SIT789 - Applications of Computer Vision and Speech Processing
Pass Task 4.2: Object detection

OBSERVATIONS:

As the scale factor increases the performance speed gets better but it directly affects the quality of performance i.e. detection result in this case the number of faces detected. Is less precise.

As the minNeighbhors increases the detection results reduces

As winStride varies towards higher range the less the number of pedestrians detected

Padding in the range doesn't affect the results much as observed with few examples except for it tries to cover the entire border of the pedestrian.

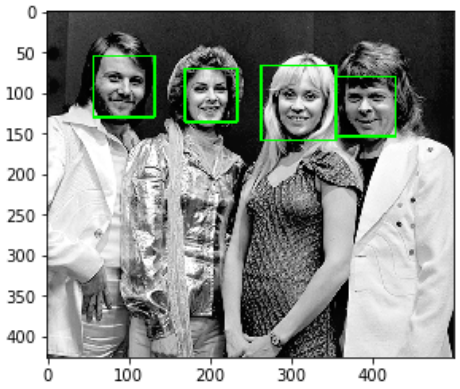
In [162]:
```
'sagar/Documents/APP of CV&SP/Ontrack resources/Resources_4.2/FaceImages/abba.png

leClassifier('haarcascade_frontalface_default.xml')

:ascade_detector)
'formed in %s seconds ---' % (time.time() - start_time))
```

```
Face detection is performed in 0.03690195083618164 seconds ---
```

In [163]:
```
if (faces is not None):
    print('Found ', len(faces), ' faces')
else:
    print('There is no face found!')
```

```
Found  4  faces
```

In [164]:
```
from matplotlib import pyplot as plt
for (x, y, w, h) in faces:
    cv.rectangle(image, (x, y), (x + w, y + h), (0, 255, 0), 2)
    plt.imshow(image[:,:,::-1]) # RGB-> BGR
```



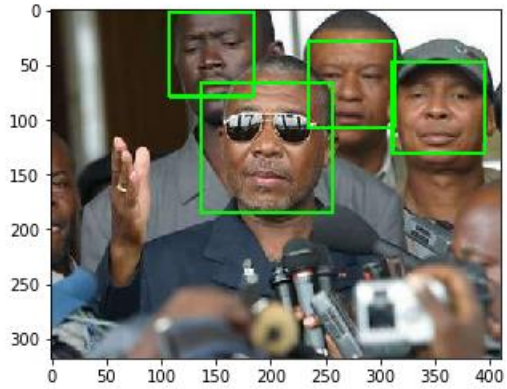
In [165]:
```
'/Documents/APP of CV&SP/Ontrack resources/Resources_4.2/FaceImages/img_1014.jpg'

ssifier('haarcascade_frontalface_default.xml')

le_detector)
ed in %s seconds ---' % (time.time() - start_time))
```

```
Face detection is performed in 0.015957117080688477 seconds ---
```

In [166]:
```
if (faces is not None):
    print('Found ', len(faces), ' faces')
else:
    print('There is no face found!')
```

```
Found  4  faces
```

In [167]:
```
from matplotlib import pyplot as plt
for (x, y, w, h) in faces:
    cv.rectangle(image, (x, y), (x + w, y + h), (0, 255, 0), 2)
    plt.imshow(image[:,:,::-1]) # RGB-> BGR
```



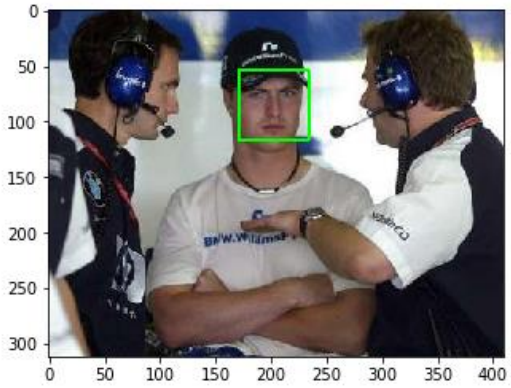
In [168]:
```
gar/Documents/APP of CV&SP/Ontrack resources/Resources_4.2/FaceImages/img_1123.jpg

assifier('haarcascade_frontalface_default.xml')

ade_detector)
med in %s seconds ---' % (time.time() - start_time))
```

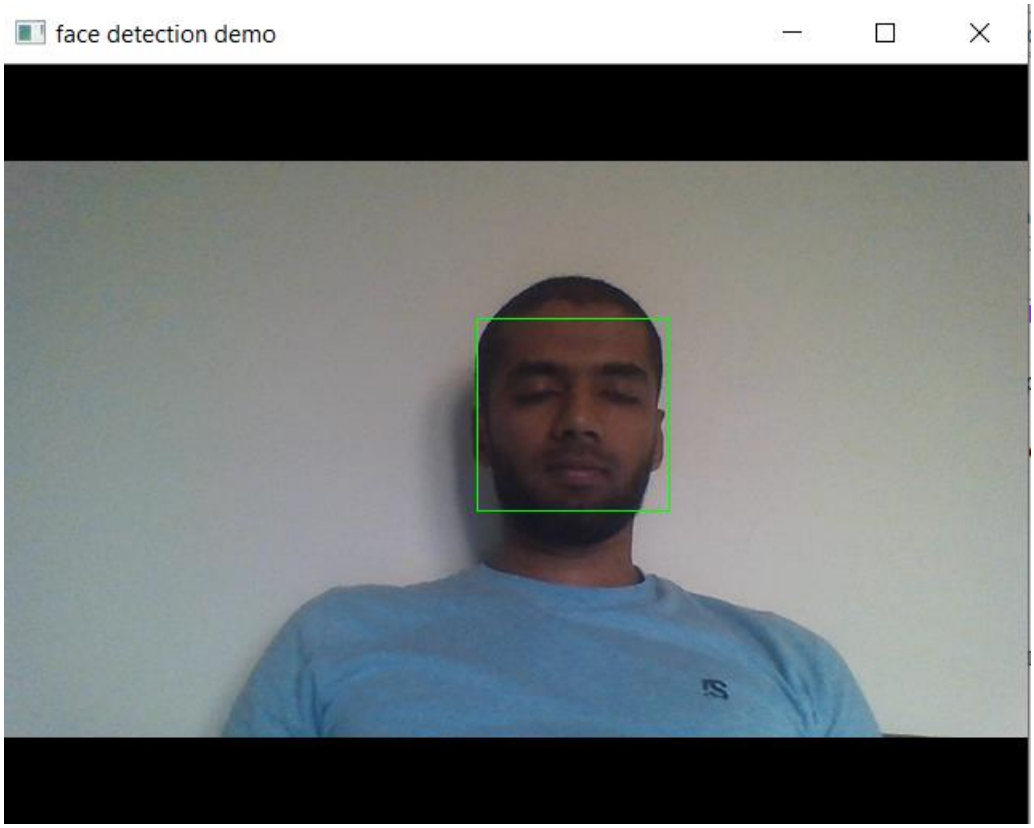
```
Face detection is performed in 0.01199960708618164 seconds ---
```

In [169]:
```
if (faces is not None):
    print('Found ', len(faces), ' faces')
else:
    print('There is no face found!')
```

```
Found  1  faces
```

In [170]:
```
from matplotlib import pyplot as plt
for (x, y, w, h) in faces:
    cv.rectangle(image, (x, y), (x + w, y + h), (0, 255, 0), 2)
    plt.imshow(image[:,:,::-1]) # RGB-> BGR
```



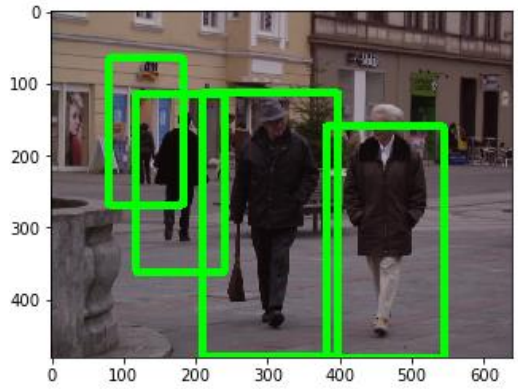


In [13]:
```
import time
image = cv.imread('C:/Users/sagar/Documents/APP of CV&SP/Ontrack resources/Resou
start_time = time.time()
pedestrians = detect_pedestrian(image)
print('Pedestrian detection is performed in %s seconds ---' % (time.time() - sta
```

```
Pedestrian detection is performed in 0.08478450775146484 seconds ---
```

In [14]:
```
from matplotlib import pyplot as plt
if (pedestrians is not None):
    print('Found ', len(pedestrians), ' pedestrians')
    for (x, y, w, h) in pedestrians:
        cv.rectangle(image, (x, y), (x + w, y + h), (0, 255, 0), 10)
    plt.imshow(image[:,:,::-1]) # RGB-> BGR
else:
        print('There is no pedestrian found!')
```

```
Found  4  pedestrians
```



In [15]:

```
ients/APP of CV&SP/Ontrack resources/Resources_4.2/PedestrianImages/person_032.png

ed in %s seconds ---' % (time.time() - start_time))
```

```
Pedestrian detection is performed in 0.15957307815551758 seconds ---
```

In [16]:

```
from matplotlib import pyplot as plt
if (pedestrians is not None):
    print('Found ', len(pedestrians), ' pedestrians')
    for (x, y, w, h) in pedestrians:
        cv.rectangle(image, (x, y), (x + w, y + h), (0, 255, 0), 10)
    plt.imshow(image[:,:,::-1]) # RGB-> BGR
else:
        print('There is no pedestrian found!')
```

```
Found  3  pedestrians
```