Task 4.1P Answer sheet

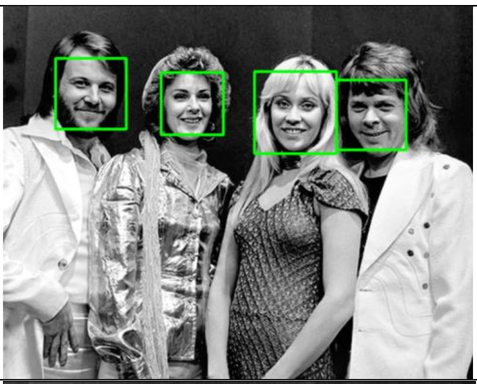
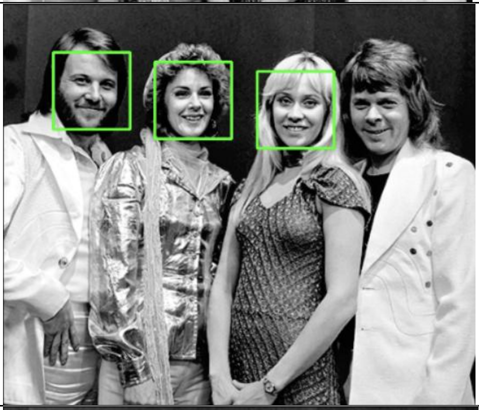
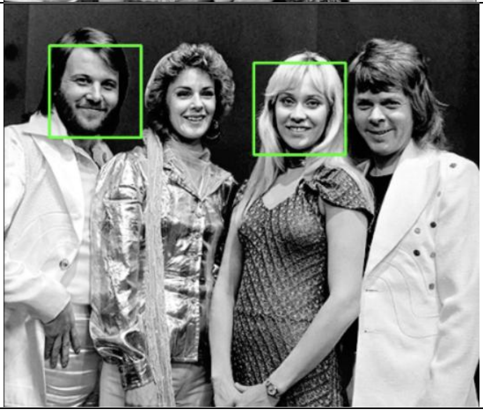
Fill in outputs and discussions required in the tables below

Notes:

- Missing any required outputs will result in a re-submission.

1. Face detection

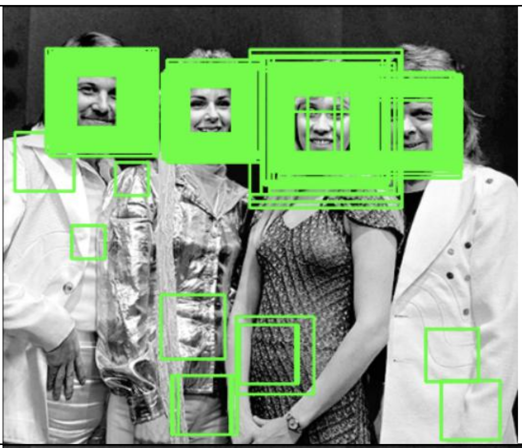
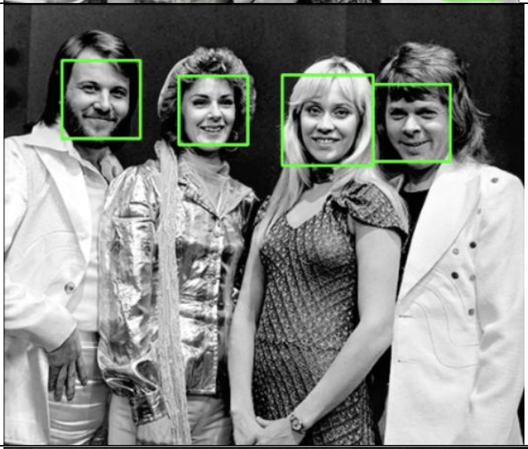
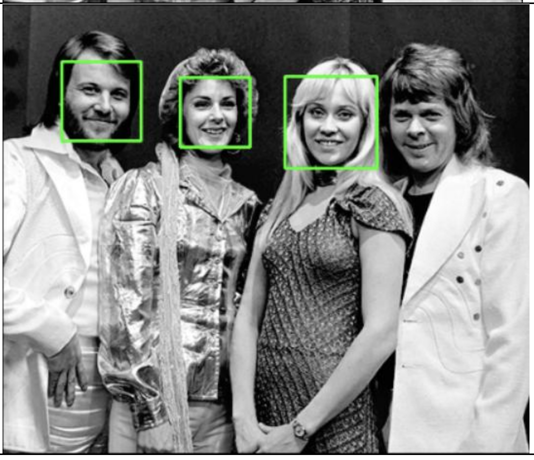
a. Face detection in 'abba.png' when varying scaleFactor (don't change other parameters)

	Detection results	**Processing time (in sec)**
scaleFactor=1.1, minNeighbors=5, minSize=(30,30)		0.026264190673828125 seconds
scaleFactor=1.5, minNeighbors=5, minSize=(30,30)		0.008511781692504883 seconds
scaleFactor=2.0, minNeighbors=5, minSize=(30,30)		0.006016016006469726 6 seconds

Discussion of the results (in terms of detection accuracy and processing speed)

The image's results show how changing the scaleFactor parameter in the detectMultiScale method affects processing time and face detection accuracy. All four faces can be accurately detected with a smaller scaleFactor of 1.1, which increases the detector's sensitivity to small size changes in faces but comes at the cost of a longer processing time (0.026s). The processing time greatly decreases (to 0.0085s and 0.0060s, respectively) as the scaleFactor rises to 1.5 and 2.0, but the detection accuracy reduces because some faces are missed in each of these scales. Larger scale steps skip possible face sizes, which results in a faster but less accurate detector. So depending on the requirements of the application, selecting a scaleFactor requires finding a balance between detection completeness and computing efficiency.

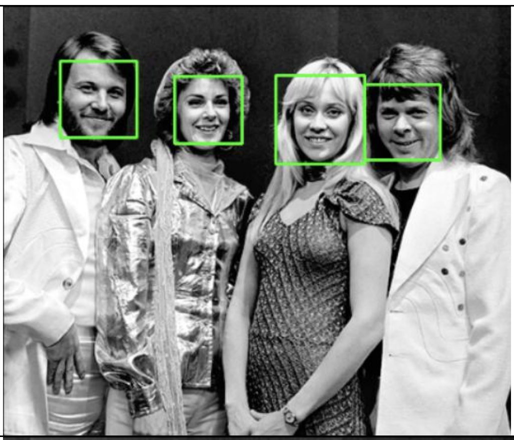
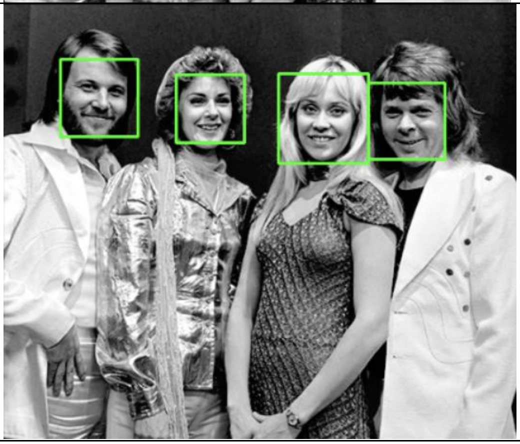
b. Face detection in 'abba.png' when varying minNeighbors (don't change other parameters)

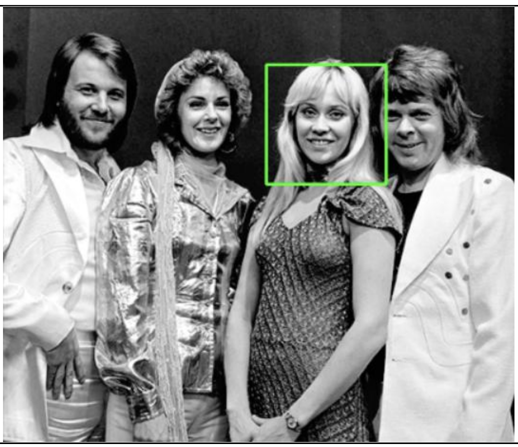
	Detection results	**Processing time (in sec)**
scaleFactor=1.1, minNeighbors=0, minSize=(30,30)		0.0242612361907959 seconds
scaleFactor=1.1, minNeighbors=5, minSize=(30,30)		0.02604222297668457 seconds
scaleFactor=1.1, minNeighbors=50, minSize=(30,30)		0.023021697998046875 seconds

Discussion of the results (in terms of detection accuracy and processing speed)

False positives are greatly impacted when changing minNeighbors. Even though processing is quick (≈ 0.0243 sec), the detector generates a lot of false positives with crowded overlapping boxes when minNeighbors=0. The algorithm finds a strong balance at minNeighbors=5, accurately detecting all four faces in a moderate amount of processing time (≈ 0.0260 sec). The same four faces are identified when minNeighbors=50, but the detection speed is somewhat faster (≈ 0.0230 sec), suggesting that high values in this situation reduce weaker detections without ignoring the correct ones. Extreme values, however, could result in missed detections in scenes that are more complicated or noisy.

c. Face detection in 'abba.png' when varying minSize (don't change other parameters)

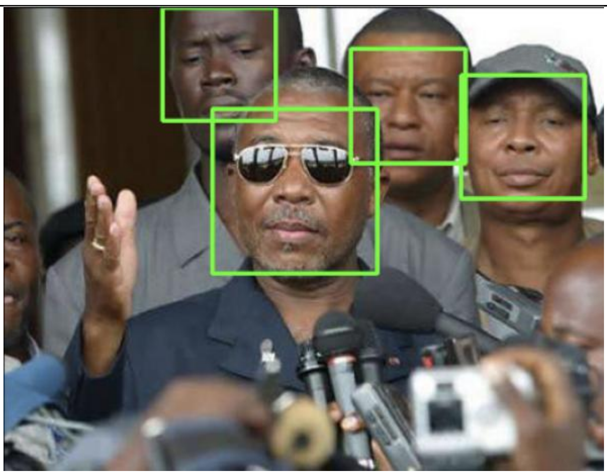
	Detection results	**Processing time (in sec)**
scaleFactor=1.1, minNeighbors=5, minSize=(10,10)		0.04557037353515625 seconds
scaleFactor=1.1, minNeighbors=5, minSize=(30,30)		0.025554418563842773 seconds

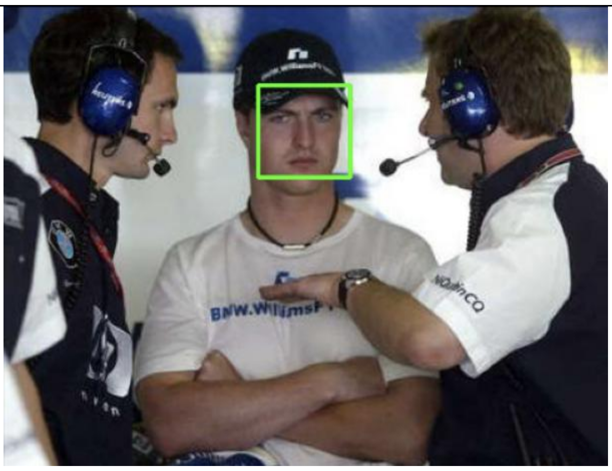
scaleFactor=1.1, minNeighbors=5, minSize=(100,100)		0.00901341438293457 seconds

Discussion of the results (in terms of detection accuracy and processing speed)

While it improves detection coverage, lowering minSize from (100, 100) to (10, 10) lengthens processing time. All four faces are discovered when minSize=(10,10) is used. Smaller faces may also be detected in other situations, however this has the largest processing cost (about 0.0456 seconds). Detection is accurate and reasonably quick (≈ 0.0256 sec) at the default (30,30). Some faces are missed at (100,100), but the pace increases significantly (≈ 0.0090 sec). This demonstrates that while a smaller minSize makes it possible to identify more and smaller faces, it also results in longer computation times because of the additional sliding window operations.

d. Set scaleFactor=1.1, minNeighbors=5, minSize=(30,30), report face detection results and processing time for the images img_1014.jpg and img_1123.jgp

	Detection results	**Processing time (in sec)**
img_1014.jpg		0.013520956039428711 seconds

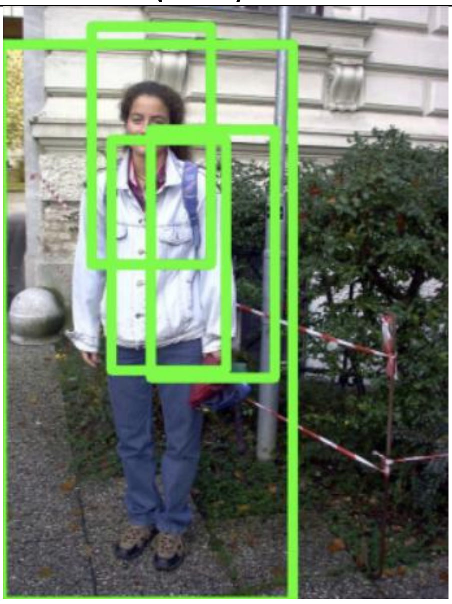
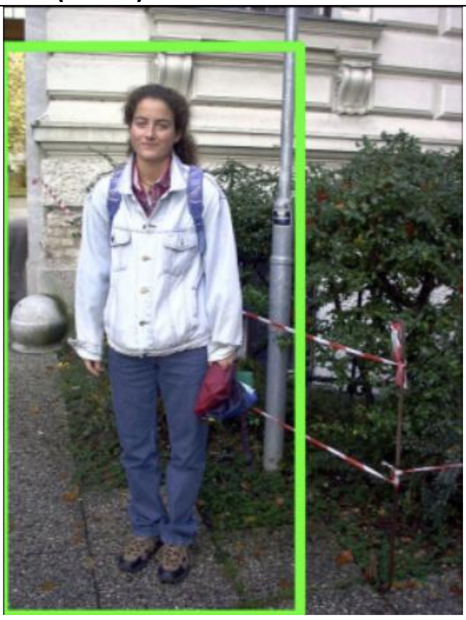
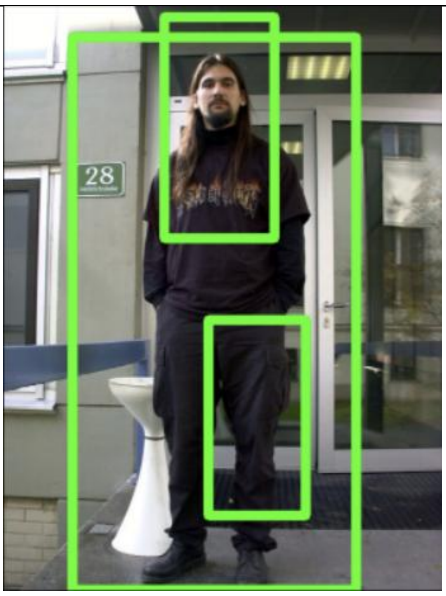
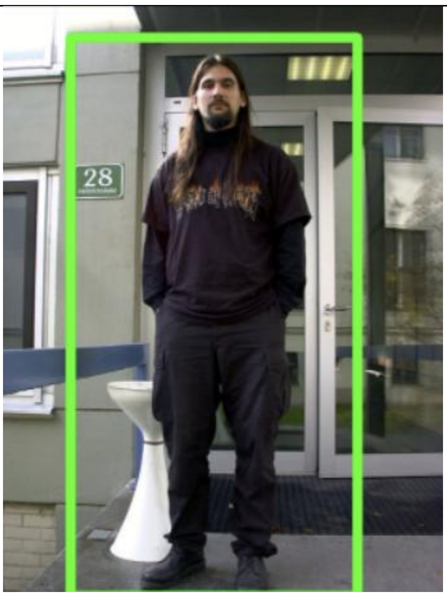
img_1123.jpg		0.013505935668945312 seconds

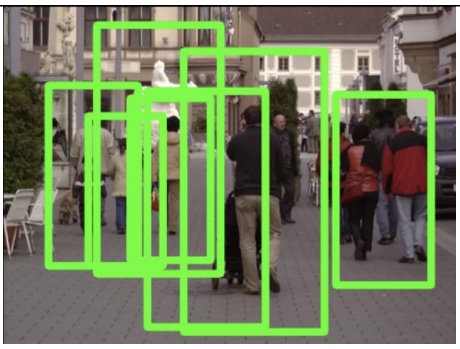
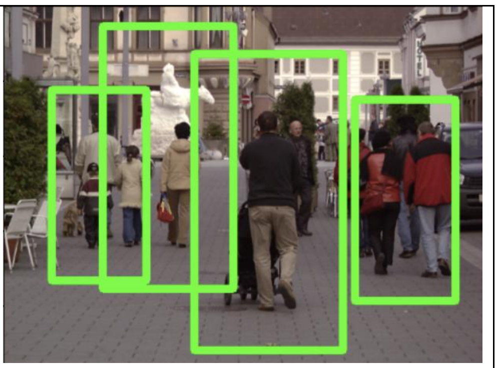
Are all the faces detected? If not, what are the missing faces? What could be the reason?

Not all the faces were detected. The missing faces include faces that are only partially shown(img_1014) and sideway faces(img_1123). Some of the possible reasons include non-optimized parameter or because of challenging conditions of the image. The most probable reason would be the challenging conditions of the image because the detector cannot work properly on challenging lighting conditions, extreme angles, face occlusion, non-frontal faces, blurring, and shadows.

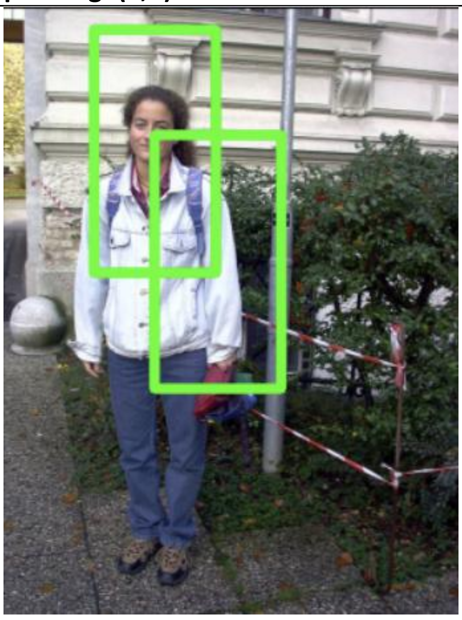
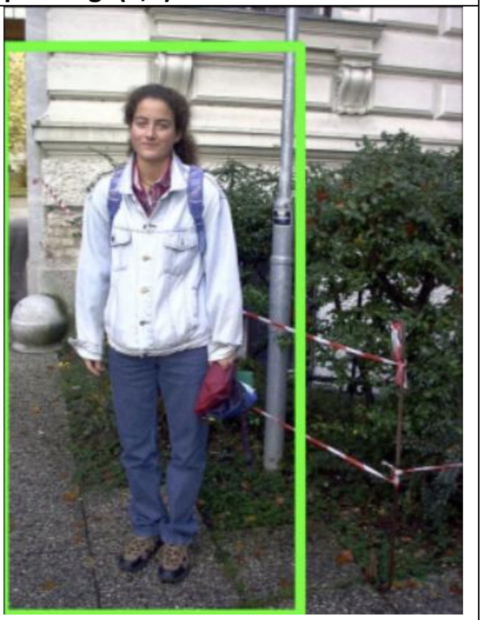
2. Pedestrian detection

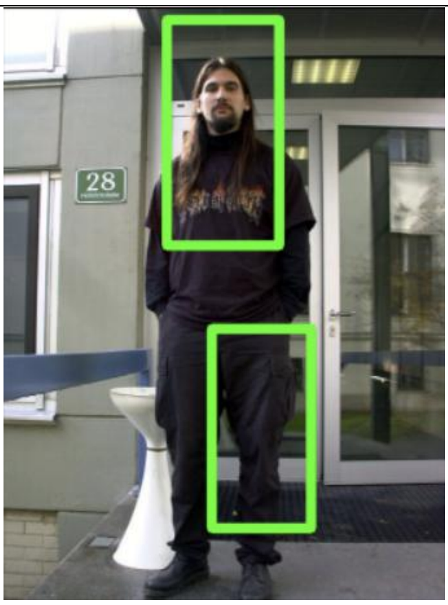
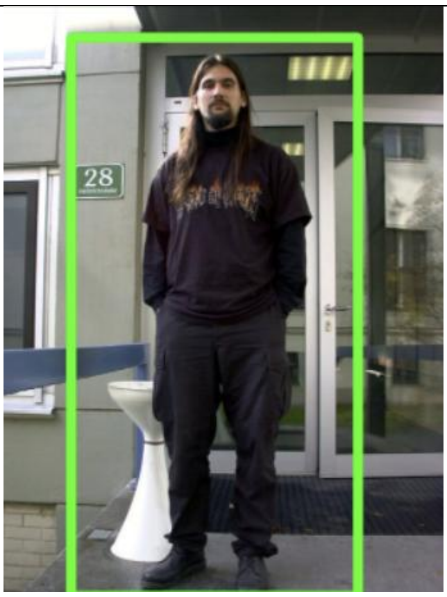
a. Investigate the impact of nms to pedestrian detection in the images person_029.png, person_032.png, and person_236.png (don't change other parameters, i.e., winStride=(4,4), padding=(6,6), scale=1.05). For each test case, please report both the detection results and processing time (in sec).

	Detection results without using boxes = nms(boxes)	**Detection results using boxes = nms(boxes)**
person_029.png	 Processing time: 0.07255053520202637 seconds	 Processing time: 0.08378982543945312 seconds
person_032.png	 Processing time: 0.06708931922912598 seconds	 Processing time: 0.0841057300567627 seconds

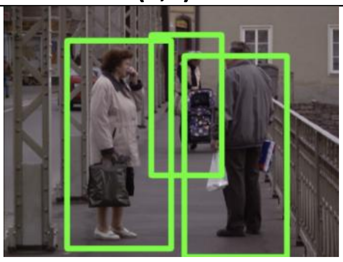
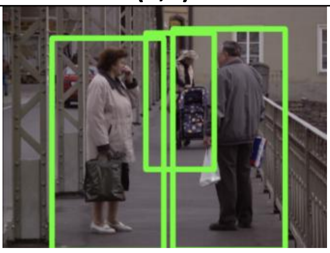
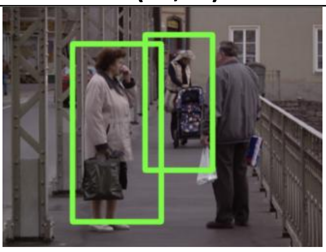
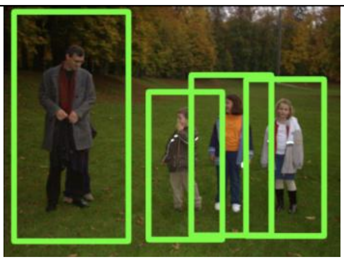
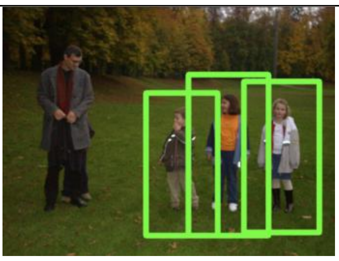
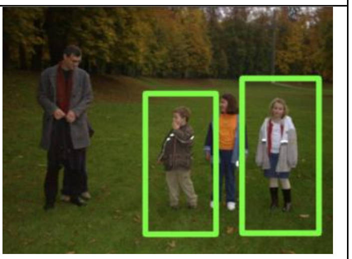
person_236.png	 Processing time: 0.03511929512023926 seconds	 Processing time: 0.041313886642456055 seconds

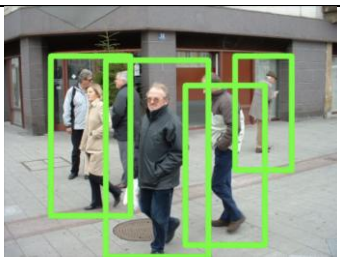
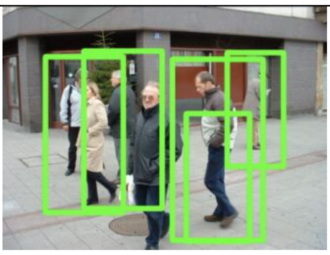
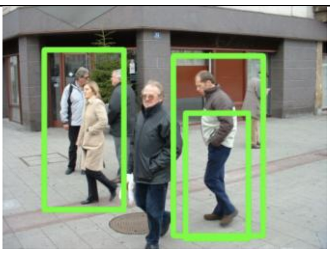
b. Investigate the impact of padding to pedestrian detection in the images person_029.png and person_032.png (don't change other parameters, i.e., winStride=(4,4), scale=1.05, using nms).

	padding=(0,0)	**padding=(6,6)**
person_029.png	 Processing time: 0.0707700252532959 seconds	 Processing time: 0.08378982543945312 seconds

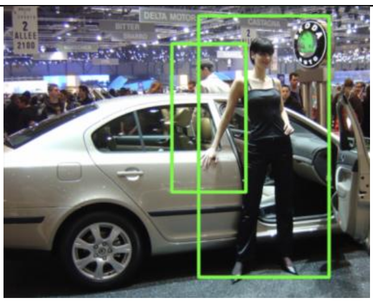
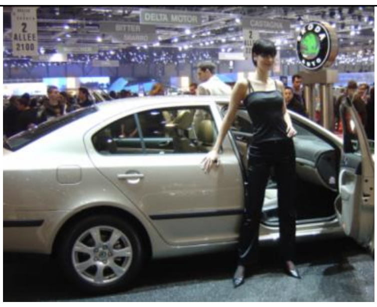
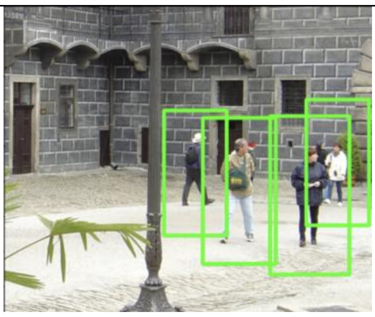
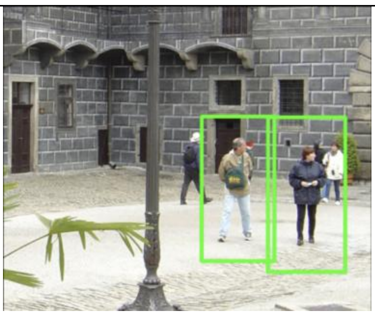
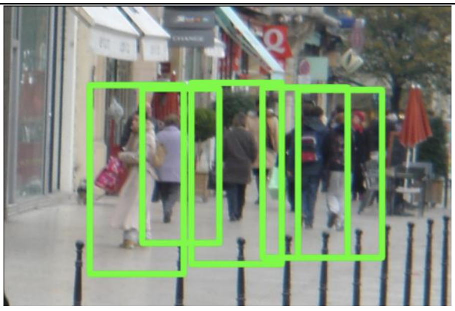
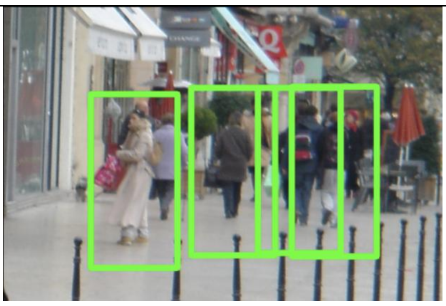
person_032.png	 Processing time: 0.06724262237548828 seconds	 Processing time: 0.041313886642456055 seconds

c. Investigate the impact of winStride to pedestrian detection in the images person_222.png, person_265.png, and person_303.png (don't change other parameters, i.e., padding=(6,6), scale=1.05, using boxes=nms(boxes)). For each test case, please report both the detection results and processing time (in sec).

	winStride=(4,4)	**winStride=(8,8)**	**winStride=(12,12)**
person_222.png	 Processing time: 0.03662276268005371 seconds	 Processing time: 0.012021541595458984 seconds	 Processing time: 0.016067981719970703 seconds
person_265.png	 Processing time: 0.03802180290222168 seconds	 Processing time: 0.013074398040771484 seconds	 Processing time: 0.020527601242065430 seconds

person_303.png	 Processing time: 0.03819751739501953 seconds	 Processing time: 0.013025283813476562 seconds	 Processing time: 0.015016555786132812 seconds

d. Investigate the impact of scale to pedestrian detection in the images crop001706.png, crop001718.png, and crop001521.png (don't change other parameters, i.e., winStride=(4,4), padding=(6,6), using boxes=nms(boxes)). For each test case, please report both the detection results and processing time (in sec).

	scale = 1.05	**scale = 1.2**
crop001706.png	 Processing time: 0.03959846496582031 seconds	 Processing time: 0.021612167358398438 seconds
crop001718.png	 Processing time: 0.043068647384643555 seconds	 Processing time: 0.023734569549560547 seconds
crop001521.png	 Processing time: 0.032566070556640625 seconds	 Processing time: 0.01979851722717285 seconds