
Count that face
Gary Kao

Q: How many people in these pictures?

A: Easy! There is one.

A: Of course. 5

A: Maybe…~50

A: Hmm……

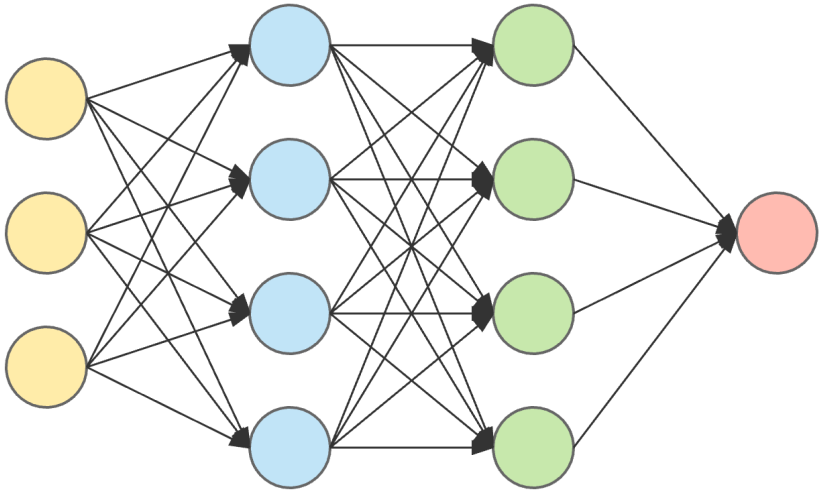
Our brain is smart enough to recognize faces and count it instantly.
However, there is a limit.

How about computers?

Convolutional neural network (CNN)

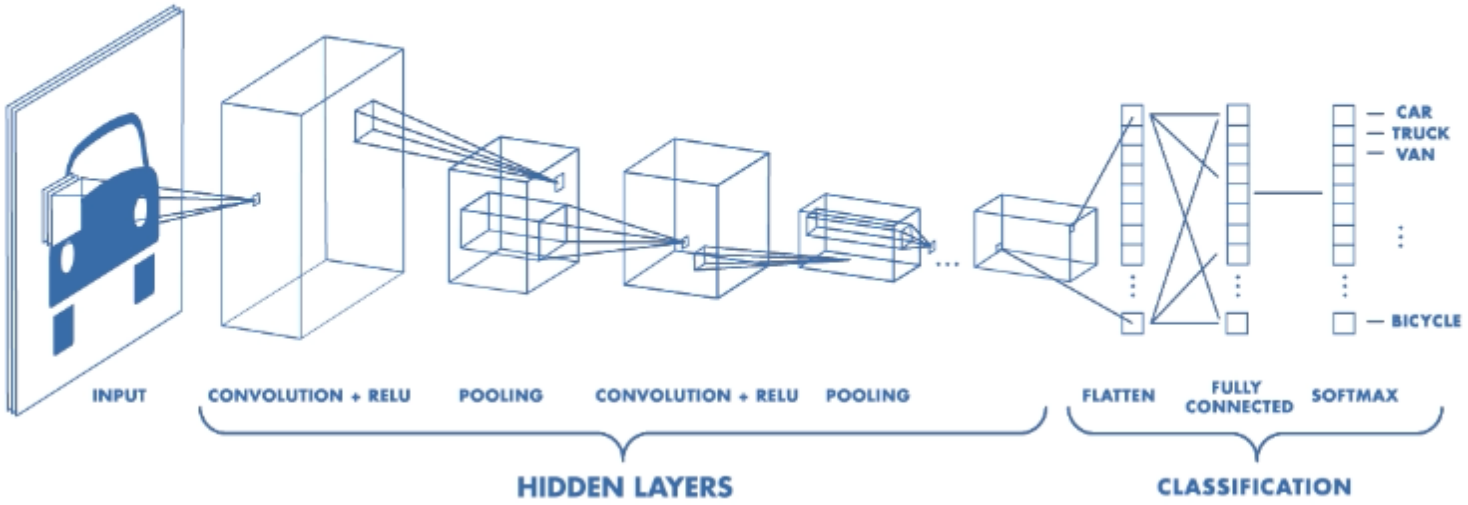
- A member of neural network families
- A great tool for image recognition
  - ex: facial images

Traditional 2D neural network



input layer hidden layer 1 hidden layer 2 output layer

CNN (3D neural network)



My plan

Part1 Facial recognition classifier

Datasets

- Facial images
  - ➢Faces only
  - ➢Faces + bodies
- Background images

Model training

- Keras package

Model evaluation

- Test models with images from different datasets

Part2 Apply classifier in crowd counting

Sliding windows

- Apply a grid of windows onto the picture

Sliding windows & classifier

- Each window will be analyzed by the classifier

Heatmap & counting

- All of the windows confirmed to be faces will indicate the number of faces by heatmap

*It would make more sense when we get to later slides

Part1
Datasets

Facial images

1. **MS-Celeb-1M:** 1 million celebrities images from Microsoft
    - **Sample ImageThumbnails:** 1522 images with celebrities with faces and bodies
    - **Sample FaceCropped:** 1392 images with celebrities with faces, necks, and shoulders
    - **Sample FaceAligned:** 1678 images with **c**elebrities with faces only

Thumbnail

Cropped

Aligned

Part1
Datasets

Facial images

2. **Labeled faces in the wild:** 13,000 images of faces collected from the web
   - Abbreviation of the name will be ‘lfw’
   - Similar to MS-Celeb-1M cropped faces



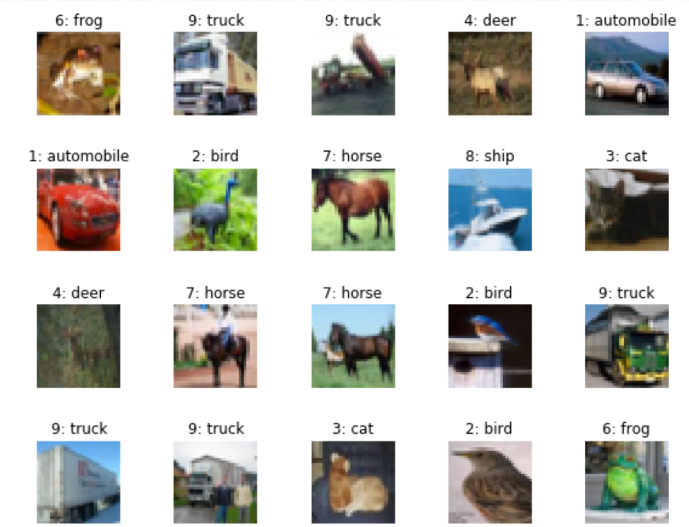
3. **Caltech Human Face Front:** 450 images
   - Abbreviation of the name will be ‘face’
   - Similar to MS-Celeb-1M cropped faces


Part1
Datasets

Non-facial images

1. **Canadian institute for advanced research (CIFAR10):** 60,000 images
   - Abbreviation of the name will be ‘C10’
2. **Caltech Cars (Rear) dataset:** 1155 images
   - Abbreviation of the name will be ‘car’
3. **Caltech Cars Rear background:** 1155 images
   - Abbreviation of the name will be ‘bg’
4. **Place365 CNNs:** 1.8 million images
   - Abbreviation of the name will be ‘ls’

C10



car

bg

ls

Part1
Model training

1. Image transformation by cv2 package: to 32 x 32

2. Training and testing data split & preprocessing:

```
from sklearn.model_selection import train_test_split
X_merge_train, X_merge_test, y_merge_train, y_merge_test = train_test_split(X, y, test_size = 0.33, stratify=y)
```

```
# Normalize data set to 0-to-1 range
X_merge_train = X_merge_train.astype('float32')
X_merge_test = X_merge_test.astype('float32')
X_merge_train /= 255
X_merge_test /= 255
```

```
# Categorize labes to [1, 0] as 0 and [0, 1] as 1
y_merge_train = keras.utils.to_categorical(y_merge_train, 2)
y_merge_test = keras.utils.to_categorical(y_merge_test, 2)
```

Part1
Model training

3. Model's summary

Layer (type)	Output Shape	Param #
conv2d_21 (Conv2D)	(None, 32, 32, 32)	896
conv2d_22 (Conv2D)	(None, 30, 30, 32)	9248
max_pooling2d_11 (MaxPooling	(None, 15, 15, 32)	0
dropout_15 (Dropout)	(None, 15, 15, 32)	0
conv2d_23 (Conv2D)	(None, 15, 15, 64)	18496
conv2d_24 (Conv2D)	(None, 13, 13, 64)	36928
max_pooling2d_12 (MaxPooling	(None, 6, 6, 64)	0
dropout_16 (Dropout)	(None, 6, 6, 64)	0
flatten_5 (Flatten)	(None, 2304)	0
dense_9 (Dense)	(None, 512)	1180160
dropout_17 (Dropout)	(None, 512)	0
dense_10 (Dense)	(None, 2)	1026

Total params: 1,246,754
Trainable params: 1,246,754
Non-trainable params: 0

Part1
Model evaluation

Here I use **model 8** for further crowd counting.

Reason:

1. Low accuracy in 'full' and 'face' datasets.

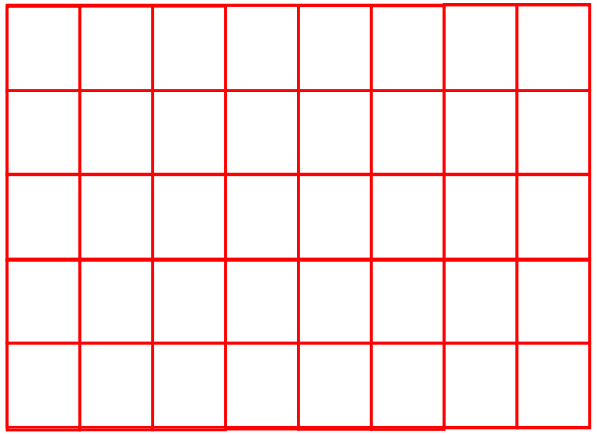
2. High accuracy in 'lfw'

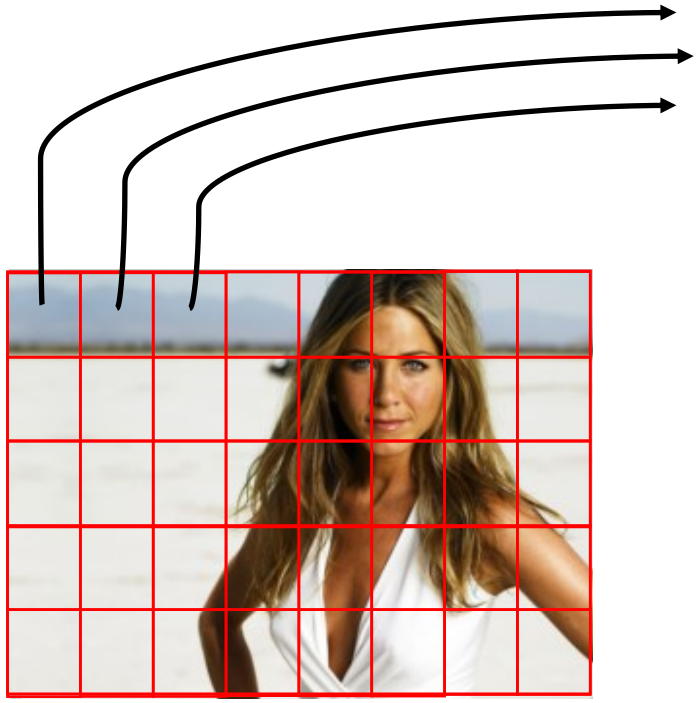
Model	Training dataset				Accuracy from other facial images Red: not included in the training				
	Non-facial images		**Facial images**		**align**	**crop**	**full**	**face**	**lfw**
model	C10	60,000	align	1,375	0.98	0.04	0.03	0.02	0.01
model_slice	C10	1,375	align	1,375	0.98	0.38	0.22	0.31	0.57
model3 (gray scale)	C10	1,375	align	1,375	0.99	0.17	0.1	0.08	0.36
model4	ls	1,375	align	1,375	0.99	0.08	0.06	0.04	0.23
model5	ls	4,256	align full crop	4,256	0.98	0.97	0.88	0.74	0.86
model6	ls	13,233	lfw	13,233	0.32	0.47	0.15	0.5	1
model7	ls	10,000	lfw	10,000	0.28	0.58	0.23	0.52	1
model8	**ls**	**2,750**	**align crop**	**2,750**	**0.99**	**0.96**	**0.52**	**0.51**	**0.91**

Part2
Sliding windows + facial classifier

+

A grid of windows

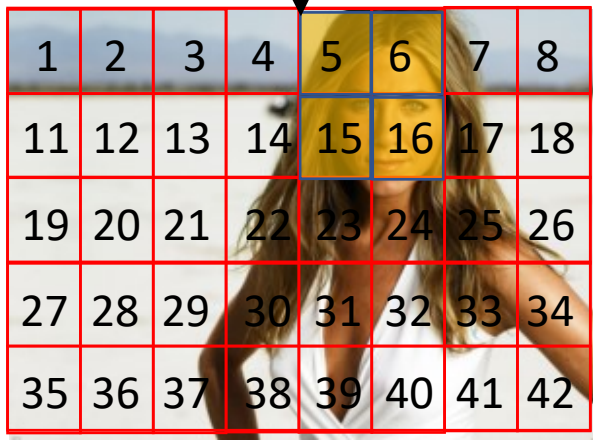




Each window will go through the classifier and cover certain size of pixels (ex: 2 x 2)

* 2 x 2 is just for demonstration prupose

Facial classifier trained by CNN



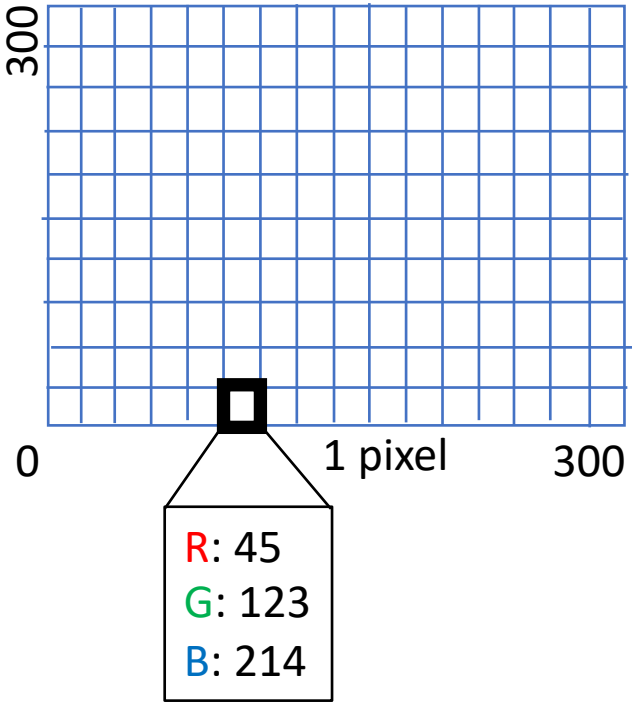
Identify the location of the face in 5, 6, 15, and 16 windows

Part2
Heat map

300 x 300 pixels

=

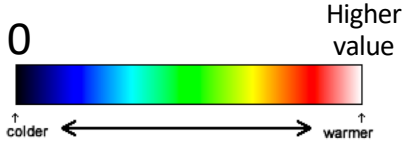
Each pixel has 3 channels (R, G, and B)
Each channel has a range from 0 ~ 255



Step1

Make each pixel to be 0 & present the image in a heat map

In the heat map, the color changes as the values and 0 represents black color

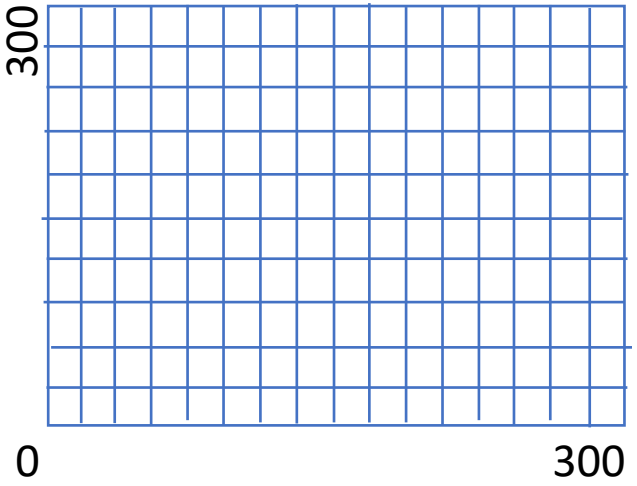


Part2
Sliding windows + facial classifier + Heat map

300 x 300 pixels

Each pixel has 3 channels (R, G, and B)
Each channel has a range from 0 ~ 255

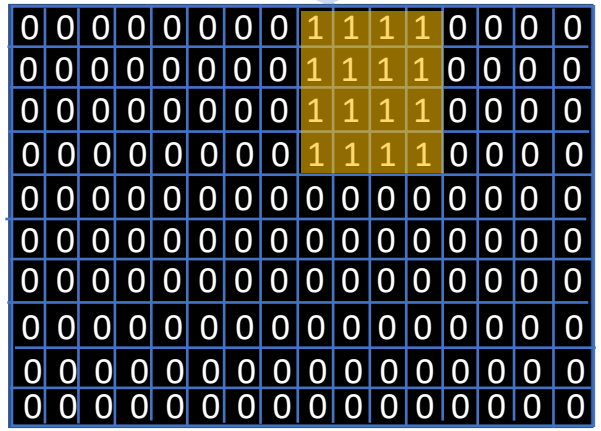
=



Step1

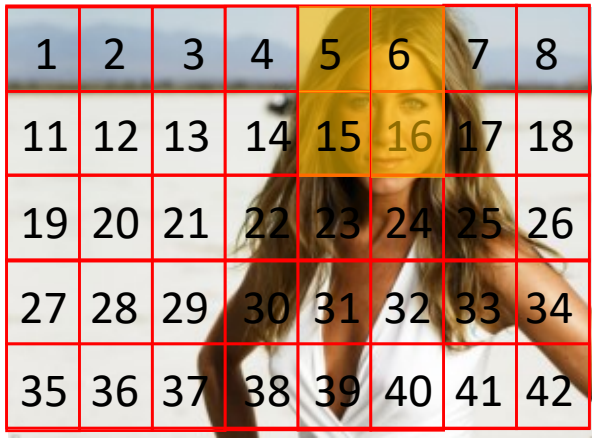
Make each pixel to be 0 & present the image in a heat map

Step3



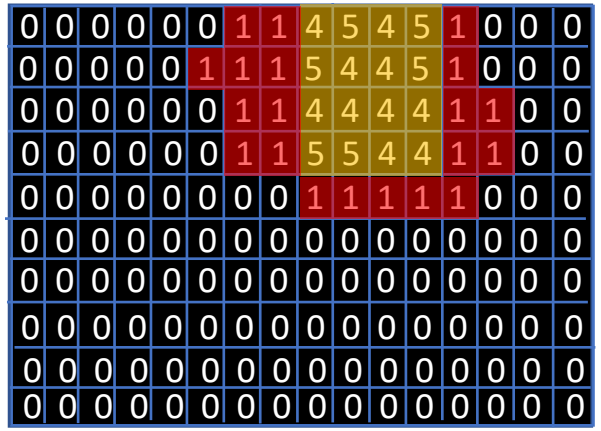
Step2

Each pixel in each window =1
(Here is 2 x 2 pixels / window)

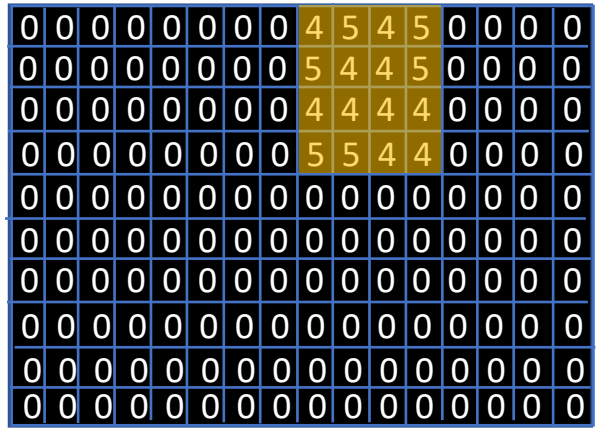


Part2
Sliding windows + facial classifier + Heat map

Here I use 3 different size of windows to overlap with each other. The face area will show higher values than other areas and present a different color.

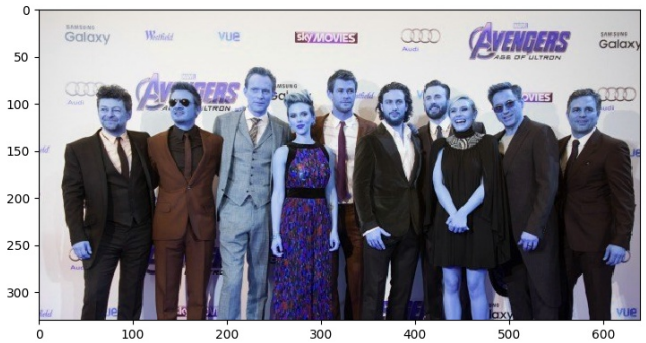


Here I set up a threshold in the function to select higher values only. For example, threshold = 4

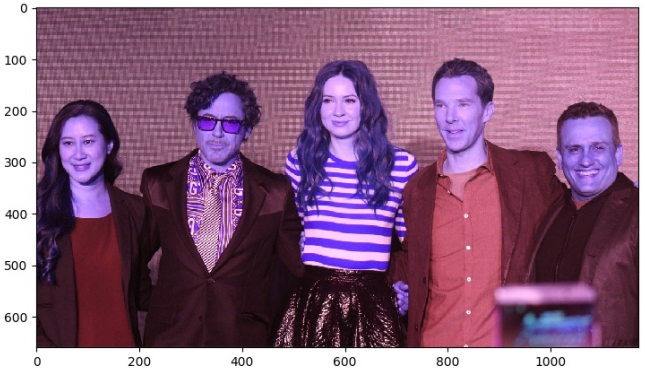


Part3
The testing of facial recognition

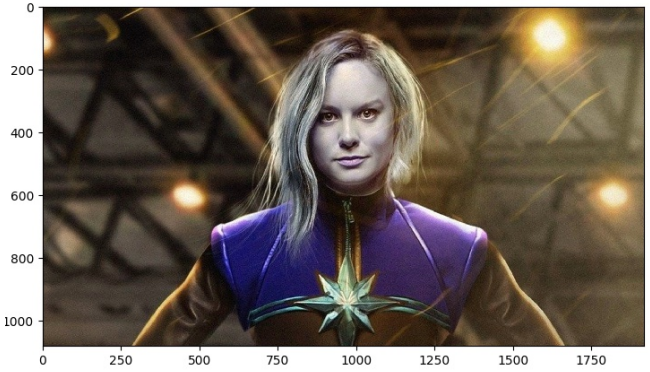
10 faces



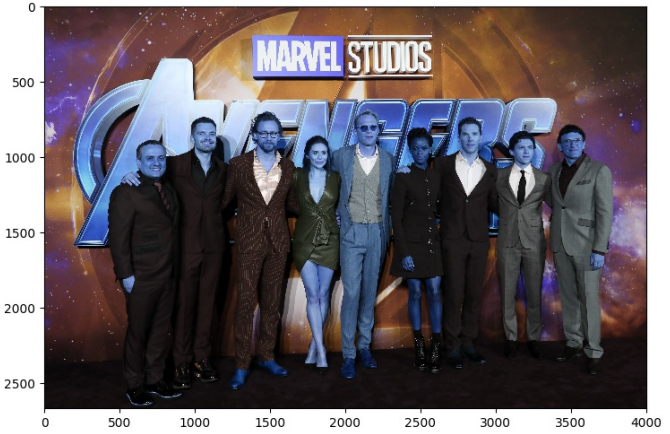
5 faces



1 face

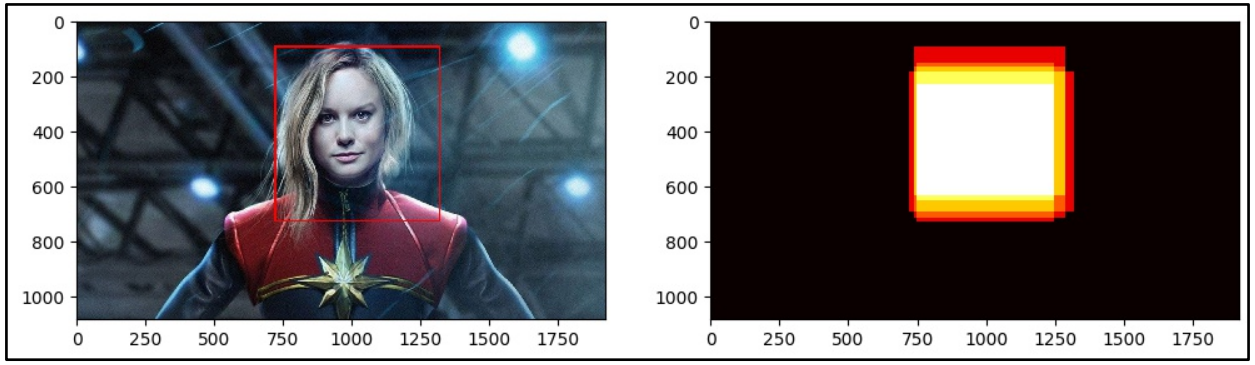


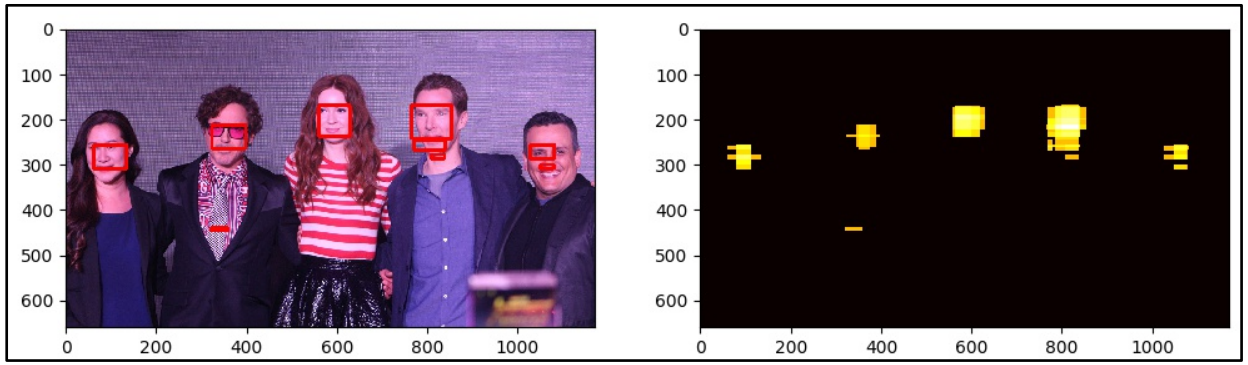
9 faces



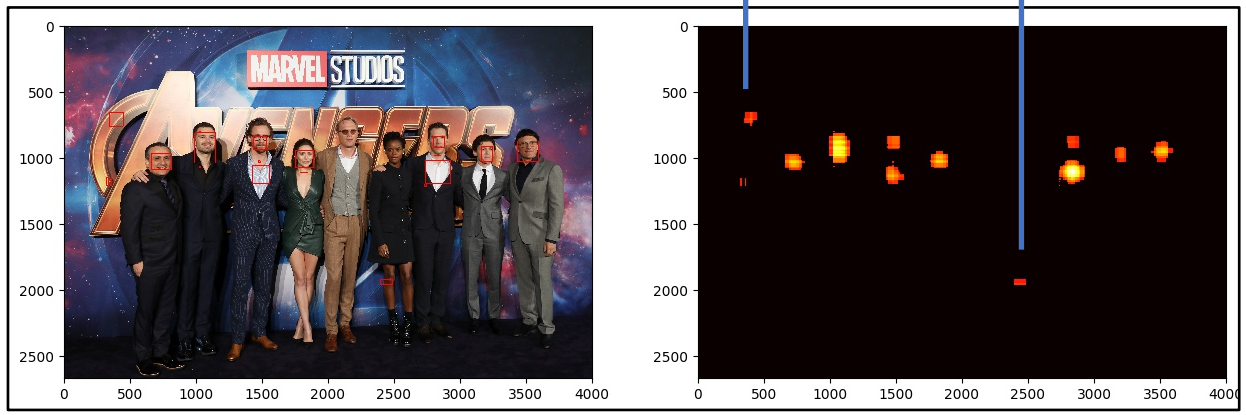
* The color is in BGR instead of RGB mode.

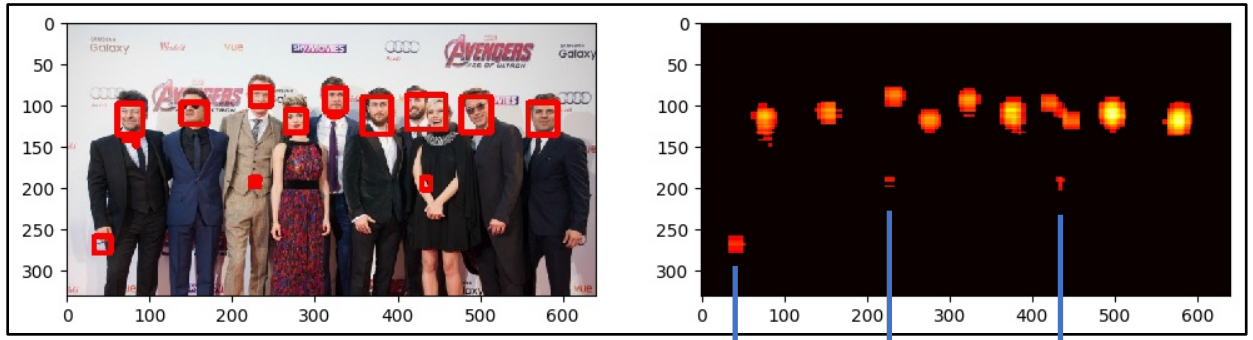
Part3
The testing results of facial recognition





If the background and full bodies exist, the classifier will still have some undesired recognitions.



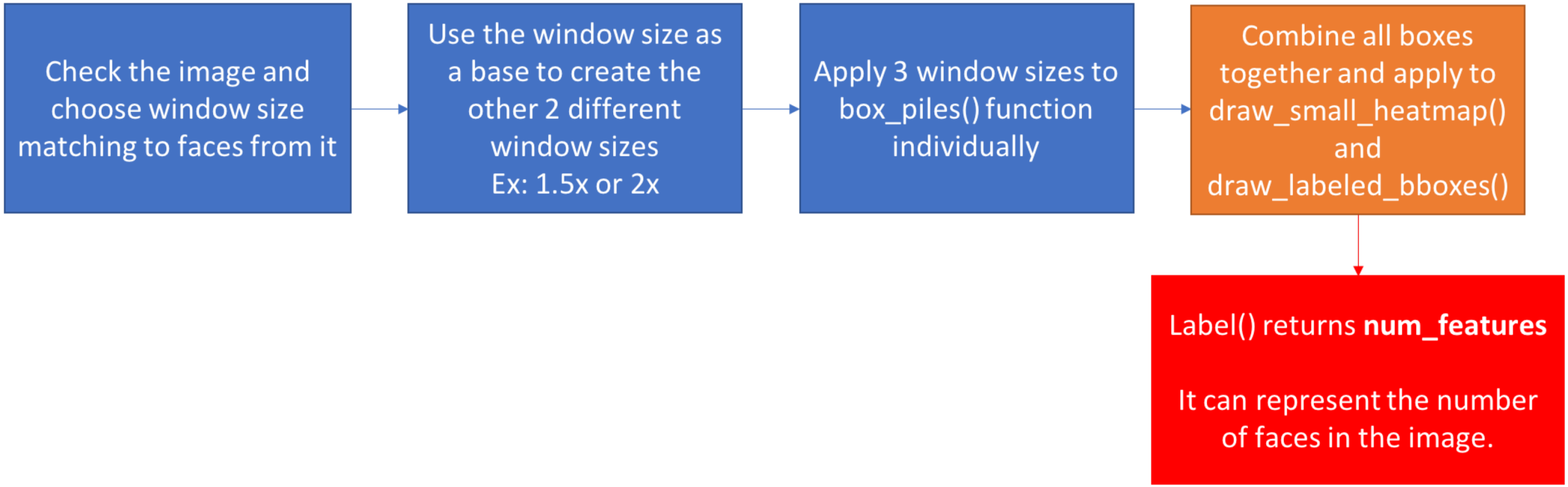


If the background and full bodies exist, the classifier will still have some undesired recognitions.

Part3
Images for crowd counting

* The color is in BGR instead of RGB mode.

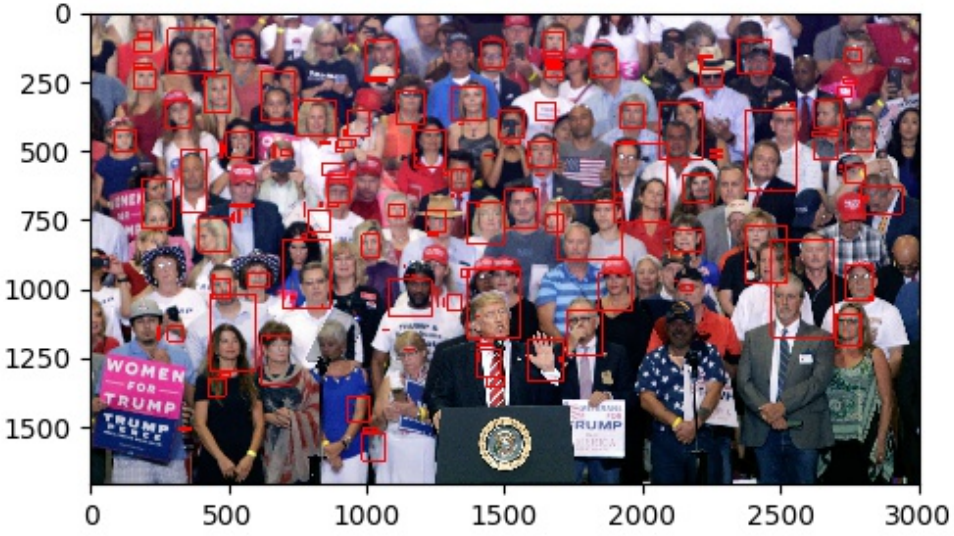
Part3
Crowd counting pipeline

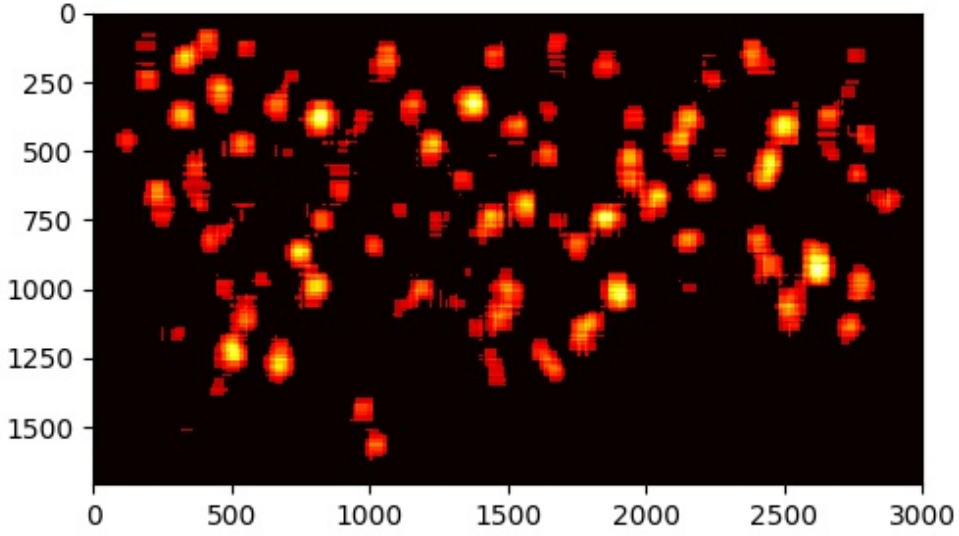


* Functions are all included in my Github repository

Part3
Crowd counting results

The image has 120 faces and the heat map recognizes 167 faces.

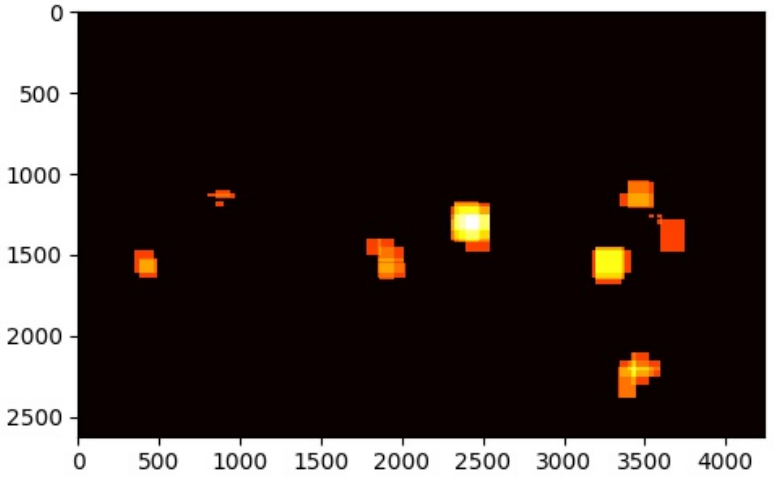




Threshold = 5,
Window size = 100x100, 150x150, 200x200,
Functions: draw_small_heat(), draw_small_heatmap()

Part3
Crowd counting results

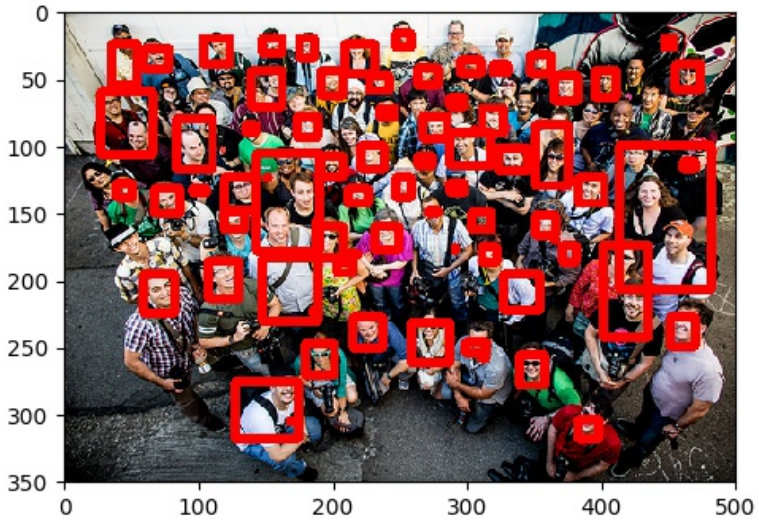
The image has ~40 faces and the heat map recognizes 8 faces.
This type of image included arms and dramatically different size of faces is not suitable for the method

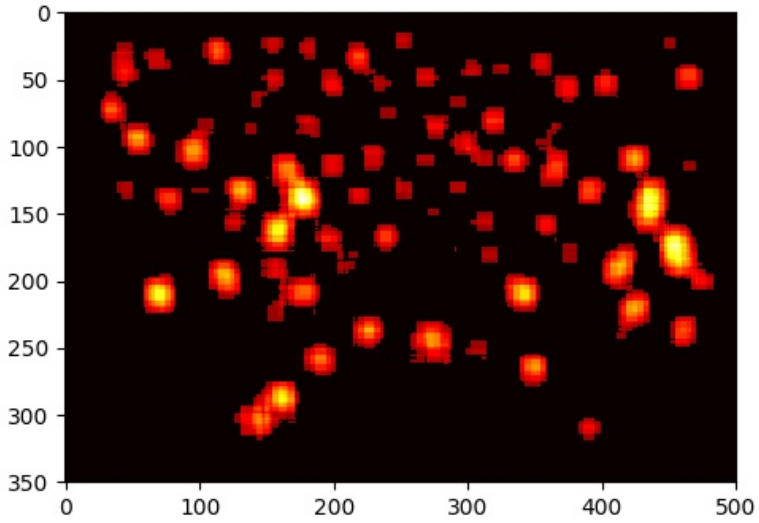


Threshold = 5,
Window size = 450x450, 500x500, 550x550,
Functions: draw_small_heat(), draw_small_heatmap()

Part3
Crowd counting results

The image has 84 faces and the heat map recognizes 77 faces.

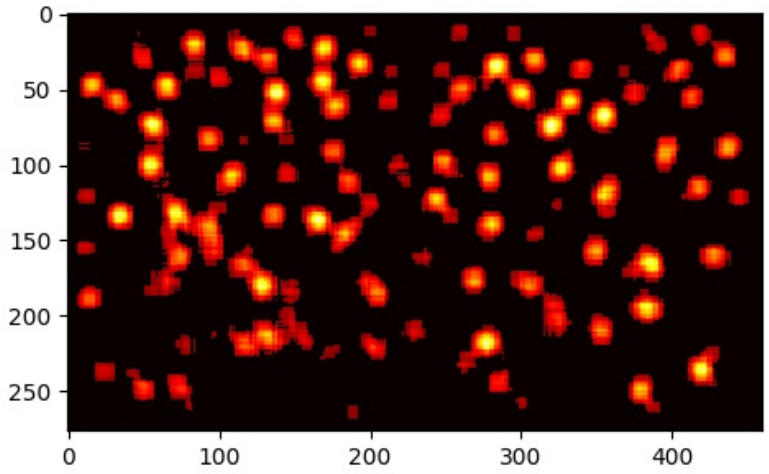




Threshold = 5,
Window size = 30x30, 45x45, 60x60,
Functions: draw_small_heat(), draw_small_heatmap()

Part3
Crowd counting results

The image has 110 faces and the heat map recognize 104 faces.



Threshold = 5,
Window size = 20x20, 25x25, 30x30,
Functions: draw_small_heat(), draw_small_heatmap()

Part4
Conclusion

- The model is based on only 5500 images training
- The model can perform well with clear faces existence without arms or full bodies.