 3)
	output:	(None, 60, 60, 32)
Relu_1: Activation	input:	(None, 60, 60, 32)
	output:	(None, 60, 60, 32)
MaxPooling2x2_1: MaxPooling2D	input:	(None, 60, 60, 32)
	output:	(None, 30, 30, 32)
conv3x1_1: Conv2D	input:	(None, 30, 30, 32)
	output:	(None, 28, 30, 32)
Relu_2: Activation	input:	(None, 28, 30, 32)
	output:	(None, 28, 30, 32)
conv3x3_2: Conv2D	input:	(None, 28, 30, 32)
	output:	(None, 26, 28, 64)
Relu_3: Activation	input:	(None, 26, 28, 64)
	output:	(None, 26, 28, 64)
MaxPooling2x2_2: MaxPooling2D	input:	(None, 26, 28, 64)
	output:	(None, 13, 14, 64)
conv1x1_1: Conv2D	input:	(None, 13, 14, 64)
	output:	(None, 13, 14, 32)
Relu_4: Activation	input:	(None, 13, 14, 32)
	output:	(None, 13, 14, 32)
MaxPooling2x2_3: MaxPooling2D	input:	(None, 13, 14, 32)
	output:	(None, 6, 7, 32)
flatten_1: Flatten	input:	(None, 6, 7, 32)
	output:	(None, 1344)
dense_1: Dense	input:	(None, 1344)
	output:	(None, 64)
dropout_1: Dropout	input:	(None, 64)
	output:	(None, 64)
dense_2: Dense	input:	(None, 64)
	output:	(None, 1)
activation_1: Activation	input:	(None, 1)
	output:	(None, 1)

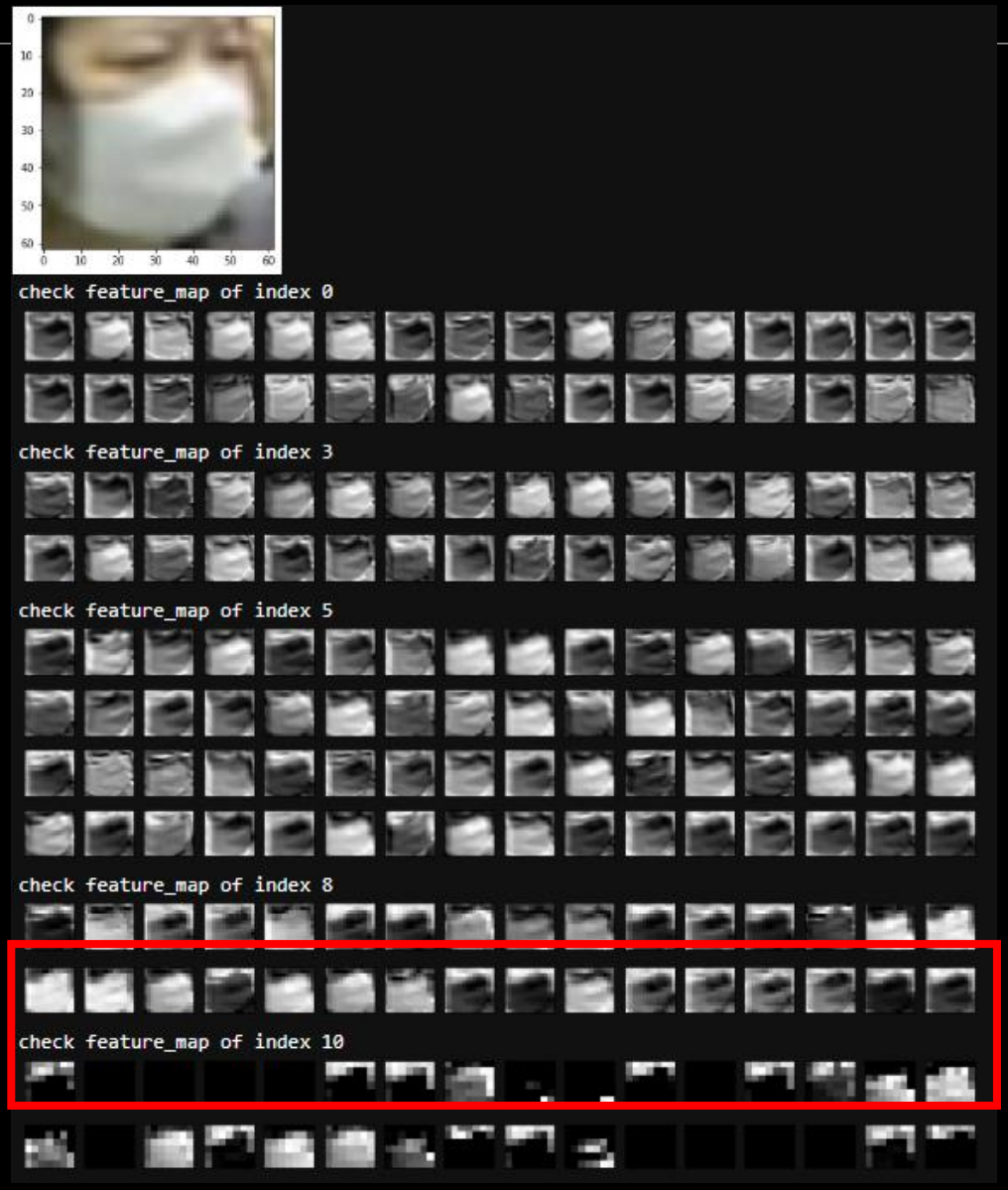
Training Result

Train acc: 0.9589
Val acc: 0.9930

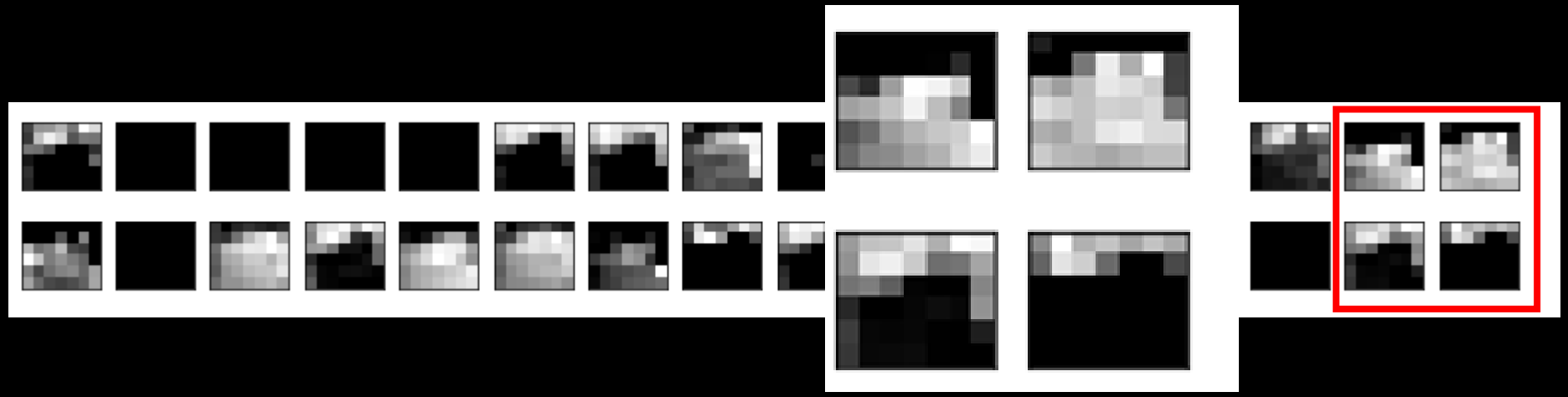
Total training time:
less than 1 min!

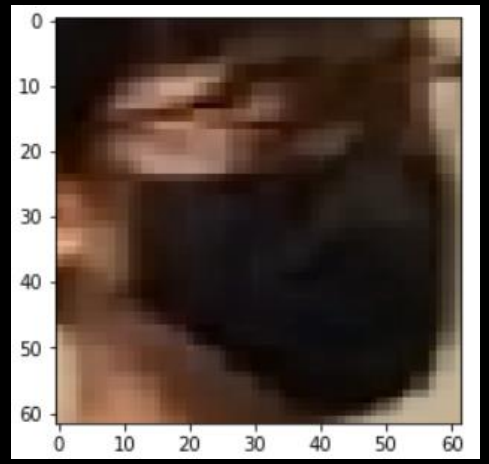
Feature map

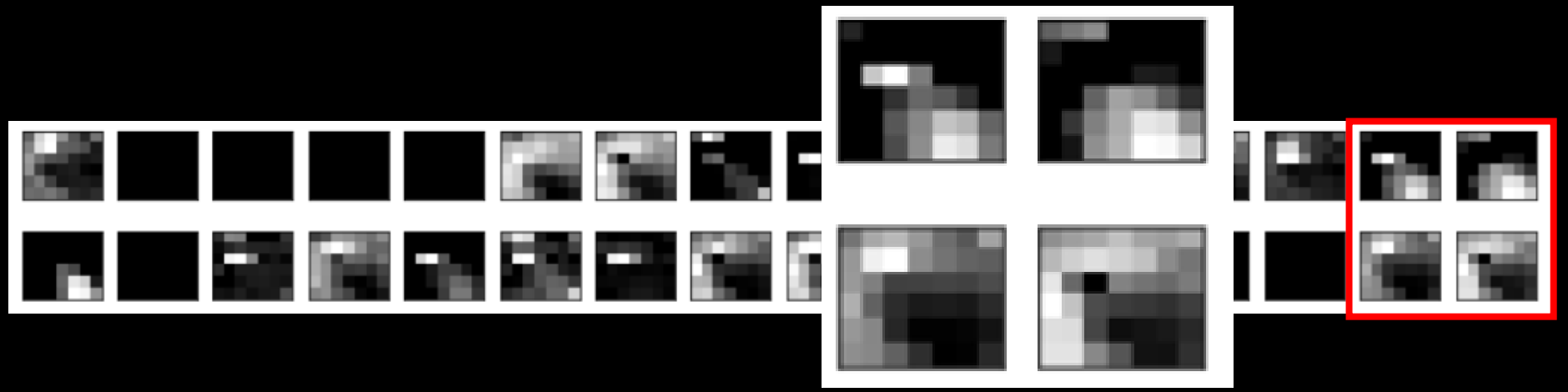

MaxPooling2x2_3: MaxPooling2D
input: (None, 13, 14, 32)
output: (None, 6, 7, 32)
flatten_1: Flatten
input: (None, 6, 7, 32)
output: (None, 1344)
dense_1: Dense
input: (None, 1344)
output: (None, 64)
dropout_1: Dropout
input: (None, 64)
output: (None, 64)


check feature_map of index 0
check feature_map of index 3
check feature_map of index 5
check feature_map of index 8
check feature_map of index 10

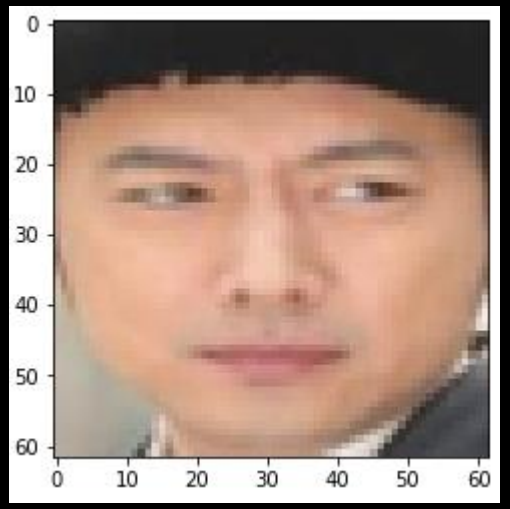
Feature map

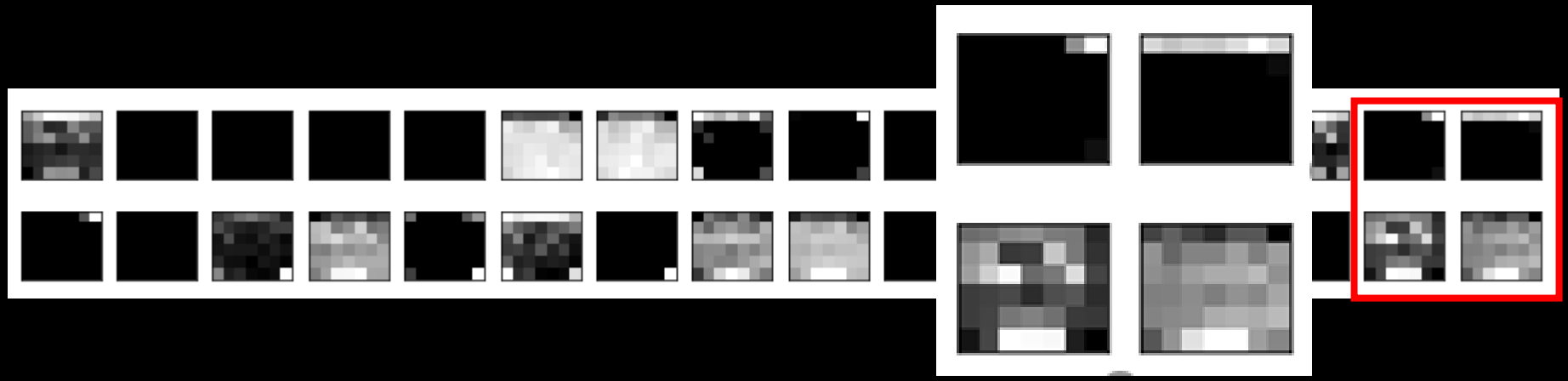


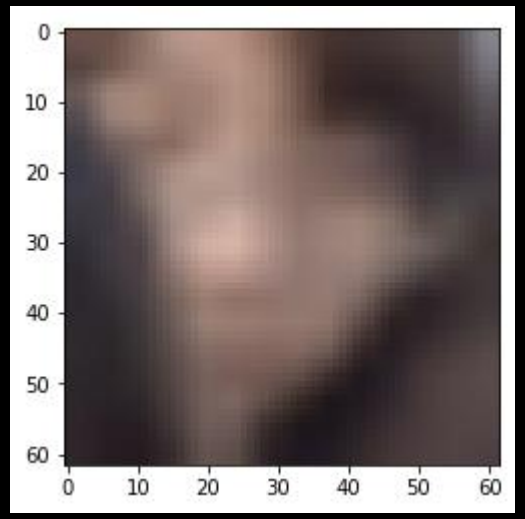


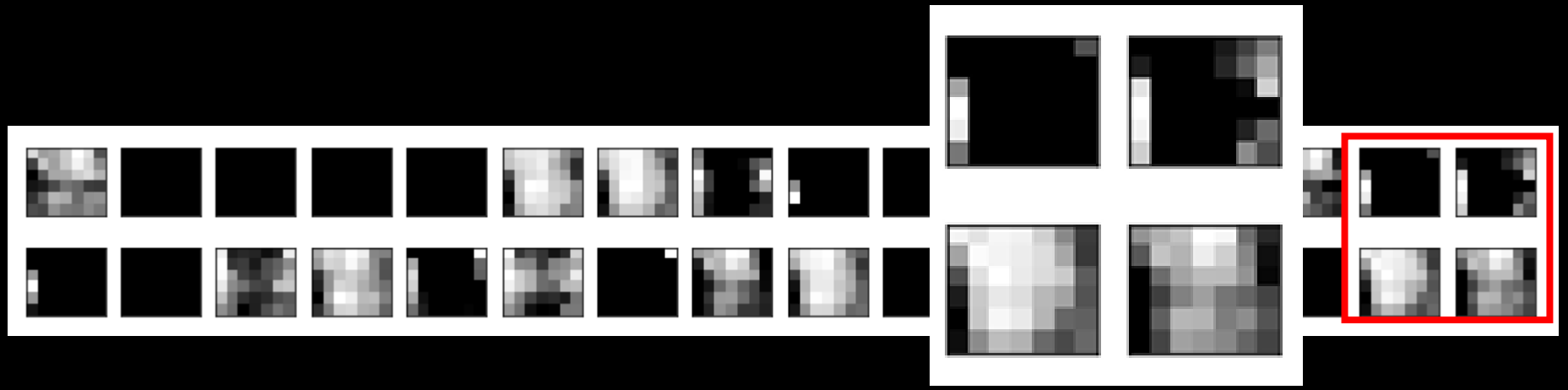


Feature map


0
10
20
30
40
50
60
0
10
20
30
40
50
60




0
10
20
30
40
50
60
0
10
20
30
40
50
60



Tracker

- Pros and Cons
- Detector and dlib.correlation_tracker
- Pyimagesearch – CentroidTracker
- Performance

Pros and Cons

- Pros:

  Have higher speed.

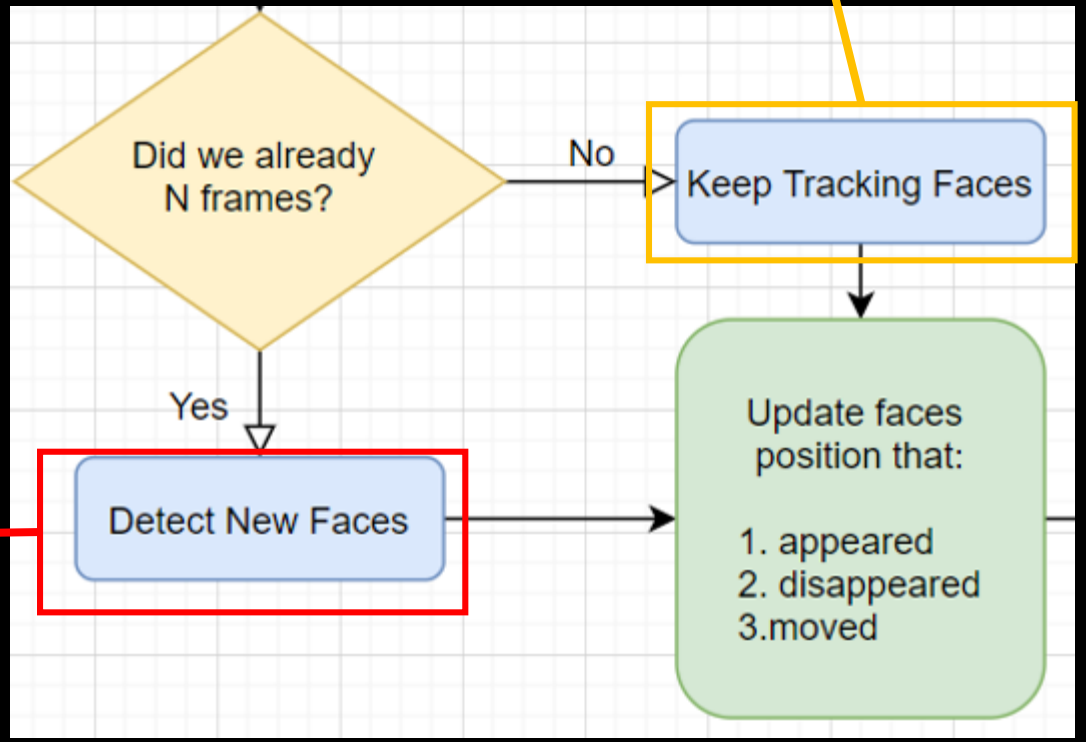
  Detection is more computation expensive than tracker. (And we only have cpu.)

- Cons:

  Have lower accuracy.

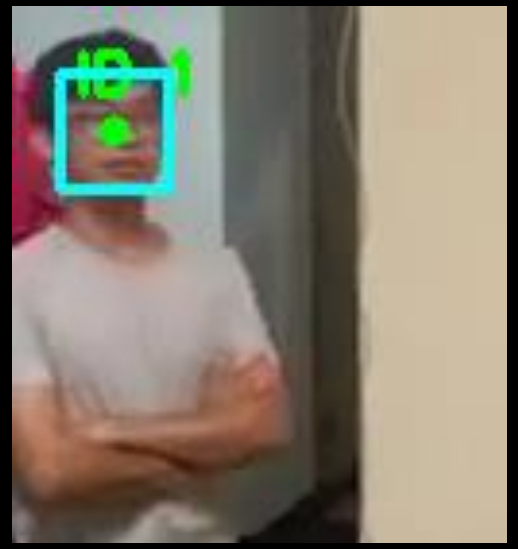
  Tracker track the face by correlation of image

Detector and dlib.correlation_tracker



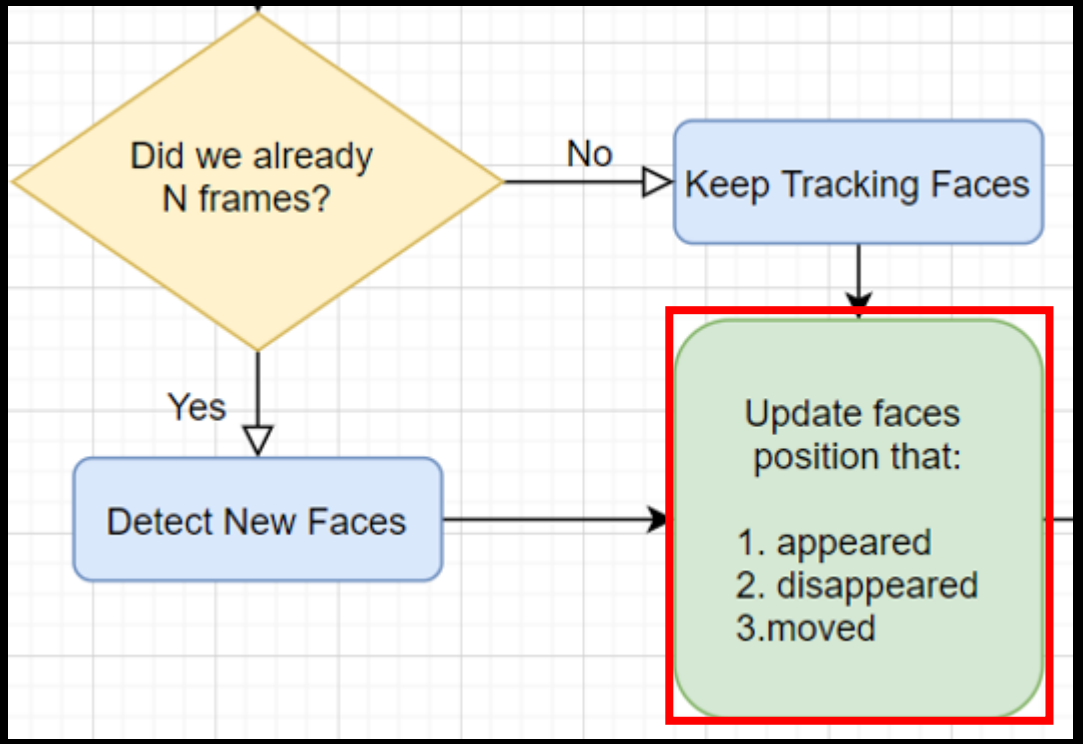
Detector and dlib.correlation_tracker

- Every *N* Frames, Detector refreshs/tells correlation_tracker that where the faces actually are.
- Then correlation_tracker keep tracking the faces by the correlation of a sequence of frames



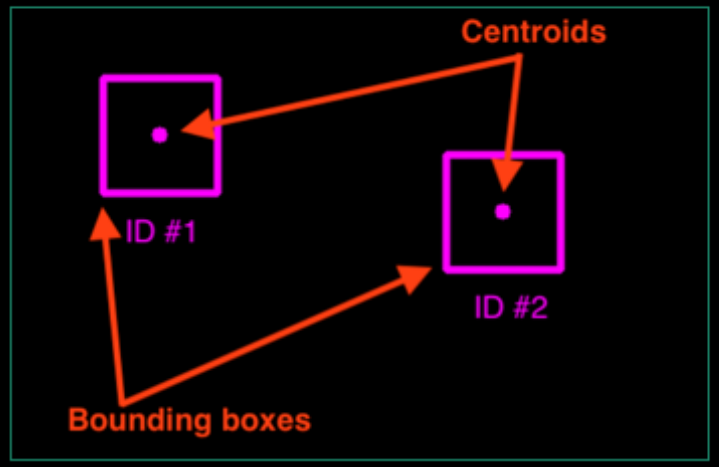
Pyimagesearch - CentroidTracker

- Based on the track result (bounding box) from Detector and CorrelationTracker, CentroidTracker decide to new an objects , delete objects and track objects

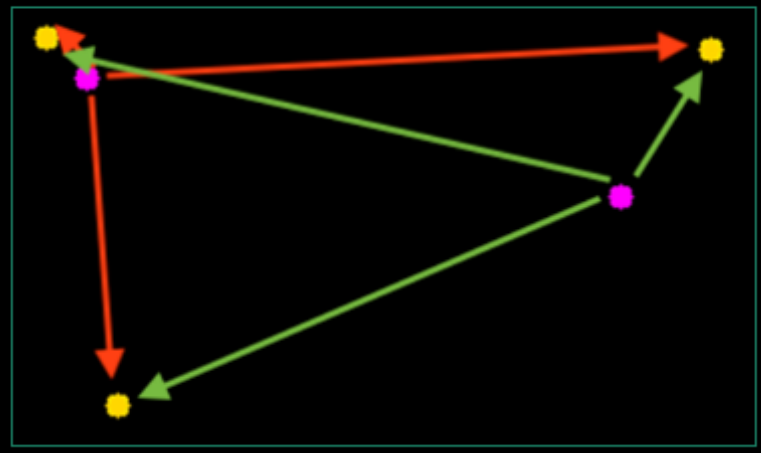


Pyimagesearch - CentroidTracker

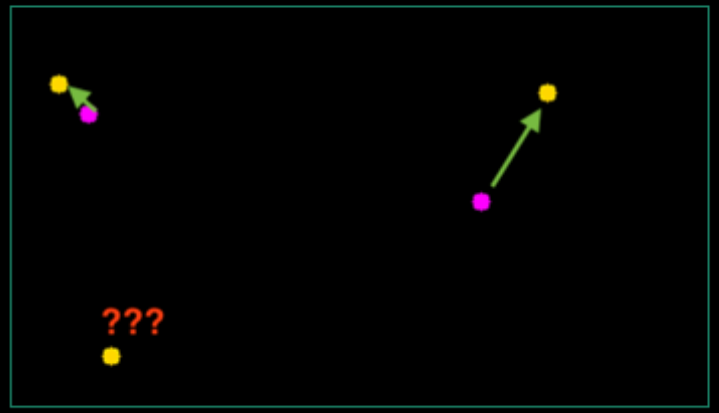
1.



2.



3.



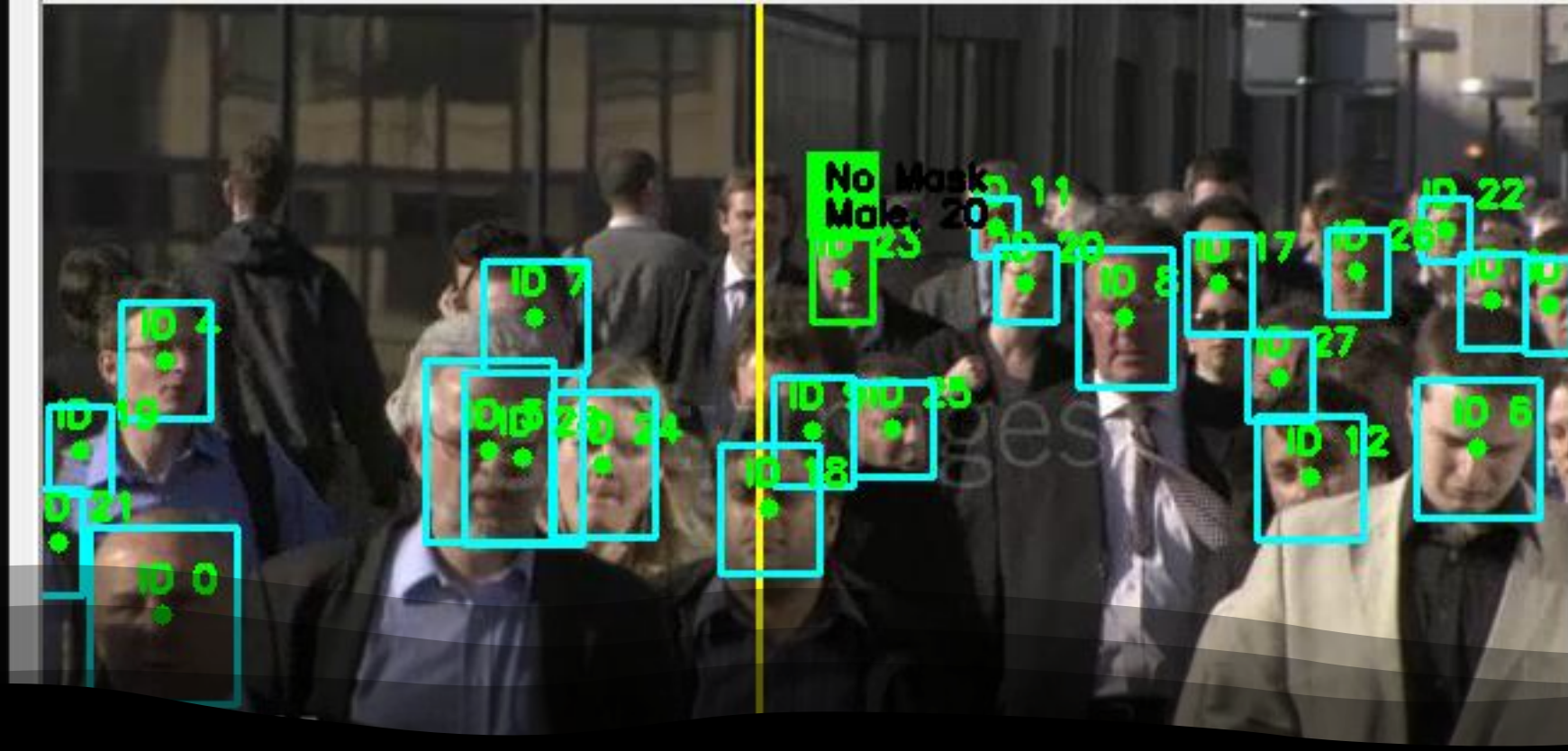
4.

```
t:spend: 0.027926s
t:spend: 0.02922s
t:Call detector ~~~
t:spend: 0.073359s
t:spend: 0.027943s
t:spend: 0.025006999999999998s
t:spend: 0.029967999999999998s
t:spend: 0.029908999999999998s
t:spend: 0.031874s
t:spend: 0.040889999999999996s
t:spend: 0.04588s
t:Call detector ~~~
t:spend: 0.090
```

Performance

- If we only have one face
- - Detector:
- 0.070 ~ 0.09s per frame
- - Correlation Tracker:
- 0.025 ~ 0.04s per frame
- Tracker is three times faster than Detector !

```
:spend: 0.18612599999999999s
:spend: 0.151972s
:spend: 0.158835s
:spend: 0.14660299999999998s
:spend: 0.139372s
:spend: 0.144289s
:spend: 0.137288s
:spend: 0.144654s
:Call detector ~~~
:spend: 0.195896s
:spend: 0.148143s
:spend: 0.149366s
:spend: 0.15064s
:spend: 0.14876699999999998s
:spend: 0.149844s
```



Performance

- If we have lots of face
- - Detector:
- 0.18 ~ 0.19s per frame
- - Correlation Tracker:
- 0.14 ~ 0.16s per frame
- The affect is not that obvious.

Analysis/Visualization

- Data we collected
- What can we Analyze/Visualize?

Data we collected

- timestamp
- mask
- gender
- age

timestamp	mask	gender	age
1592786280	No Mask	Male	30
1592786282	Mask	Male	29
1592786282	No Mask	Male	48
1592786284	No Mask	Male	38
1592786289	No Mask	Male	29
1592786315	No Mask	Male	21
1592786317	No Mask	Male	40
1592786319	No Mask	Male	34
1592786323	No Mask	Male	37
1592786323	No Mask	Male	47
1592786327	No Mask	Male	28
1592786330	No Mask	Male	51
1592786339	No Mask	Male	50

What can we Analyze/Visualize

- *Relationship between mask and (age, gender)*
- Distribution of gender, age, (gender, age)
- People flow (by gender) (by age)
  - In the month distributed by “date”
  - Relationship with “isHoliday”
  - Stats for each hour
  - ……..
  - Combine all these stuff above
- If we are in a big building, then we can stats for each area in the building.
- Lots of work!

mask	gender	age	year	month	date	day	hour	isHoliday
No Mask	Female	68	2020	6	1	0	15	False
Mask	Female	32	2020	6	1	0	15	False
Mask	Male	37	2020	6	1	0	15	False
No Mask	Female	47	2020	6	1	0	16	False
Mask	Female	68	2020	6	1	0	16	False
...	...	...	...	...	...	...	...	...
Mask	Male	75	2020	7	1	2	6	False
No Mask	Female	67	2020	7	1	2	6	False
Mask	Male	36	2020	7	1	2	7	False
No Mask	Male	31	2020	7	1	2	7	False
Mask	Female	52	2020	7	1	2	7	False

Demo

End

https://github.com/AxotZero/NTUST--MaskFaceDetection