

2019

CSE534 Project2 Report

VISUAL SEARCH SYSTEM

YICHEN WANG #50161143

GitHub link:

<https://github.com/yichenwa/CSE534P2.git>

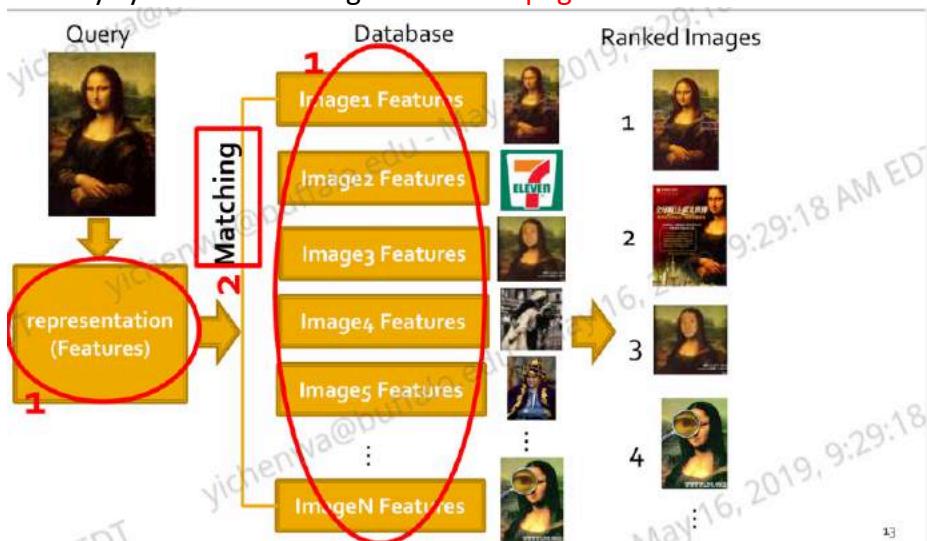
Demo link:

This is a video-demo:

<https://youtu.be/pjR3OqpKXWA>

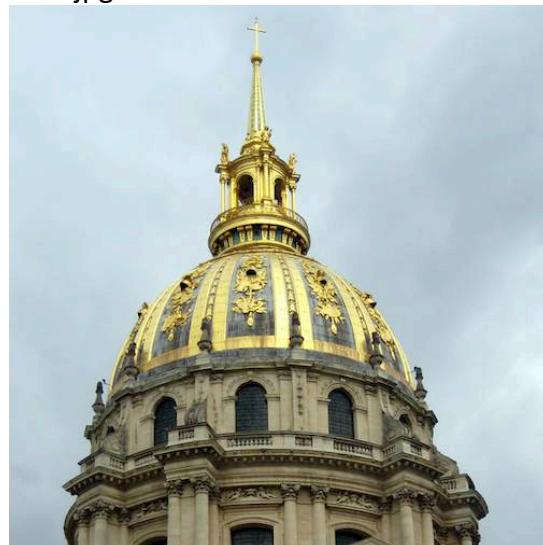
Description of my system:

The main idea of my system is following the slide 10 page 13:



1. Import sample image img1

- Import the sample image manually.
this is my sample image "01.jpg"



2. Go through all pictures in the folder.

- I use a variable called “**DATADIR**” be the path to folder “paris” on my laptop. You can change it to run the code on your machine.
- I use a variable called “**CATEGORIES**” be the sub folders under “/paris”.
- To save the running time, I resize the images into its $1/2\text{width} \times 1/2\text{height}$.

3. Image Representation

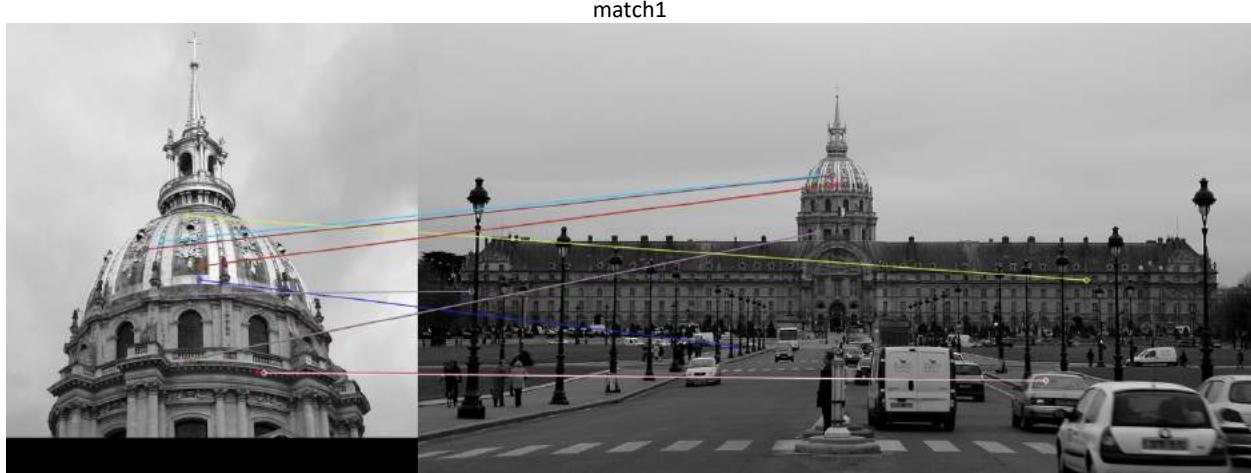
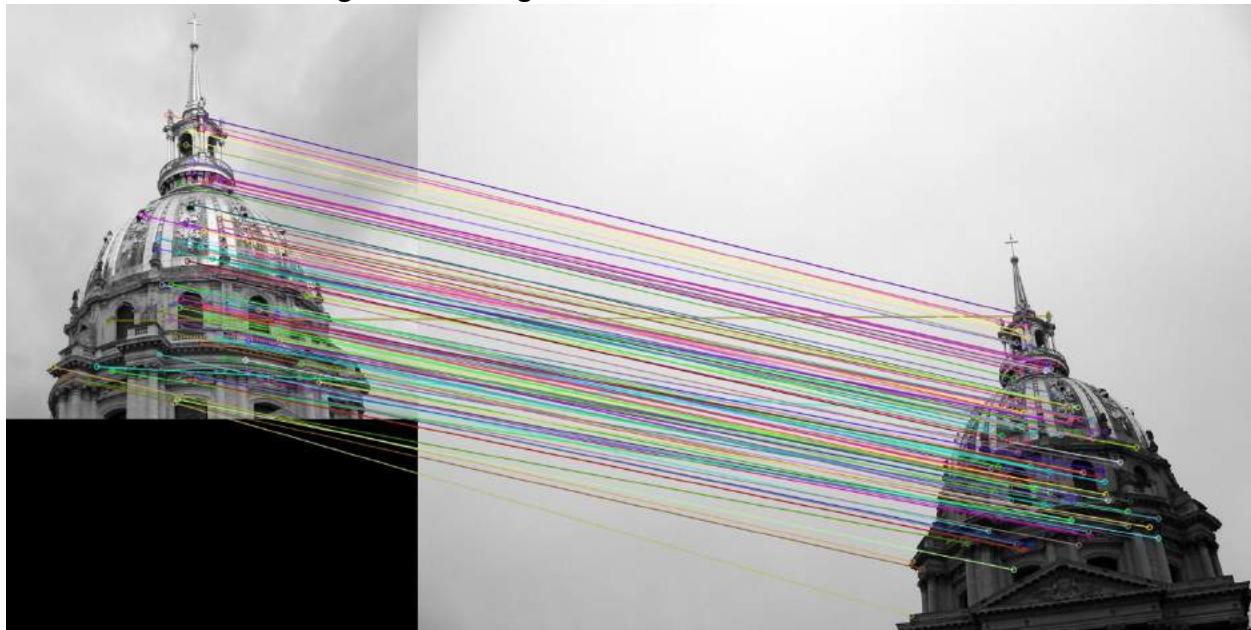
- I think local features might be better than global features, so I select to use SIFT descriptors to match local features.

4. Matching

- I used Brute-Force Matching with SIFT Descriptors.
- I followed the tutorial of OpenCV from this link: https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_matcher/py_matcher.html
- Under this step I found sometimes it cannot match the correct key point, I think there are several reasons:

The images are taken from different perspectives.

Different light in the image reflects different results.



match2



match3



match4

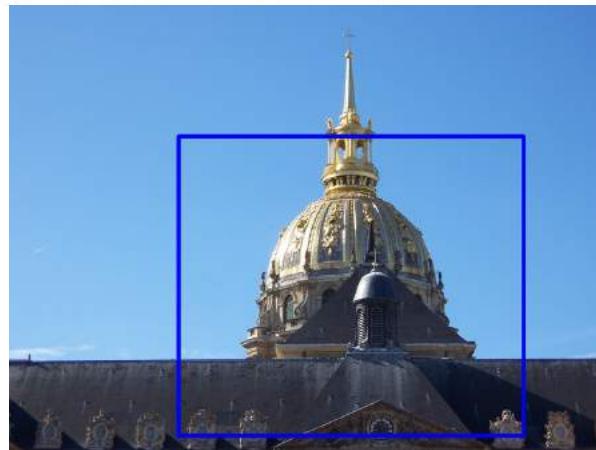
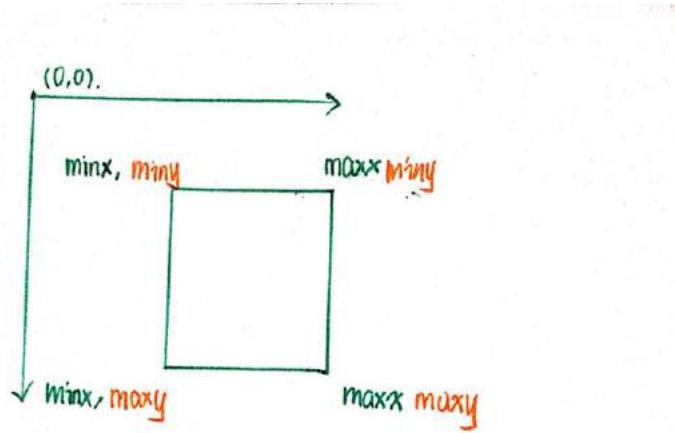
These are some outputs, you can find that some images work well (match1 and match 3), some just not very good (match 4).

The difference between them is **the size of list of lists as matches**. In my code the list's name is **“good”**. In the next part I will rank the images by the size of **“good”**. Those has large **“good”** means it is closer to the sample images.

In the step5, I will ignore the test image has less than **15** matched key points.

5. draw the rectangle.

- Get the key points in test image matched with sample image.
- Save them coordinates, get the **largest and smallest** x-coordinate (`max_x, min_x`), the largest and smallest y-coordinate (`max_y, min_y`).
- Then we can get the four points of rectangle and draw it by left-top point and right-bottom point.



6. ranking

Ranking the test images has more than 15 matched key points with test image.

- Save all images **“good”** size, its index in the folder, the box left top point and right bottom point in a list `l`, its folder's index in the `“/paris”`.
- Then sort the `l` by its first parameter (**“good”** size) select the last 10 of them.
- draw rectangle on the 10 images.

Quantitative results, mAP and time.

$$\text{precision} = \frac{\text{TruePositives}}{\text{TruePositives} + \text{FalsePositives}}$$

$$\text{recall} = \frac{\text{TruePositives}}{\text{TruePositives} + \text{FalseNegatives}}$$

$$\text{accuracy} = \frac{\text{TruePositives} + \text{TrueNegatives}}{\text{TruePositives} + \text{FalseNegatives} + \text{TrueNegatives} + \text{FalsePositives}}$$

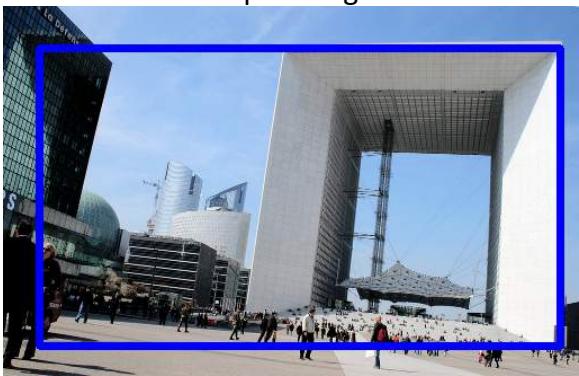
— · — · — · — · — · Class 0 -- <Defense>

Sample image



<2>

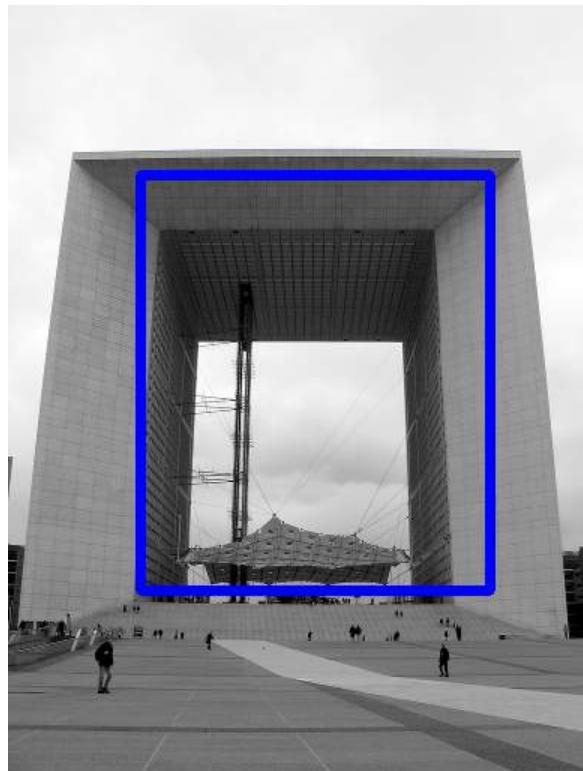
Output images



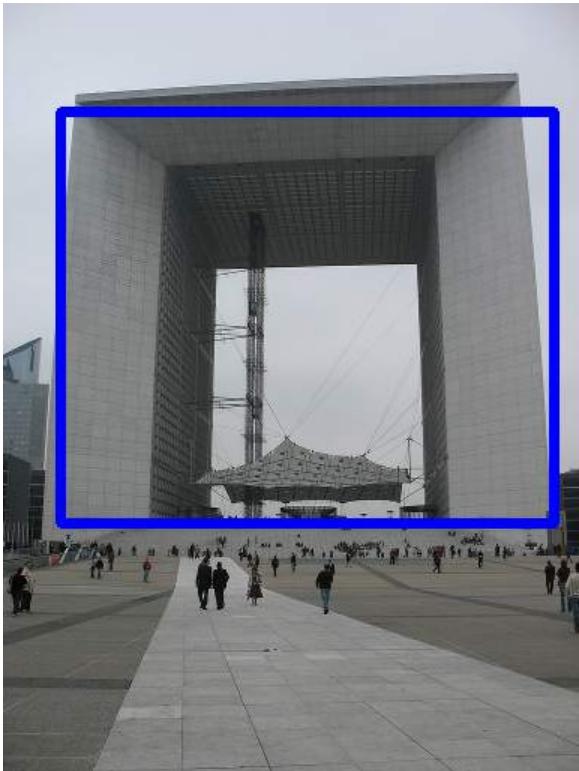
<1>



<3>



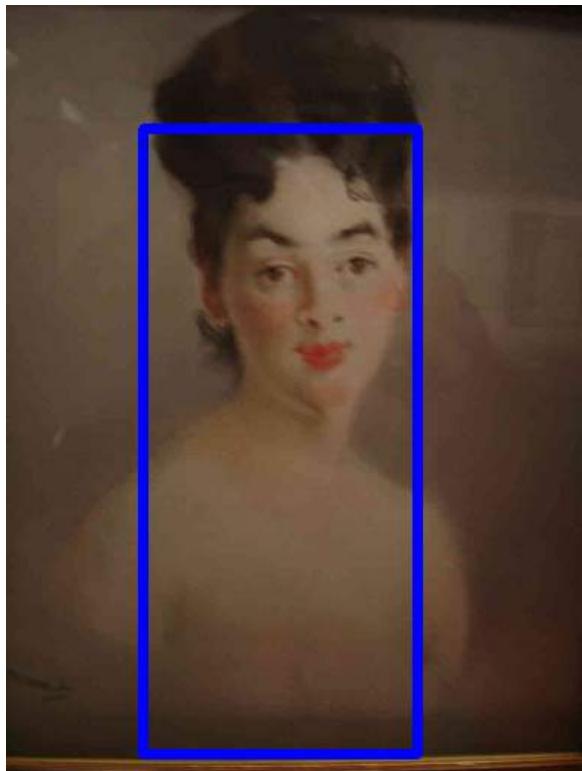
<5>



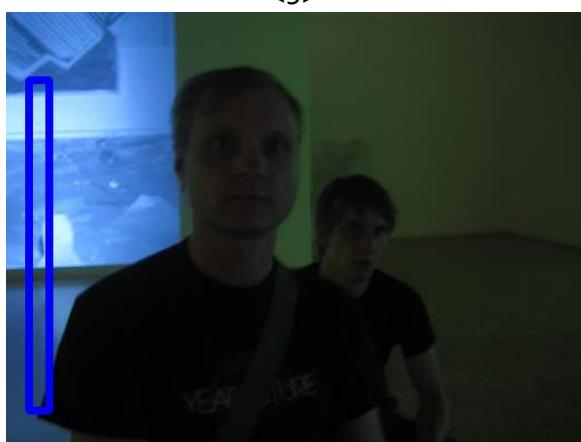
<4>



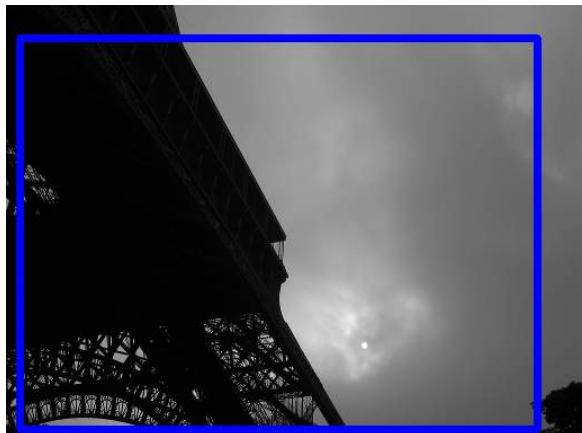
<6>



<7>



<9>



<8>



mAP and Time

```
534Project2 ×
↑ In 0 th class it has 381 images, gets 22 images contains sample images
↓ In 1 th class it has 147 images, gets 4 images contains sample images
→ In 2 th class it has 1497 images, gets 48 images contains sample images
← In 3 th class it has 452 images, gets 19 images contains sample images
↑ In 4 th class it has 267 images, gets 6 images contains sample images
↓ In 5 th class it has 445 images, gets 17 images contains sample images
→ In 6 th class it has 740 images, gets 31 images contains sample images
← In 7 th class it has 445 images, gets 9 images contains sample images
↑ In 8 th class it has 660 images, gets 22 images contains sample images
↓ In 9 th class it has 438 images, gets 14 images contains sample images
→ In 10 th class it has 380 images, gets 14 images contains sample images
← In 11 th class it has 551 images, gets 7 images contains sample images
Finish in 662 seconds

Process finished with exit code 0
```

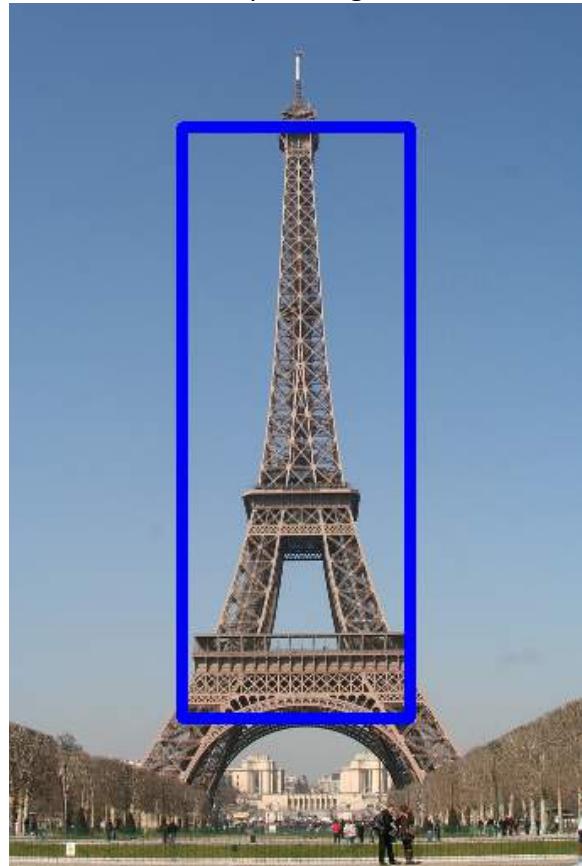
Defense								
Class defense size=381, True Positive=22, False Negative=358 TIME=662seconds								mAP
	class name	class size	False Positives	True Negatives	precision	recall	accuracy	
1	eiffel	147	4	143	0.846154	0.0578947	0.313093	
2	general	1497	48	1449	0.314286	0.0578947	0.7836974	
3	invalides	452	19	433	0.536585	0.0578947	0.546875	
4	louvre	267	6	261	0.785714	0.0578947	0.4374034	
5	moulinrouge	445	17	428	0.564103	0.0578947	0.5454545	
6	museedorsa	740	31	709	0.415094	0.0578947	0.6526786	
7	notredame	445	9	436	0.709677	0.0578947	0.5551515	
8	pantheon	660	22	638	0.5	0.0578947	0.6346154	
9	pompidou	438	14	424	0.611111	0.0578947	0.5452323	
10	sacrecoeur	380	14	366	0.611111	0.0578947	0.5105263	
11	triomphe	551	7	544	0.758621	0.0578947	0.6079484	0.557516

Class 1 -- <Eiffel>

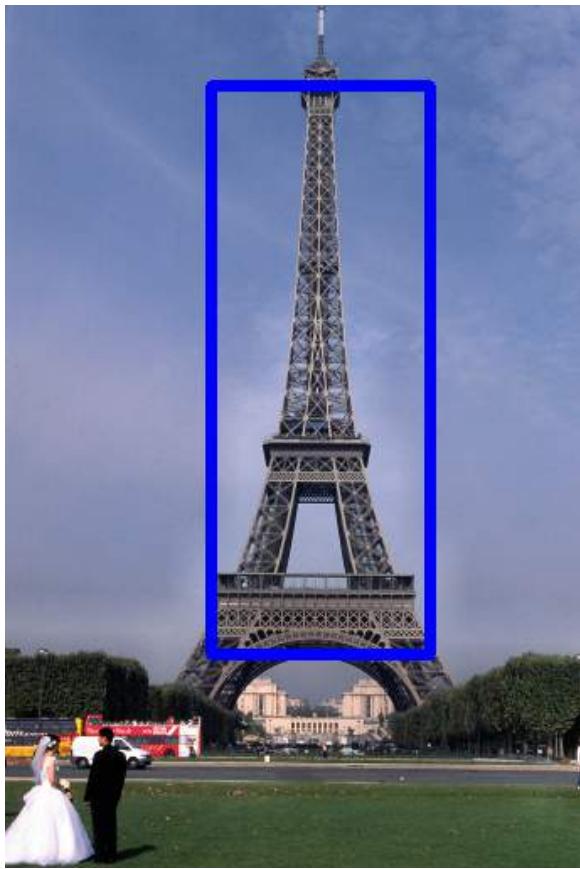
sample image



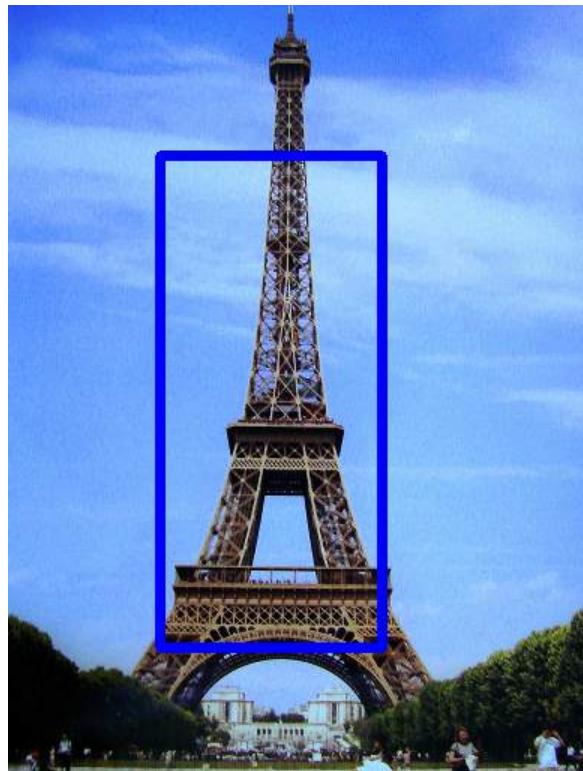
Output images



<1>



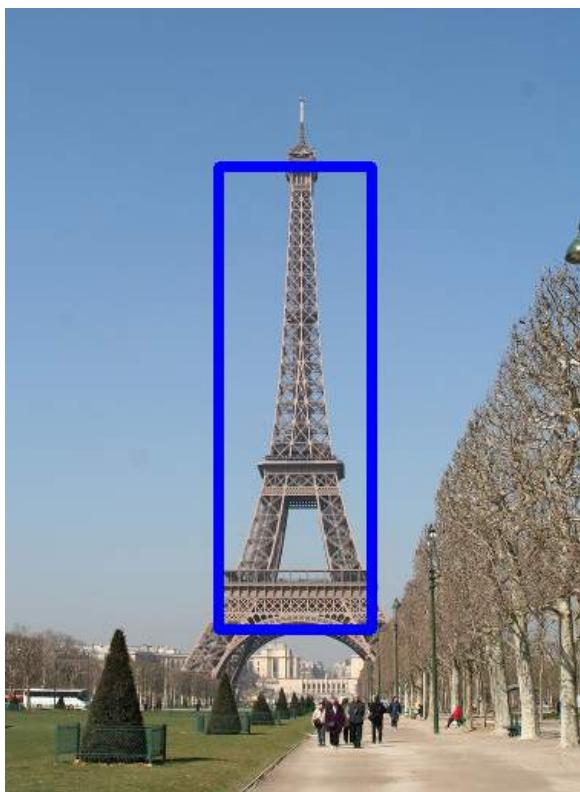
<2>



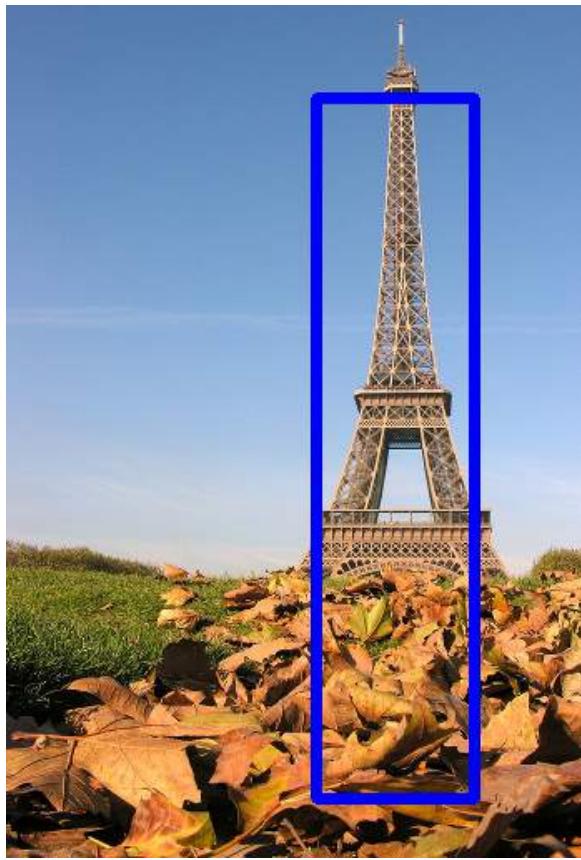
<4>



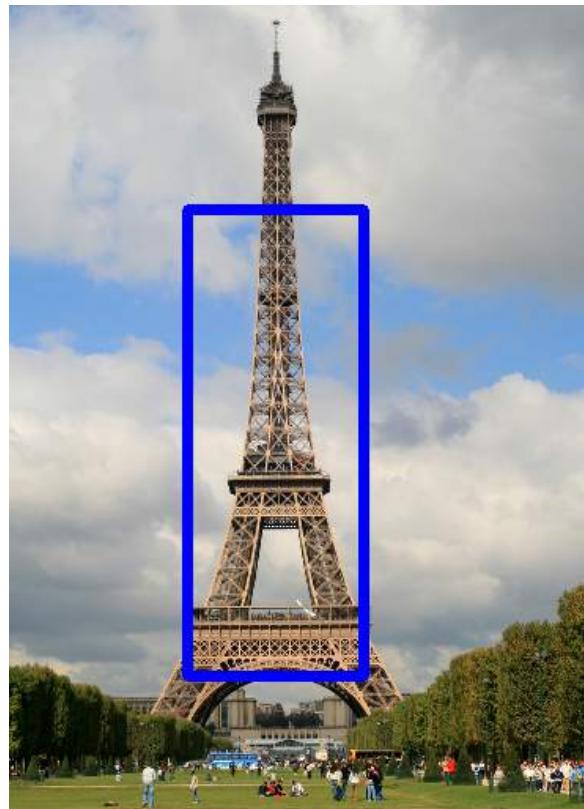
<5>



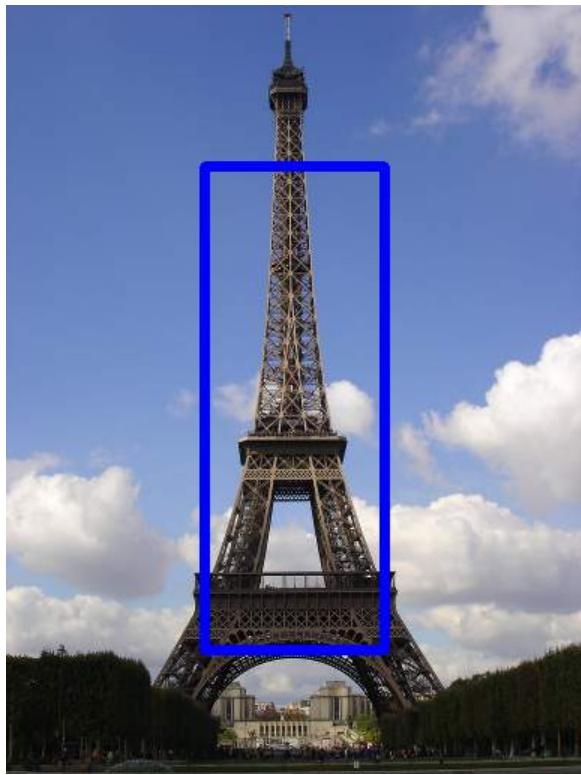
<3>



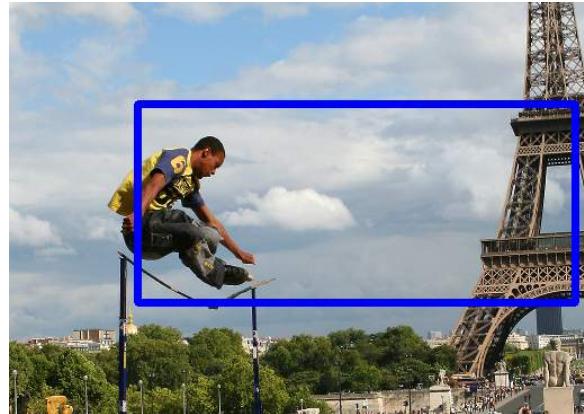
<6>



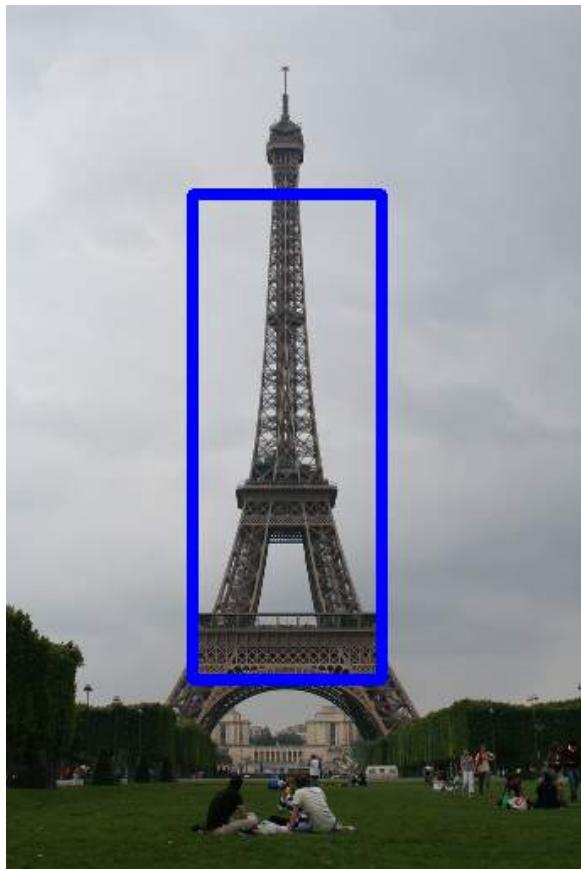
<8>



<7>



<9>



<10>

mAP and Time

```
534Project2 ×
: /Users/saber/PycharmProjects/534P1/venv/bin/python /Users/saber/PycharmPro
In 0 th class it has 381 images, gets 0 images contains sample images
In 1 th class it has 147 images, gets 9 images contains sample images
In 2 th class it has 1497 images, gets 45 images contains sample images
In 3 th class it has 452 images, gets 0 images contains sample images
In 4 th class it has 267 images, gets 0 images contains sample images
In 5 th class it has 445 images, gets 0 images contains sample images
In 6 th class it has 740 images, gets 0 images contains sample images
In 7 th class it has 445 images, gets 0 images contains sample images
In 8 th class it has 660 images, gets 0 images contains sample images
In 9 th class it has 438 images, gets 0 images contains sample images
In 10 th class it has 380 images, gets 0 images contains sample images
In 11 th class it has 551 images, gets 0 images contains sample images
Finish in 681 seconds
```

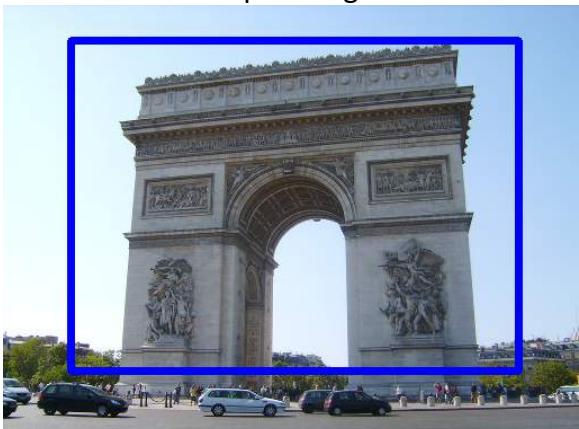
Eiffel								
Class eiffel size=147, True Positive=9, False Negative=138 TIME=681seconds								
	class name	class size	False Positives	True Negatives	precision	recall	accuracy	mAP
0	defense	381	0	381	1	0.0612245	0.7386364	0.7795764
2	general	1497	45	1452	0.166667	0.0612245	0.8886861	
3	invalides	452	0	452	1	0.0612245	0.769616	
4	louvre	267	0	267	1	0.0612245	0.6666667	
5	moulinrouge	445	0	445	1	0.0612245	0.7668919	
6	museedorsai	740	0	740	1	0.0612245	0.8444194	
7	notredame	445	0	445	1	0.0612245	0.7668919	
8	pantheon	660	0	660	1	0.0612245	0.8289963	
9	pompidou	438	0	438	1	0.0612245	0.7641026	
10	sacrecoeur	380	0	380	1	0.0612245	0.7381404	
11	triomphe	551	0	551	1	0.0612245	0.8022923	

Class 2 -- <general>

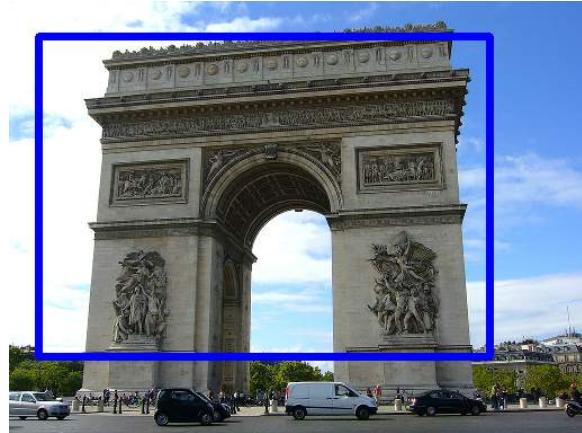
Sample image



Output images



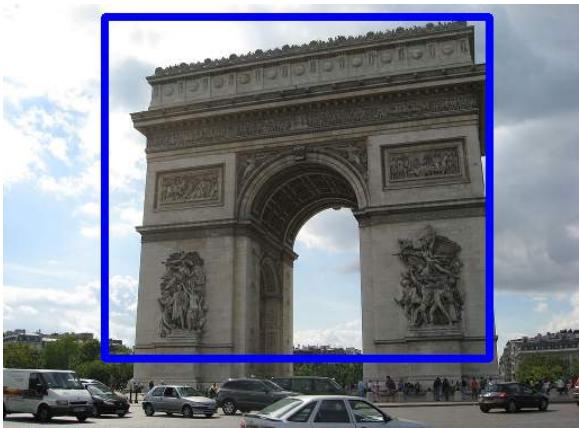
<1>



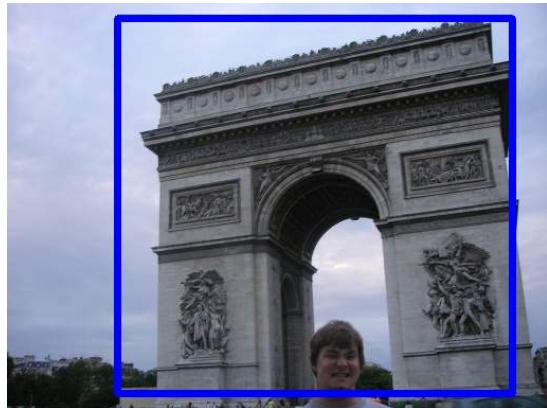
<3>



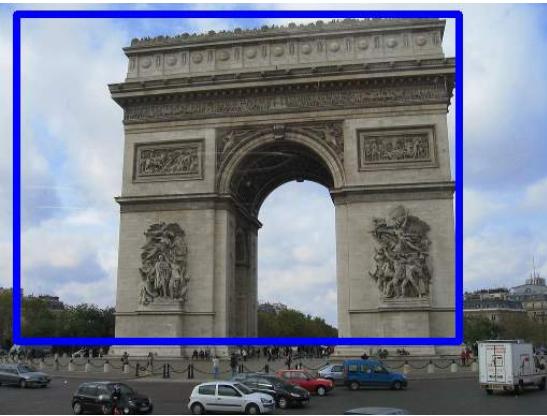
<4>



<2>



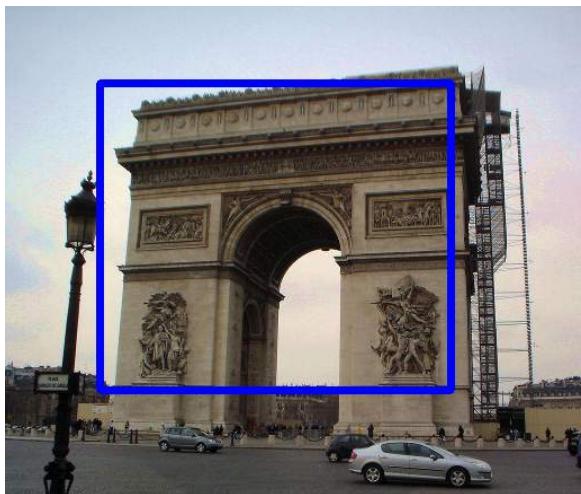
<5>



<6>



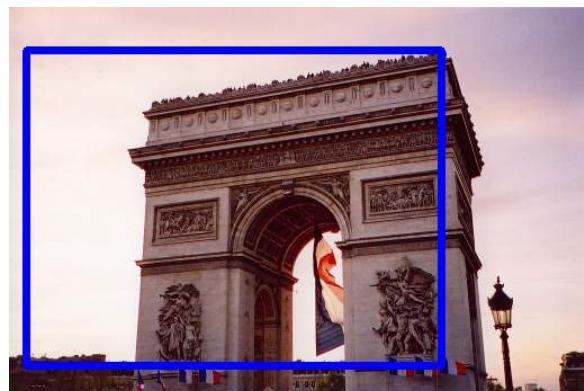
<7>



<8>



<9>



<10>

mAP and Time

```

534Project2 x
/Users/saber/PycharmProjects/534P1/venv/bin/python /Users/saber/PycharmProjects/534P1/534Project2.py
In 0 th class it has 381 images, gets 244 images contains sample images
In 1 th class it has 147 images, gets 87 images contains sample images
In 2 th class it has 1497 images, gets 954 images contains sample images
In 3 th class it has 452 images, gets 307 images contains sample images
In 4 th class it has 267 images, gets 169 images contains sample images
In 5 th class it has 445 images, gets 309 images contains sample images
In 6 th class it has 740 images, gets 417 images contains sample images
In 7 th class it has 445 images, gets 272 images contains sample images
In 8 th class it has 660 images, gets 376 images contains sample images
In 9 th class it has 438 images, gets 253 images contains sample images
In 10 th class it has 380 images, gets 251 images contains sample images
In 11 th class it has 551 images, gets 379 images contains sample images
Finish in 809 seconds
|
```

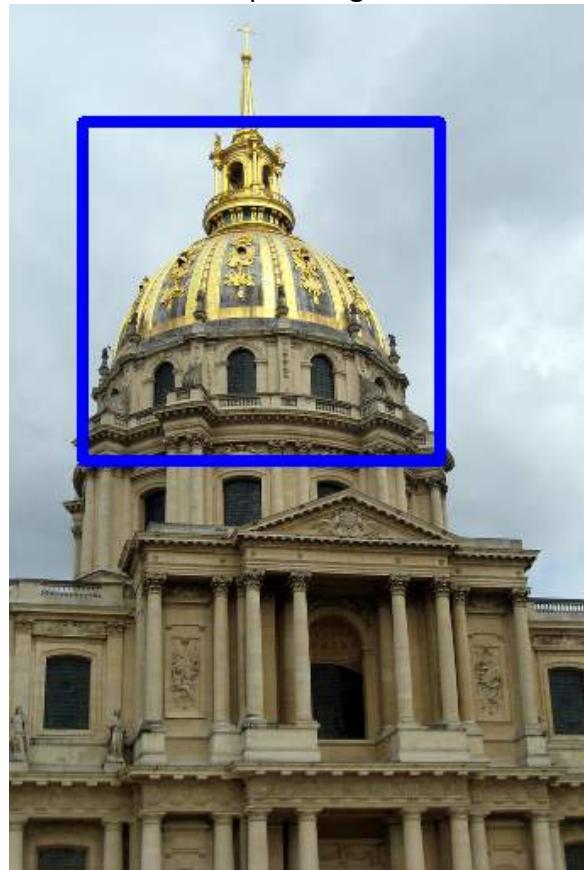
General								
Class general size=1497, True Positive=954, False Negative=543							TIME=809seconds	
	class name	class size	False Positives	True Negatives	precision	recall	accuracy	mAP
0	defense	381	244	137	0.796327	0.6372745	0.5809372	0.5781638
1	eiffel	147	87	60	0.916427	0.6372745	0.6167883	
3	invalides	452	307	145	0.756542	0.6372745	0.5638789	
4	louvre	267	169	98	0.84951	0.6372745	0.5963719	
5	moulinrouge	445	309	136	0.755344	0.6372745	0.561277	
6	museedorsay	740	417	323	0.695842	0.6372745	0.5708538	
7	notredame	445	272	173	0.77814	0.6372745	0.5803296	
8	pantheon	660	376	284	0.717293	0.6372745	0.5739453	
9	pompidou	438	253	185	0.790389	0.6372745	0.5886305	
10	sacrecoeur	380	251	129	0.791701	0.6372745	0.5769845	
11	triomphe	551	379	172	0.715679	0.6372745	0.5498047	

Class 3 -- <Invalides>

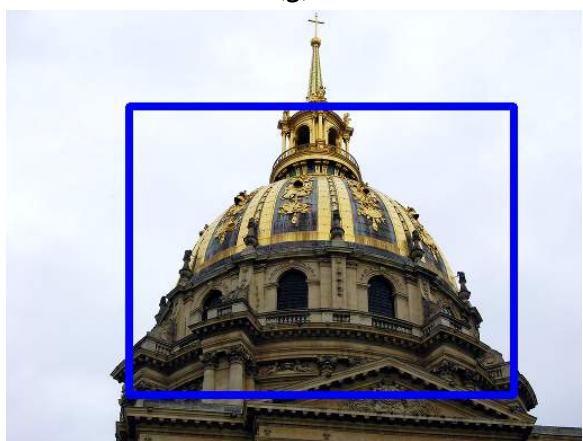
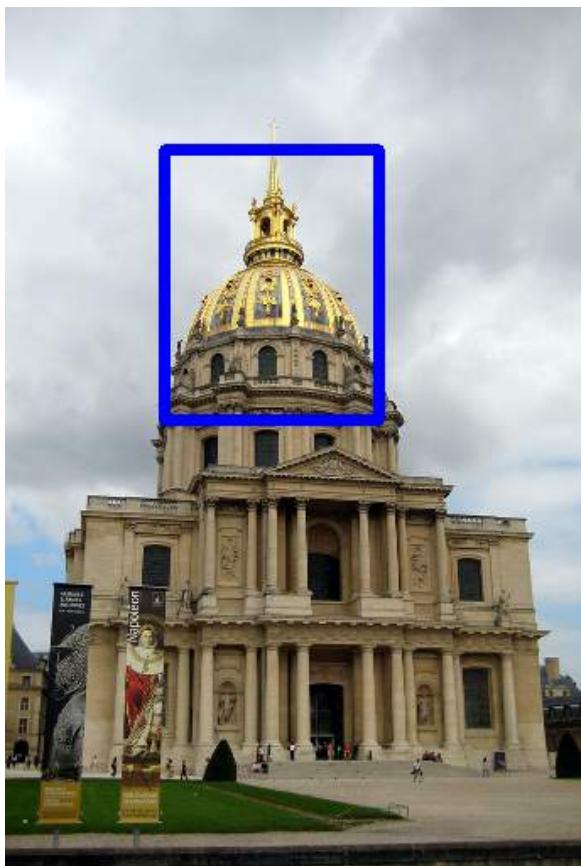
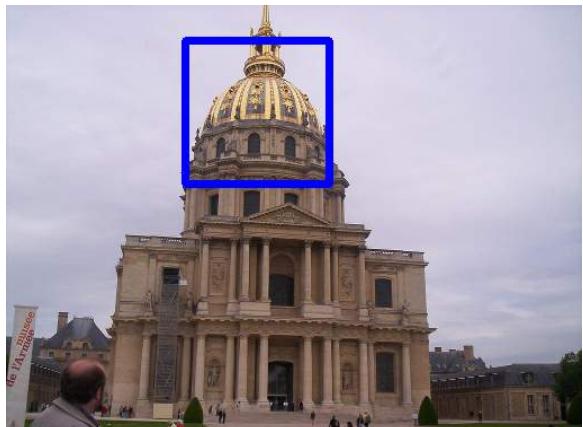
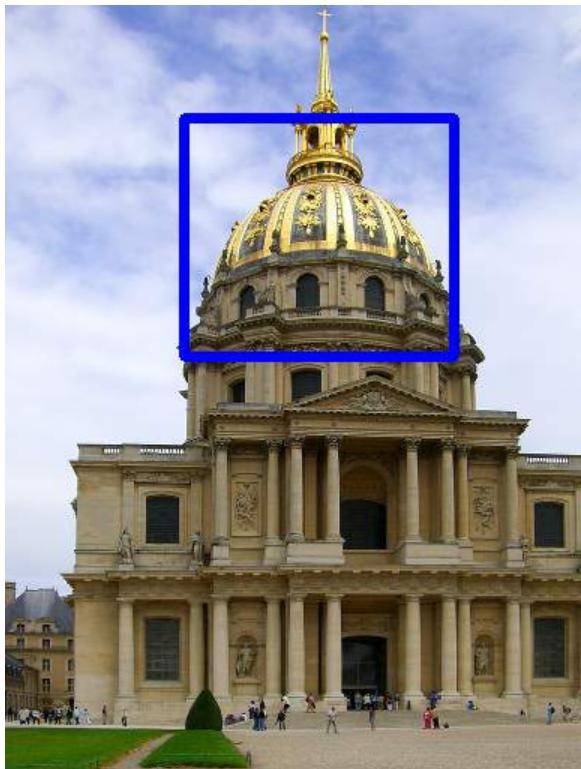
Sample Image

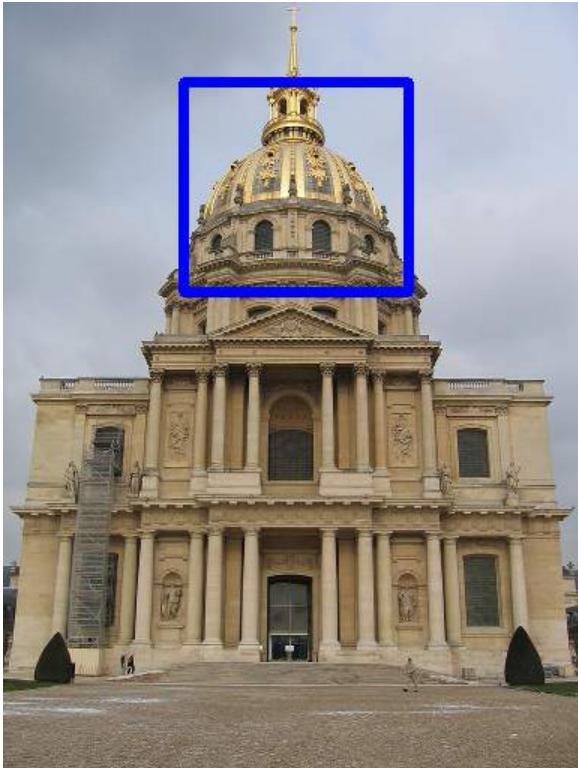


Output Images

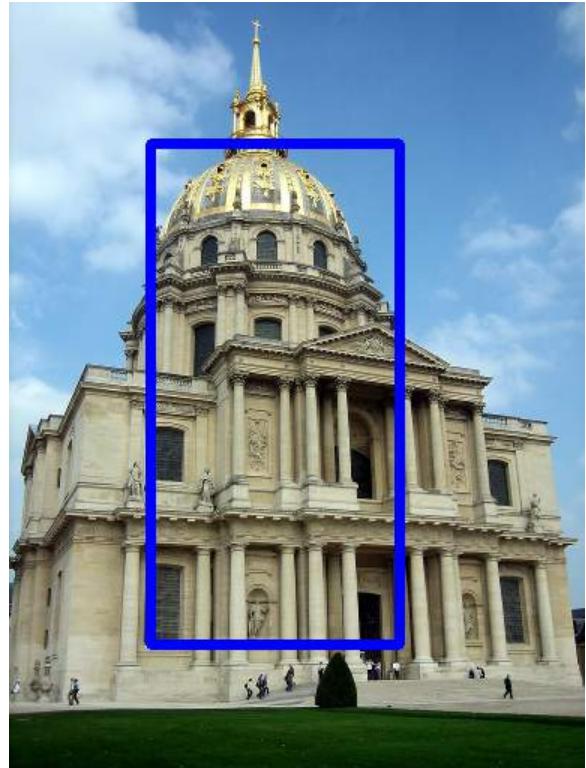


<1>





<7>



<9>



<8>



<10>

mAP and Time

```
534Project2 ×
/Users/saber/PycharmProjects/534P1/venv/bin/python /Users/saber/PycharmP
In 0 th class it has 381 images, gets 0 images contains sample images
In 1 th class it has 147 images, gets 0 images contains sample images
In 2 th class it has 1497 images, gets 2 images contains sample images
In 3 th class it has 452 images, gets 63 images contains sample images
In 4 th class it has 267 images, gets 0 images contains sample images
In 5 th class it has 445 images, gets 0 images contains sample images
In 6 th class it has 740 images, gets 0 images contains sample images
In 7 th class it has 445 images, gets 0 images contains sample images
In 8 th class it has 660 images, gets 0 images contains sample images
In 9 th class it has 438 images, gets 0 images contains sample images
In 10 th class it has 380 images, gets 0 images contains sample images
In 11 th class it has 551 images, gets 0 images contains sample images
Finish in 738 seconds

Process finished with exit code 0
```

Invalides								
Class Invalides size=452, True Positive=63, False Negative=389						TIME=738seconds		
	class name	class size	False Positives	True Negatives	precision	recall	accuracy	mAP
0	defense	381	0	381	1	0.1393805	0.5330132	
1	eiffel	147	0	147	1	0.1393805	0.3505843	
2	genral	1497	2	1495	0.969231	0.1393805	0.7993843	
4	louvre	267	0	267	1	0.1393805	0.4589708	
5	moulinrouge	445	0	445	1	0.1393805	0.5663322	
6	museedorsa	740	0	740	1	0.1393805	0.6736577	
7	notredame	445	0	445	1	0.1393805	0.5663322	
8	pantheon	660	0	660	1	0.1393805	0.6501799	
9	pompidou	438	0	438	1	0.1393805	0.5629213	
10	sacrecoeur	380	0	380	1	0.1393805	0.5324519	
11	triomphe	551	0	551	1	0.1393805	0.6121635	0.5732719

Class 4 -- <Louver>

Sample Image



Output Images



<1>



<2>



<3>

mAP and Time

```
534Project2 ×
/Users/saber/PycharmProjects/534P1/venv/bin/python /Users/saber/PycharmPro
In 0 th class it has 381 images, gets 0 images contains sample images
In 1 th class it has 147 images, gets 0 images contains sample images
In 2 th class it has 1497 images, gets 0 images contains sample images
In 3 th class it has 452 images, gets 0 images contains sample images
In 4 th class it has 267 images, gets 3 images contains sample images
In 5 th class it has 445 images, gets 0 images contains sample images
In 6 th class it has 740 images, gets 0 images contains sample images
In 7 th class it has 445 images, gets 0 images contains sample images
In 8 th class it has 660 images, gets 0 images contains sample images
In 9 th class it has 438 images, gets 0 images contains sample images
In 10 th class it has 380 images, gets 0 images contains sample images
In 11 th class it has 551 images, gets 0 images contains sample images
Finish in 640 seconds
```

Louvre								
Class louvre size=267, True Positive=3, False Negative=264 TIME=640seconds								
	class name	class size	False Positives	True Negatives	precision	recall	accuracy	mAP
0	defense	381	0	381	1	0.011236	0.5925926	
1	eiffel	147	0	147	1	0.011236	0.3623188	
2	genral	1497	0	1497	1	0.011236	0.8503401	
3	invalides	452	0	452	1	0.011236	0.6328234	
5	moulinrouge	445	0	445	1	0.011236	0.6292135	
6	museedorsay	740	0	740	1	0.011236	0.7378352	
7	notredame	445	0	445	1	0.011236	0.6292135	
8	pantheon	660	0	660	1	0.011236	0.7152104	
9	pompidou	438	0	438	1	0.011236	0.6255319	
10	sacrecoeur	380	0	380	1	0.011236	0.5919629	
11	triomphe	551	0	551	1	0.011236	0.6772616	0.6403913

Class 5 -- <Moulin rouge>

Sample Image



Output Image



<1>



<2>



<3>



<4>



<5>



<6>



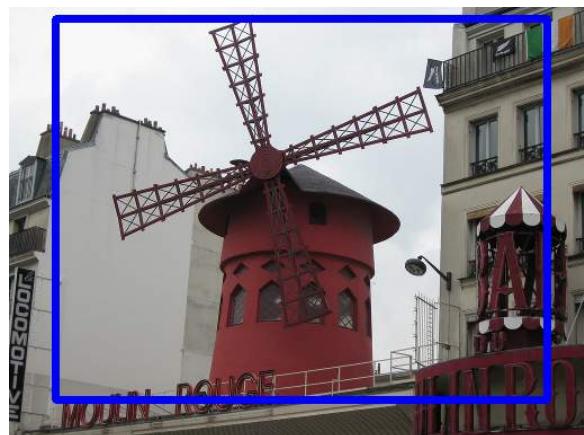
<7>



<8>



<9>



<10>

mAP and Time

```

534Project2 ×
/Users/saber/PycharmProjects/534P1/venv/bin/python /Users/saber/PycharmProjects/534P1/534Project2
In 0 th class it has 381 images, gets 4 images contains sample images
In 1 th class it has 147 images, gets 2 images contains sample images
In 2 th class it has 1497 images, gets 8 images contains sample images
In 3 th class it has 452 images, gets 2 images contains sample images
In 4 th class it has 267 images, gets 5 images contains sample images
In 5 th class it has 445 images, gets 151 images contains sample images
In 6 th class it has 740 images, gets 7 images contains sample images
In 7 th class it has 445 images, gets 3 images contains sample images
In 8 th class it has 660 images, gets 2 images contains sample images
In 9 th class it has 438 images, gets 10 images contains sample images
In 10 th class it has 380 images, gets 2 images contains sample images
In 11 th class it has 551 images, gets 2 images contains sample images
Finish in 820 seconds

Process finished with exit code 0

```

Moulinrouge								
Class moulinrouge size=445, True Positive=151, False Negative=294 TIME=820seconds								
	class name	class size	False Positives	True Negatives	precision	recall	accuracy	mAP
0	defense	381	4	377	0.974194	0.3393258	0.6392252	
1	eiffel	147	2	145	0.986928	0.3393258	0.5	
2	genral	1497	8	1489	0.949686	0.3393258	0.8444902	
3	invalides	452	2	450	0.986928	0.3393258	0.6700111	
4	louvre	267	5	262	0.967949	0.3393258	0.5800562	
6	museedorsay	740	7	733	0.955696	0.3393258	0.7459916	
7	notredame	445	3	442	0.980519	0.3393258	0.6662921	
8	pantheon	660	2	658	0.986928	0.3393258	0.7321267	
9	pompidou	438	10	428	0.937888	0.3393258	0.6557191	
10	sacrecoeur	380	2	378	0.986928	0.3393258	0.6412121	
11	triomphe	551	2	549	0.986928	0.3393258	0.7028112	0.6707214

Class 6 -- <Museedorsay>

Sample Image



Output Image



<1>



<2>



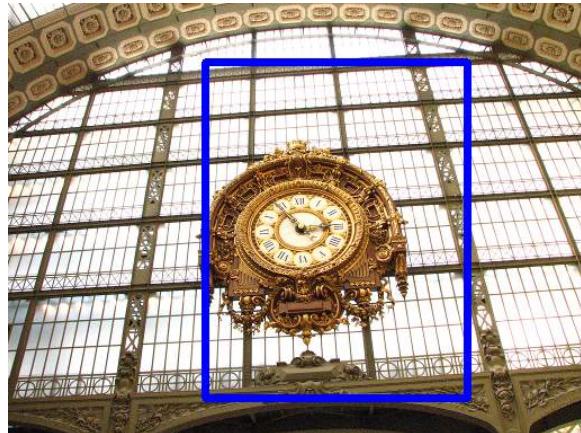
<3>



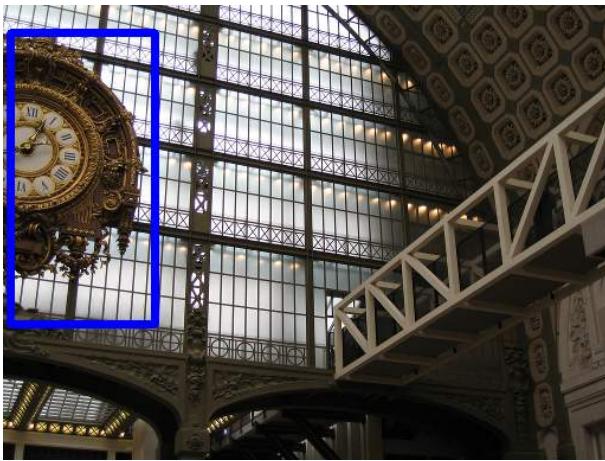
<5>



<4>



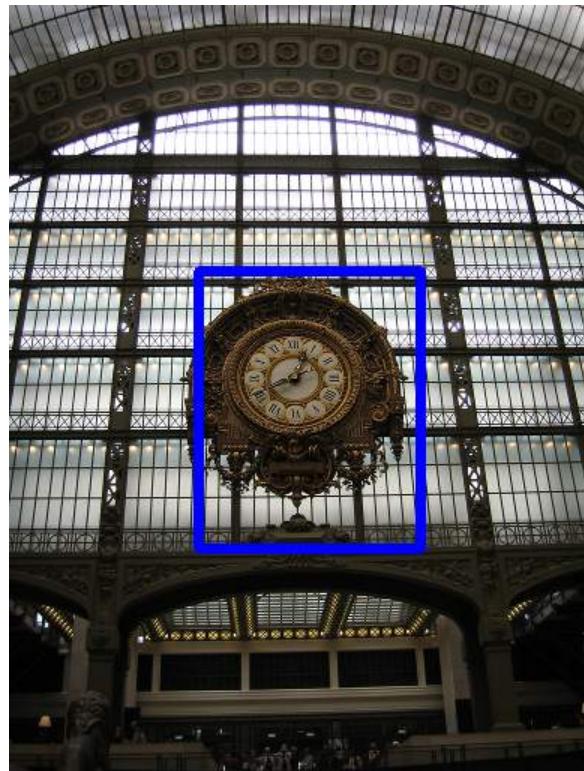
<6>



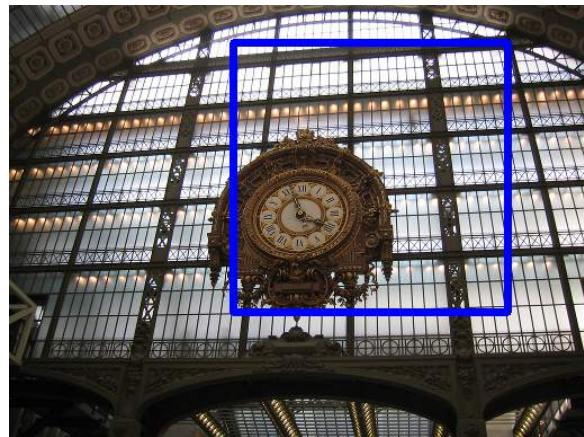
<7>



<8>



<9>



<10>

mAP and Time

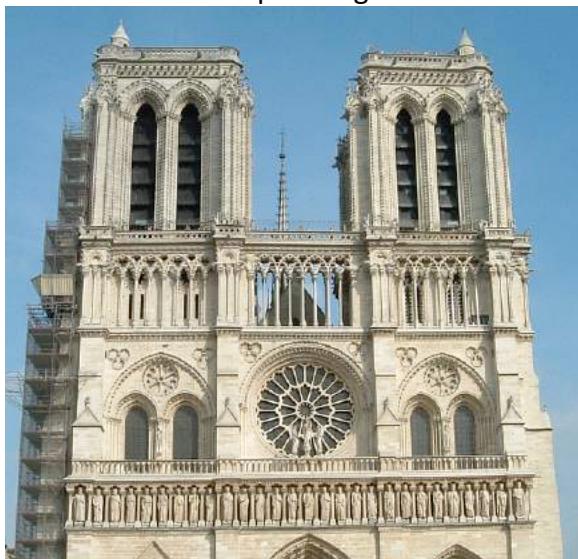
```
/Users/saber/PycharmProjects/534P1/venv/bin/python /Users/saber/PycharmProjects/534P1/534Project2.py
In 0 th class it has 381 images, gets 23 images contains sample images
In 1 th class it has 147 images, gets 6 images contains sample images
In 2 th class it has 1497 images, gets 74 images contains sample images
In 3 th class it has 452 images, gets 29 images contains sample images
In 4 th class it has 267 images, gets 14 images contains sample images
In 5 th class it has 445 images, gets 38 images contains sample images
In 6 th class it has 740 images, gets 70 images contains sample images
In 7 th class it has 445 images, gets 19 images contains sample images
In 8 th class it has 660 images, gets 48 images contains sample images
In 9 th class it has 438 images, gets 28 images contains sample images
In 10 th class it has 380 images, gets 18 images contains sample images
In 11 th class it has 551 images, gets 20 images contains sample images
Finish in 998 seconds
```

Process finished with exit code 0

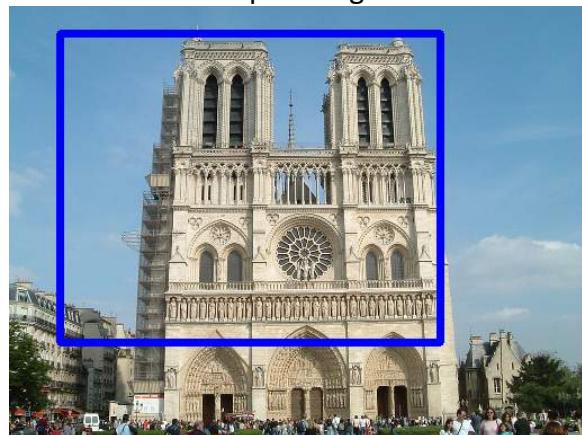
Museedorsay								
Class museedorsay size=740, True Positive=70, False Negative=670 TIME=998seconds								
	class name	class size	False Positives	True Negatives	precision	recall	accuracy	mAP
0	defense	381	23	358	0.752688	0.0945946	0.381802	
1	eiffel	147	6	141	0.921053	0.0945946	0.2378805	
2	genral	1497	74	1423	0.486111	0.0945946	0.6674117	
3	invalides	452	29	423	0.707071	0.0945946	0.4135906	
4	louvre	267	14	253	0.833333	0.0945946	0.3207547	
5	moulinrouge	445	38	407	0.648148	0.0945946	0.4025316	
7	notredame	445	19	426	0.786517	0.0945946	0.4185654	
8	pantheon	660	48	612	0.59322	0.0945946	0.4871429	
9	pompidou	438	28	410	0.714286	0.0945946	0.4074703	
10	sacrecoeur	380	18	362	0.795455	0.0945946	0.3857143	
11	triomphe	551	20	531	0.777778	0.0945946	0.4655306	0.4171268

Class 7 -- <Notredame>

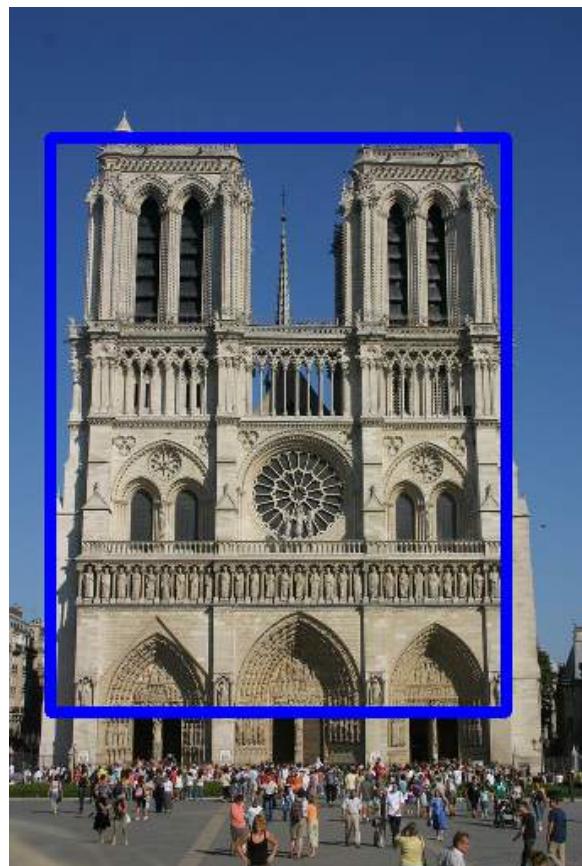
Sample Image



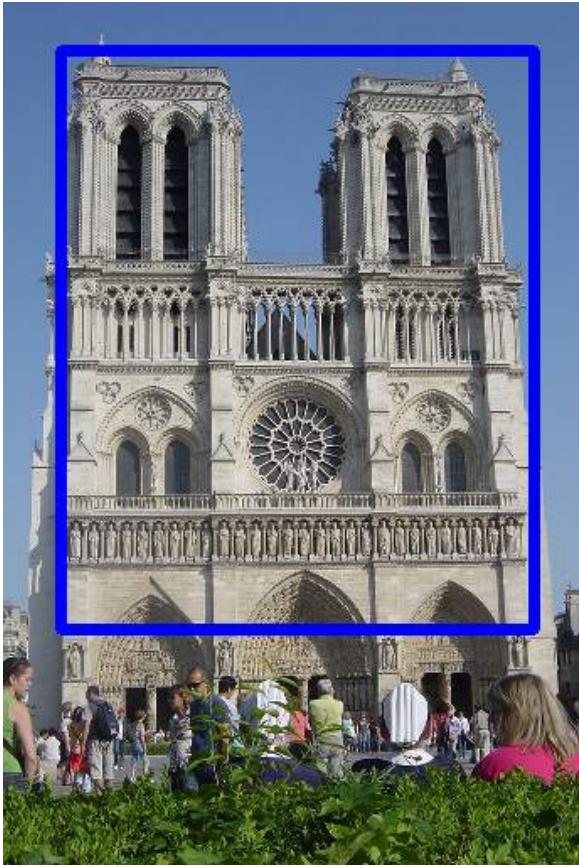
Output Images



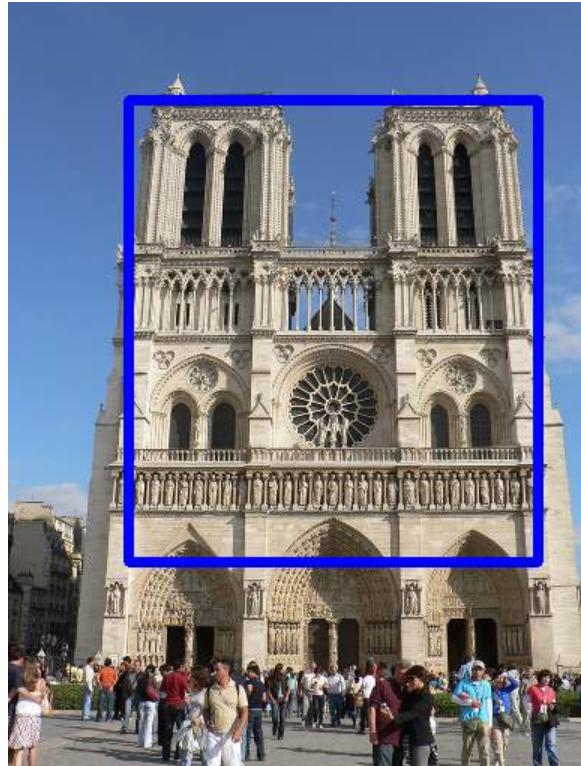
<1>



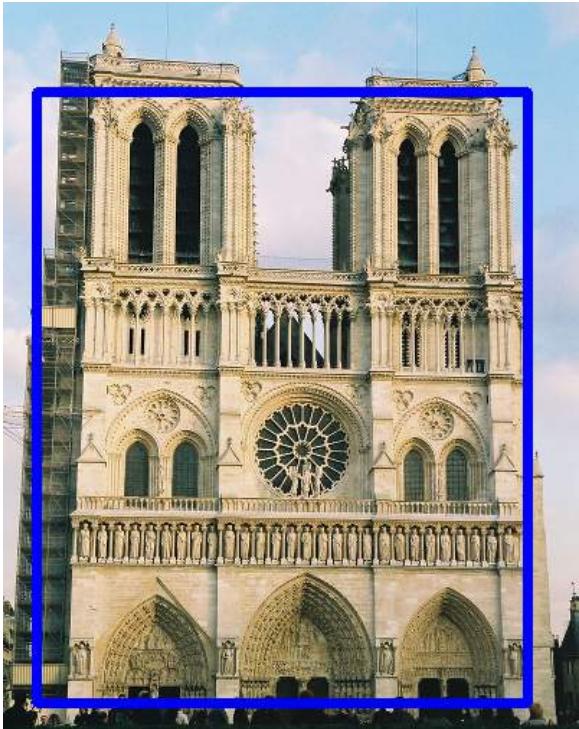
<2>



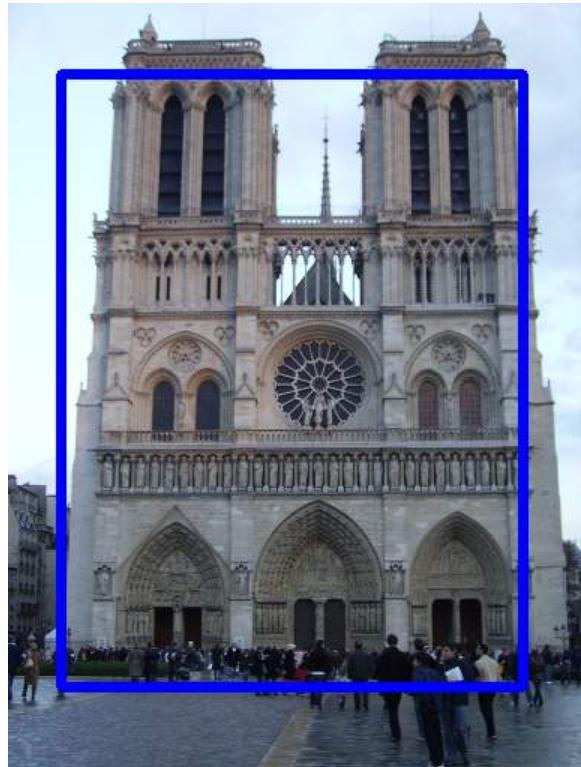
<3>



<5>



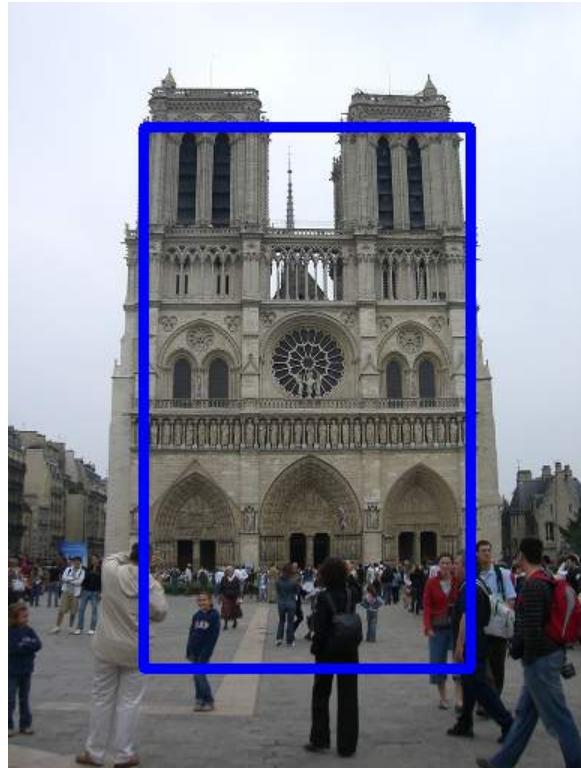
<4>



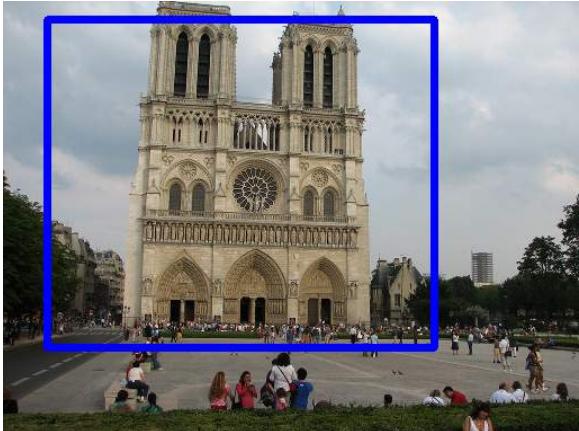
<6>



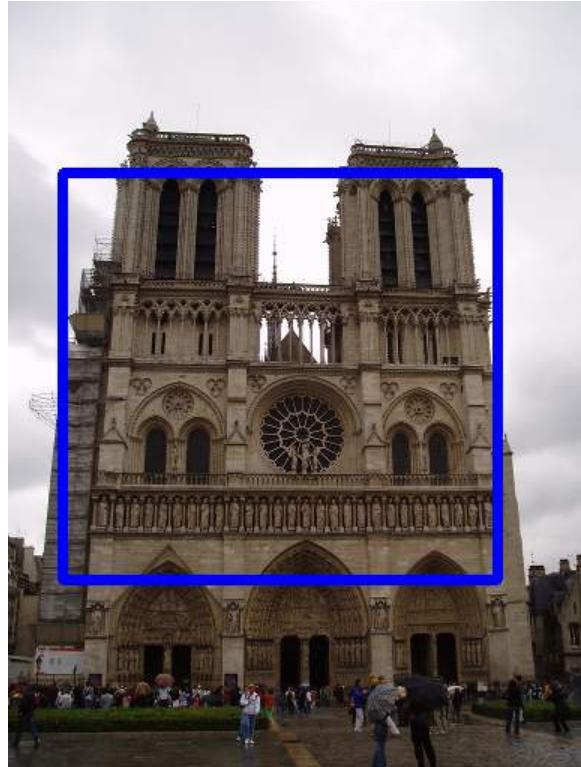
<7>



<9>



<8>



<10>

mAP and Time

```
Run: 534Project2 ×
▶ ⏚ ⏴ ⏵ ⏷ ⏸ ⏹ ⏺ ⏻
/Users/saber/PycharmProjects/534P1/venv/bin/python /Users/saber/PycharmProjects/534P1/534Project2.py
In 0 th class it has 381 images, gets 9 images contains sample images
In 1 th class it has 147 images, gets 4 images contains sample images
In 2 th class it has 1497 images, gets 23 images contains sample images
In 3 th class it has 452 images, gets 11 images contains sample images
In 4 th class it has 267 images, gets 5 images contains sample images
In 5 th class it has 445 images, gets 9 images contains sample images
In 6 th class it has 740 images, gets 15 images contains sample images
In 7 th class it has 445 images, gets 88 images contains sample images
In 8 th class it has 660 images, gets 17 images contains sample images
In 9 th class it has 438 images, gets 13 images contains sample images
In 10 th class it has 380 images, gets 10 images contains sample images
In 11 th class it has 551 images, gets 9 images contains sample images
Finish in 879 seconds

Process finished with exit code 0
```

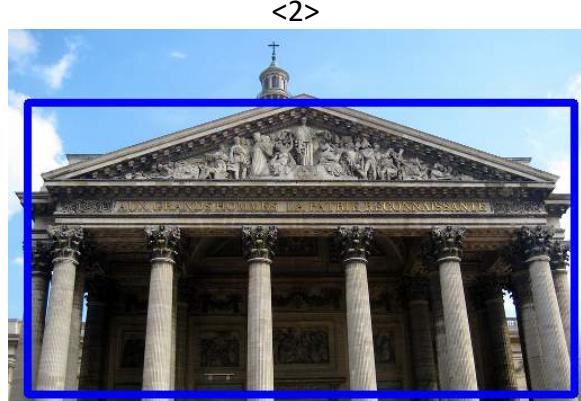
Notredame								
Class notredame size=445, True Positive=88, False Negative=357 TIME=879seconds								
	class name	class size	False Positives	True Negatives	precision	recall	accuracy	mAP
0	defense	381	9	372	0.907216	0.1977528	0.5569007	
1	eiffel	147	4	143	0.956522	0.1977528	0.3902027	
2	genral	1497	23	1474	0.792793	0.1977528	0.8043254	
3	invalides	452	11	441	0.888889	0.1977528	0.5897436	
4	louvre	267	5	262	0.946237	0.1977528	0.491573	
5	moulinrouge	445	9	436	0.907216	0.1977528	0.588764	
6	museedorsay	740	15	725	0.854369	0.1977528	0.6860759	
8	pantheon	660	17	643	0.838095	0.1977528	0.6615385	
9	pompidou	438	13	425	0.871287	0.1977528	0.580974	
10	sacrecoeur	380	10	370	0.897959	0.1977528	0.5551515	
11	triomphe	551	9	542	0.907216	0.1977528	0.6325301	0.5943436

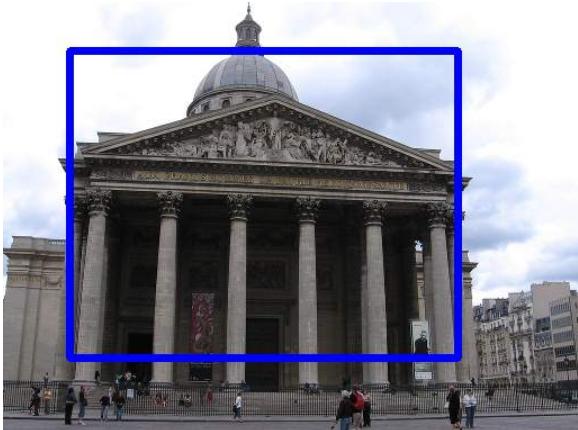
Class 8 -- <Pantheon>

Sample Image



Output Images

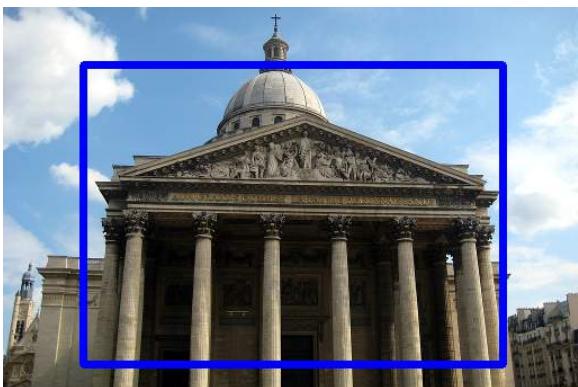




<4>



<8>



<5>



<9>



<6>



<10>



<7>

mAP and Time

```
/Users/saber/PycharmProjects/534P1/venv/bin/python /Users/saber/PycharmProjects/534P1/534Project2.py
In 0 th class it has 381 images, gets 140 images contains sample images
In 1 th class it has 147 images, gets 47 images contains sample images
In 2 th class it has 1497 images, gets 521 images contains sample images
In 3 th class it has 452 images, gets 188 images contains sample images
In 4 th class it has 267 images, gets 96 images contains sample images
In 5 th class it has 445 images, gets 211 images contains sample images
In 6 th class it has 740 images, gets 243 images contains sample images
In 7 th class it has 445 images, gets 153 images contains sample images
In 8 th class it has 660 images, gets 317 images contains sample images
In 9 th class it has 438 images, gets 158 images contains sample images
In 10 th class it has 380 images, gets 133 images contains sample images
In 11 th class it has 551 images, gets 127 images contains sample images
Finish in 849 seconds
```

Process finished with exit code 0

Pantheon								
Class pantheon size=660, True Positive=317, False Negative=343 TIME=849seconds								
	class name	class size	False Positives	True Negatives	precision	recall	accuracy	mAP
0	defense	381	140	241	0.693654	0.480303	0.5360231	
1	eiffel	147	47	100	0.870879	0.480303	0.5167286	
2	genral	1497	521	976	0.378282	0.480303	0.5994437	
3	invalides	452	188	264	0.627723	0.480303	0.522482	
4	louvre	267	96	171	0.767554	0.480303	0.5264293	
5	moulinrouge	445	211	234	0.600379	0.480303	0.4986425	
6	museedorsay	740	243	497	0.566071	0.480303	0.5814286	
7	notredame	445	153	292	0.674468	0.480303	0.5511312	
9	pompidou	438	158	280	0.667368	0.480303	0.5437158	
10	sacrecoeur	380	133	247	0.704444	0.480303	0.5423077	
11	triomphe	551	127	424	0.713964	0.480303	0.611891	0.5482021

Class 9 -- <Pompidou>

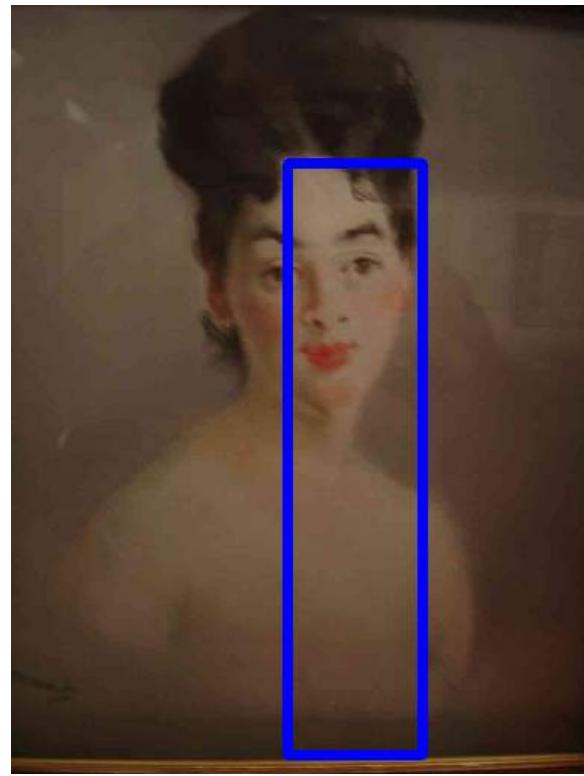
Sample Image



Output Images



<1>



<2>

mAP and Time

```

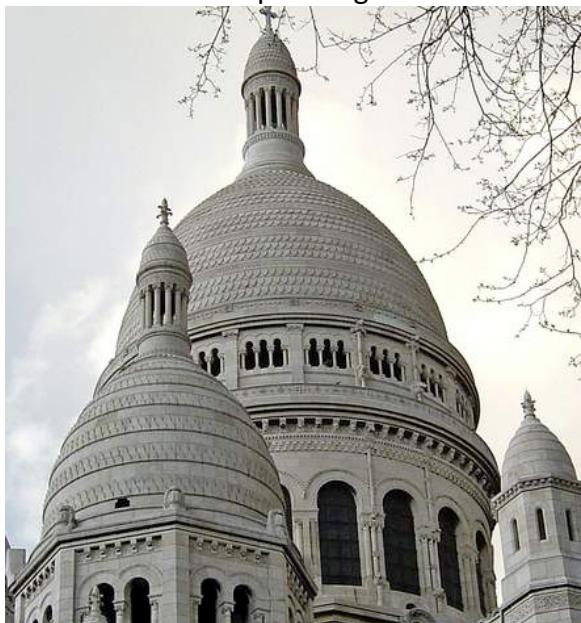
↑ /Users/saber/PycharmProjects/534P1/venv/bin/python /Users/saber/PycharmProjects/534P1/534Project2.py
In 0 th class it has 381 images, gets 0 images contains sample images
↓ In 1 th class it has 147 images, gets 0 images contains sample images
≡ In 2 th class it has 1497 images, gets 0 images contains sample images
≡ In 3 th class it has 452 images, gets 0 images contains sample images
≡ In 4 th class it has 267 images, gets 0 images contains sample images
≡ In 5 th class it has 445 images, gets 0 images contains sample images
≡ In 6 th class it has 740 images, gets 1 images contains sample images
≡ In 7 th class it has 445 images, gets 0 images contains sample images
≡ In 8 th class it has 660 images, gets 0 images contains sample images
≡ In 9 th class it has 438 images, gets 1 images contains sample images
≡ In 10 th class it has 380 images, gets 0 images contains sample images
≡ In 11 th class it has 551 images, gets 0 images contains sample images
Finish in 729 seconds

```

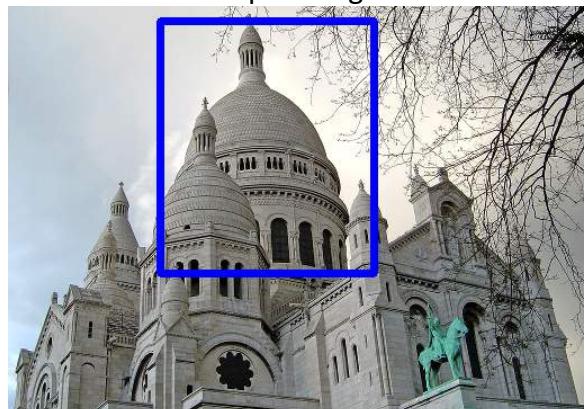
Pompidou								
Class pompidou size=438, True Positive=1, False Negative= 437						TIME=729seconds		
	class name	class size	False Positives	True Negatives	precision	recall	accuracy	mAP
0	defense	381	0	381	1	0.0022831	0.4664225	
1	eiffel	147	0	147	1	0.0022831	0.2529915	
2	genral	1497	0	1497	1	0.0022831	0.7741602	
3	invalides	452	0	452	1	0.0022831	0.5089888	
4	louvre	267	0	267	1	0.0022831	0.3801418	
5	moulinrouge	445	0	445	1	0.0022831	0.5050963	
6	museedorsay	740	1	739	0.984375	0.0022831	0.6281834	
7	notredame	445	0	445	1	0.0022831	0.5050963	
8	pantheon	660	0	660	1	0.0022831	0.6020036	
10	sacrecoeur	380	0	380	1	0.0022831	0.4657702	
11	triomphe	551	0	551	1	0.0022831	0.5581395	0.5133631

Class 10 -- <Sacrecoeur>

Sample Image



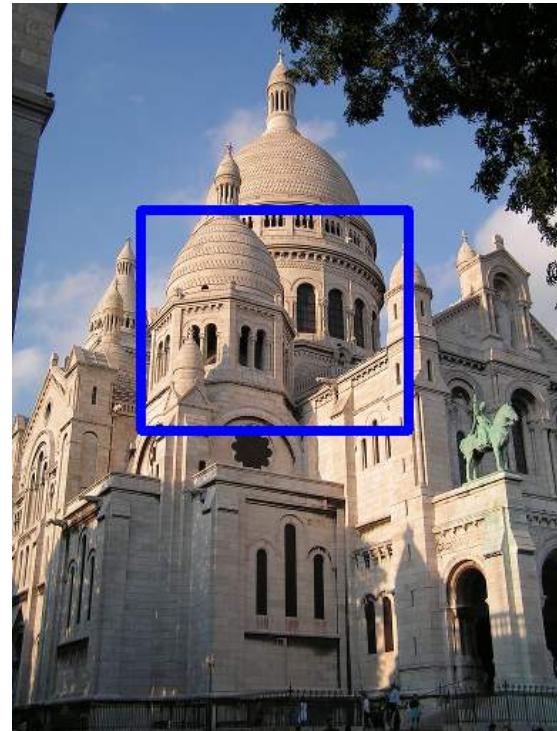
Output Images



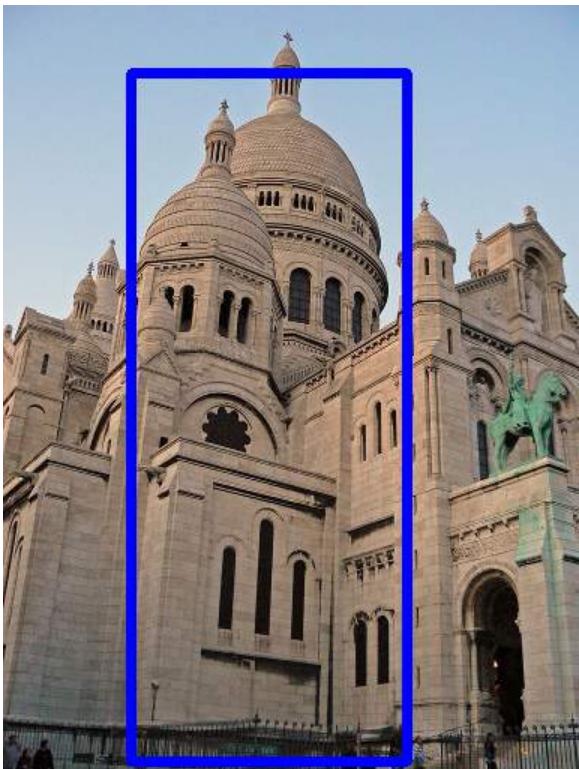
<1>



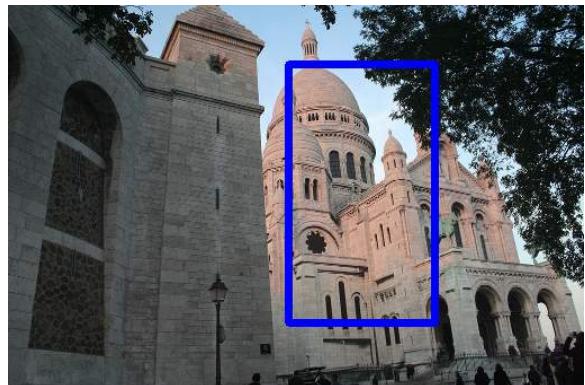
<2>



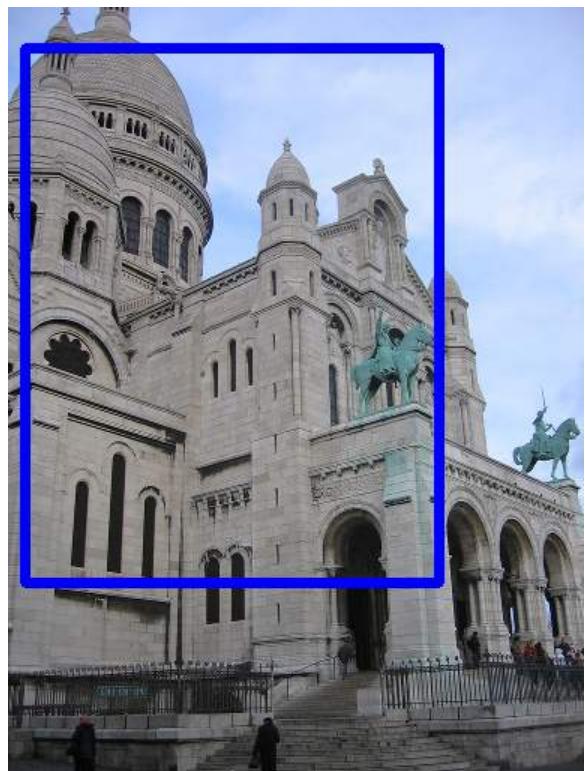
<3>



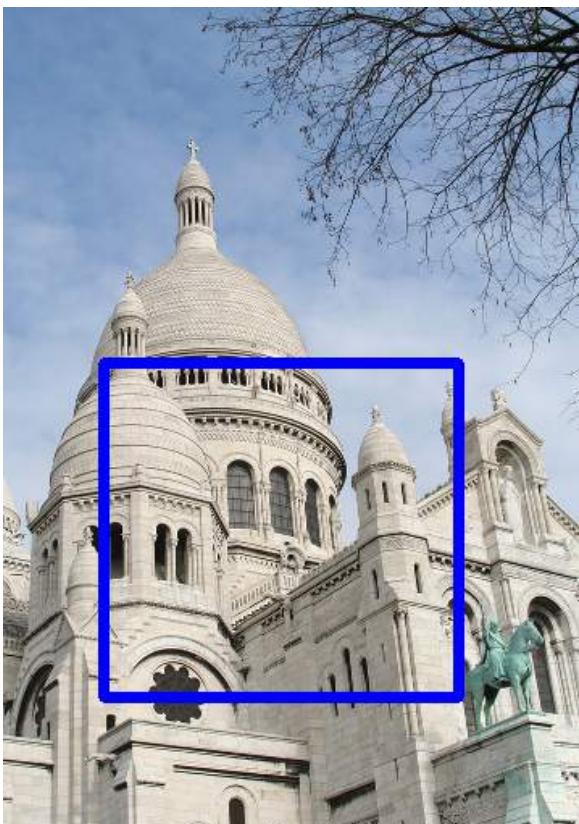
<4>



<6>



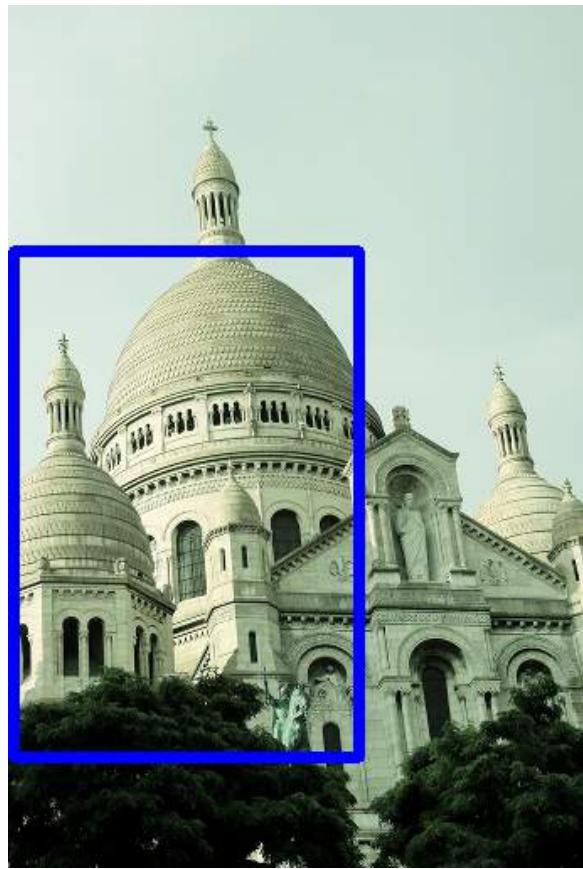
<7>



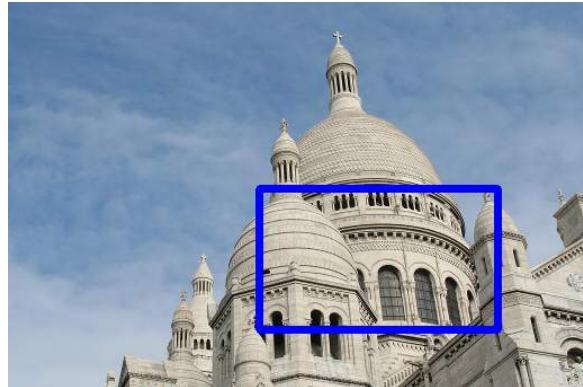
<5>



<8>



<9>



<10>

mAP and Time

```
/Users/saber/PycharmProjects/534P1/venv/bin/python /Users/saber/PycharmProjects/534P1/534Project2.py
In 0 th class it has 381 images, gets 0 images contains sample images
In 1 th class it has 147 images, gets 0 images contains sample images
In 2 th class it has 1497 images, gets 0 images contains sample images
In 3 th class it has 452 images, gets 0 images contains sample images
In 4 th class it has 267 images, gets 0 images contains sample images
In 5 th class it has 445 images, gets 1 images contains sample images
In 6 th class it has 740 images, gets 0 images contains sample images
In 7 th class it has 445 images, gets 0 images contains sample images
In 8 th class it has 660 images, gets 0 images contains sample images
In 9 th class it has 438 images, gets 0 images contains sample images
In 10 th class it has 380 images, gets 45 images contains sample images
In 11 th class it has 551 images, gets 0 images contains sample images
Finish in 791 seconds
```

Sacrecoeur								
Class sacrecoeur size=380, True Positive=45, False Negative=335							TIME=791seconds	
	class name	class size	False Positives	True Negatives	precision	recall	accuracy	mAP
0	defense	381	0	381	1	0.1184211	0.5597898	
1	eiffel	147	0	147	1	0.1184211	0.3643264	
2	genral	1497	0	1497	1	0.1184211	0.8215237	
3	invalides	452	0	452	1	0.1184211	0.5973558	
4	louvre	267	0	267	1	0.1184211	0.4822257	
5	moulinrouge	445	1	444	0.978261	0.1184211	0.5927273	
6	museedorsay	740	0	740	1	0.1184211	0.7008929	
7	notredame	445	0	445	1	0.1184211	0.5939394	
8	pantheon	660	0	660	1	0.1184211	0.6778846	
9	pompidou	438	0	438	1	0.1184211	0.5904645	
11	triomphe	551	0	551	1	0.1184211	0.6401719	0.6019365

Class 11 -- <Triomphe>

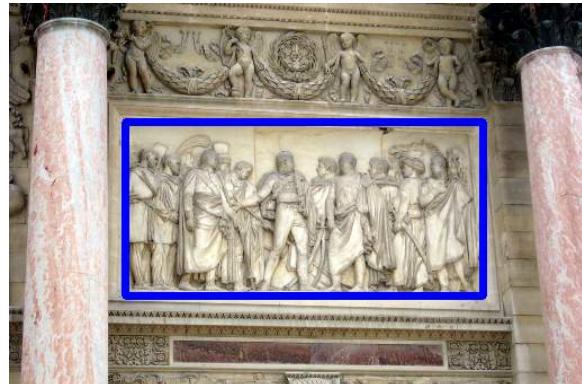
Sample Image



Output Images



<1>



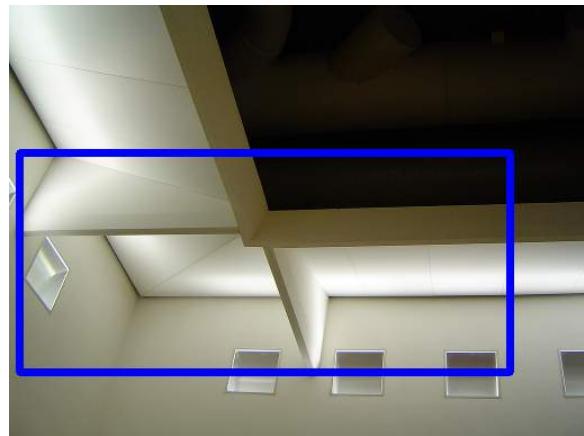
<2>



<3>



<4>



<6>



<7>



<5>

mAP and Time

```
/Users/saber/PycharmProjects/534P1/venv/bin/python /Users/saber/PycharmProjects/534P1/534Project2.py
In 0 th class it has 381 images, gets 0 images contains sample images
In 1 th class it has 147 images, gets 0 images contains sample images
In 2 th class it has 1497 images, gets 1 images contains sample images
In 3 th class it has 452 images, gets 0 images contains sample images
In 4 th class it has 267 images, gets 0 images contains sample images
In 5 th class it has 445 images, gets 0 images contains sample images
In 6 th class it has 740 images, gets 0 images contains sample images
In 7 th class it has 445 images, gets 0 images contains sample images
In 8 th class it has 660 images, gets 1 images contains sample images
In 9 th class it has 438 images, gets 2 images contains sample images
In 10 th class it has 380 images, gets 0 images contains sample images
In 11 th class it has 551 images, gets 3 images contains sample images
Finish in 913 seconds
```

Triomphe								
Class triomphe size=551, True Positive=3, False Negative=548						TIME=913seconds		
	class name	class size	False Positives	True Negatives	precision	recall	accuracy	mAP
0	defense	381	0	381	1	0.0054446	0.4120172	
1	eiffel	147	0	147	1	0.0054446	0.2148997	
2	genral	1497	1	1496	0.75	0.0054446	0.7319336	
3	invalides	452	0	452	1	0.0054446	0.4536391	
4	louvre	267	0	267	1	0.0054446	0.3300733	
5	moulinrouge	445	0	445	1	0.0054446	0.4497992	
6	museedorsay	740	0	740	1	0.0054446	0.5755229	
7	notredame	445	0	445	1	0.0054446	0.4497992	
8	pantheon	660	1	659	0.75	0.0054446	0.5466557	
9	pompidou	438	2	436	0.6	0.0054446	0.4438827	
10	sacrecoeur	380	0	380	1	0.0054446	0.4113856	0.456328

Discussion:

mAP depends on the quality of sample image and the threshold we selected.

- If we set threshold (the variable h's value) small, the precision will be small, the recall will be larger.
- If we set threshold (the variable h's value) large, the precision will be large, the recall will be small.
- If the sample image is clearly, the mAP is larger.
- If there are many other objects around our target objects, the mAP is small.