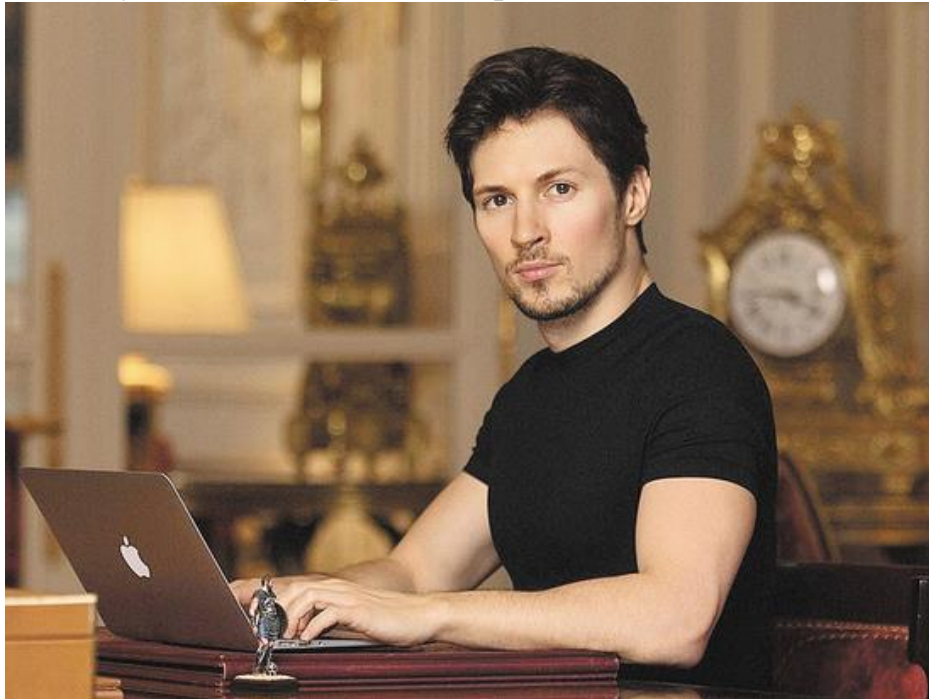


# Laboratorium №10

## Wykrywanie obiektów za pomocą OpenCV

### 1. Wykrywanie twarzy na zdjęciu:

Algorytm Viola-Jones został opracowany do wykrywania twarzy na zdjęciu. Załóżmy, że musimy podkreślić przednią twarz



```
import cv2
import numpy
import imutils

face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
scaling_factor = 0.5
frame = cv2.imread("durov.jpg")
frame = cv2.resize(frame, None, fx=scaling_factor, fy=scaling_factor, interpolation=cv2.INTER_AREA)
face_rects = face_cascade.detectMultiScale(frame, scaleFactor=1.3, minNeighbors=5)

for(x,y,w,h) in face_rects:
    cv2.rectangle(frame, (x,y), (x+w,y+h), (255,0,0), 3)

cv2.imshow("durov.jpg", frame)
cv2.waitKey(0)
print(f'Found {len(face_rects)} faces!')
```



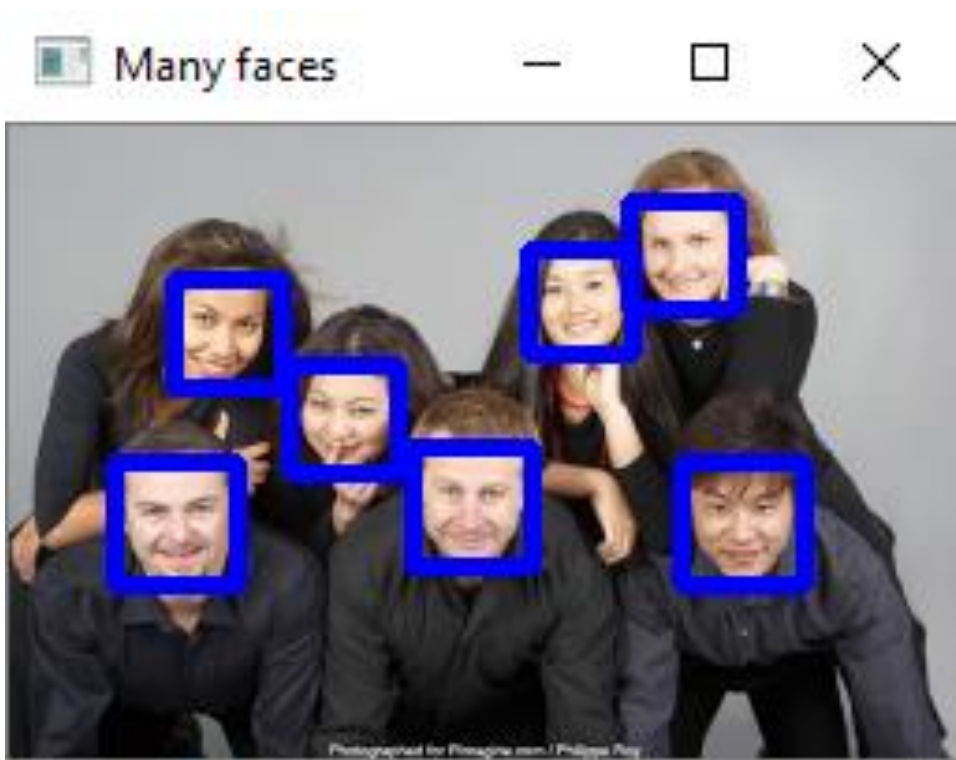
## 2. Wykrywanie wielu twarzy na zdjęciu

Ten klasyfikator może również wykryć wiele twarzy na zdjęciu i policzyć ich liczbę

```
face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
scaling_factor = 0.5
frame = cv2.imread("many_faces.jpg")
frame = cv2.resize(frame, None, fx=scaling_factor, fy=scaling_factor, interpolation=cv2.INTER_AREA)
face_rects = face_cascade.detectMultiScale(frame, scaleFactor=1.1, minNeighbors=5)

for(x,y,w,h) in face_rects:
    cv2.rectangle(frame, (x,y), (x+w,y+h), (255,0,0), 3)

cv2.imshow("Many faces", frame)
cv2.waitKey(0)
print(f'Found {len(face_rects)} faces!')
```

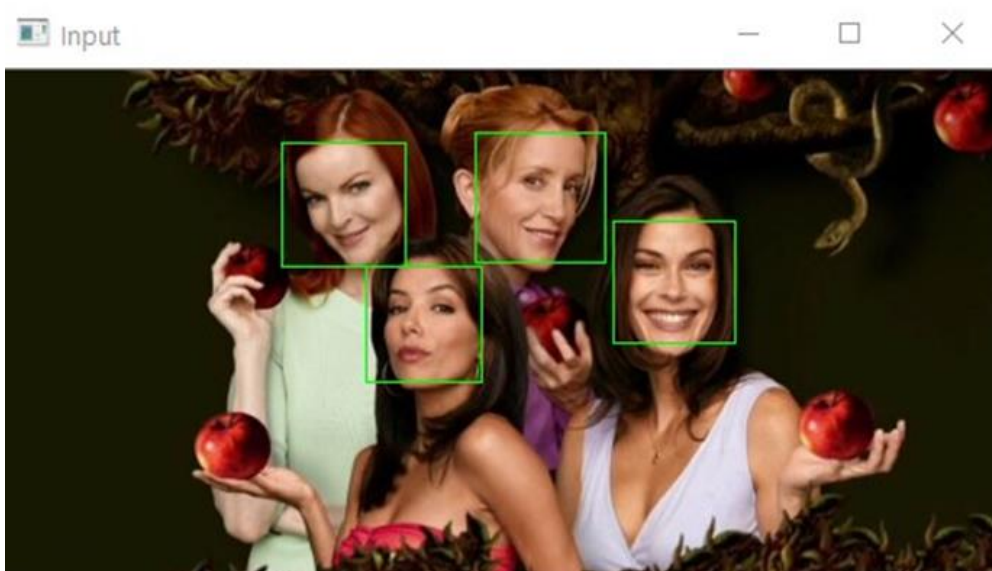


Found 7 faces!

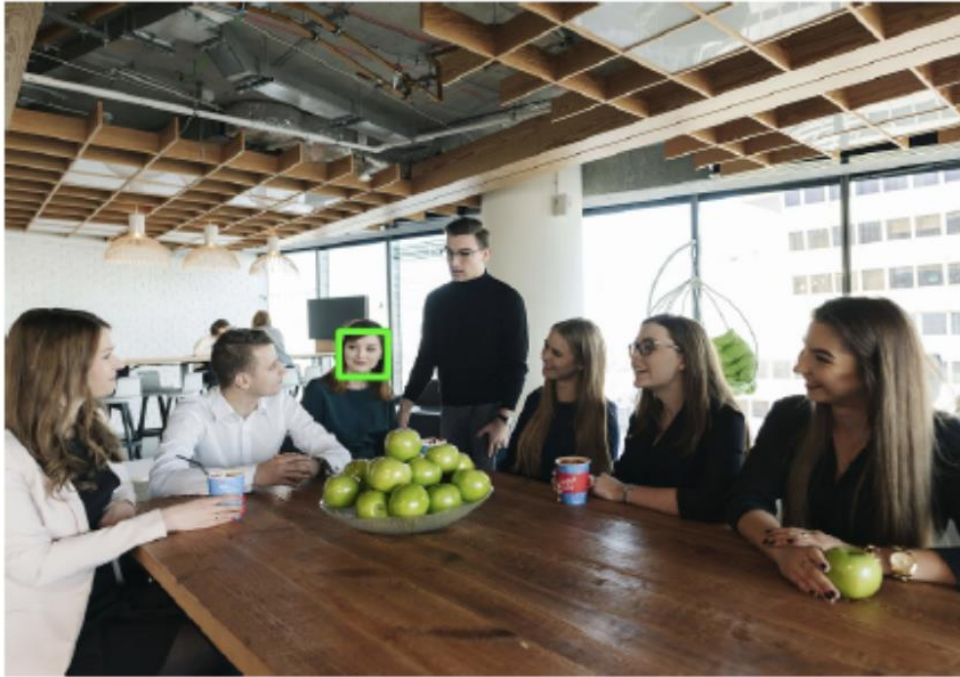
Process finished with exit code 0

|

Lub:



Ale jeśli twarze nie znajdują się z przodu, lub są z profilu, ten klasyfikator może nie wykryć takiej twarzy, przykład takiej sytuacji jest na poniższym zdjęciu (znaleziono tylko jedną twarz):



Zmieniając ustawienia można zwiększyć liczbę wykrywanych osób

### 3. Wykrywanie twarzy, uśmiechów, oczu ludzi

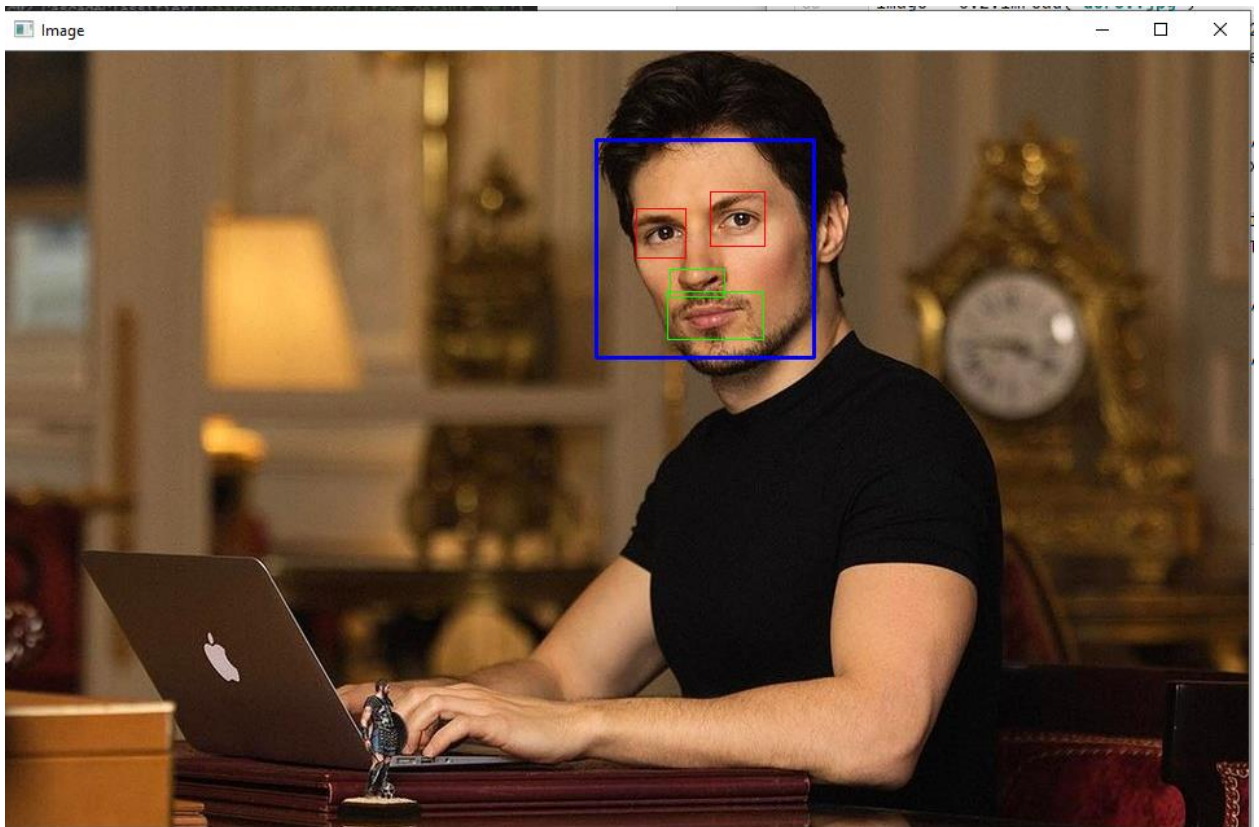
```
face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
smile_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_smile.xml')
eye_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_eye.xml')

image = cv2.imread("durov.jpg")
gray_filter = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
faces = face_cascade.detectMultiScale(gray_filter, 1, 4)

for(x,y,w,h) in faces:
    cv2.rectangle(image, (x,y), (x+w, y+h), (255,0,0), 2)
    roi_gray = gray_filter[y:y+h, x:x+w]
    roi_color = image[y:y+h, x:x+w]
    smile = smile_cascade.detectMultiScale(roi_gray)
    eye = eye_cascade.detectMultiScale(roi_gray)
    for(sx,sy,sw,sh) in smile:
        cv2.rectangle(roi_color, (sx,sy), (sx + sw, sy+sh), (0,255,0), 1)
    for (ex, ey, ew, eh) in eye:
        cv2.rectangle(roi_color, (ex, ey), (ex + ew, ey + eh), (0, 0, 255), 1)

cv2.imshow("Image", image)
cv2.waitKey(0)
```





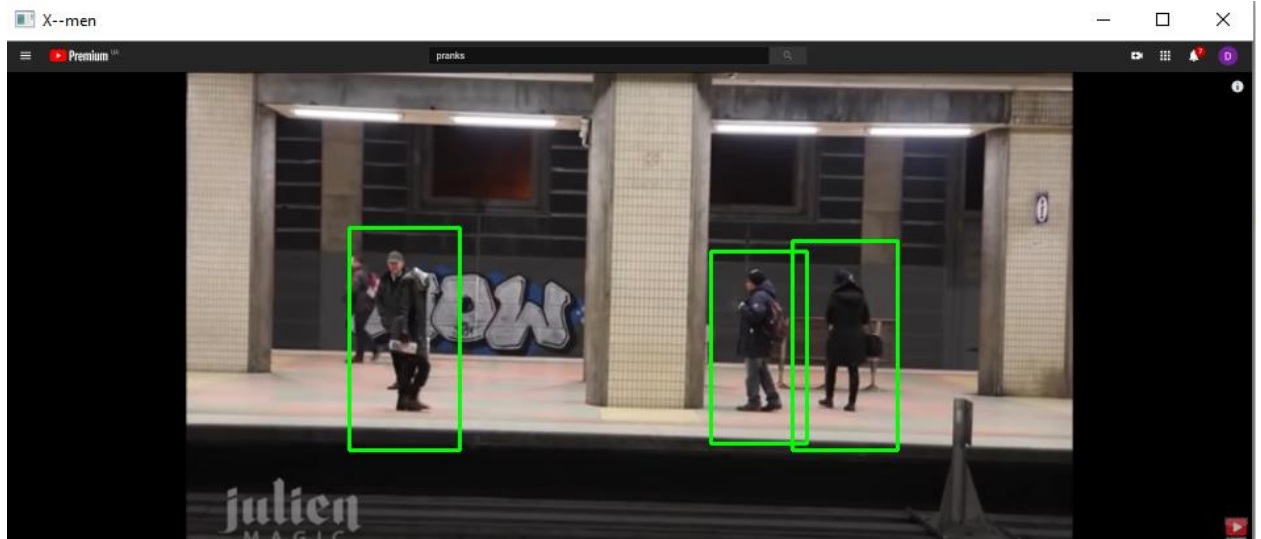
#### 4. Zliczanie osób na zdjęciu

##### Zastosowanie klasyfikatorów HOG

```
scaling_factor = 0.5
hog = cv2.HOGDescriptor()
hog.setSVMDetector(cv2.HOGDescriptor_getDefaultPeopleDetector())
image = cv2.imread("screenshot.png")
image = cv2.resize(image, None, fx=scaling_factor, fy=scaling_factor, interpolation=cv2.INTER_AREA)
people_rects = hog.detectMultiScale(image, winStride=(8,8), padding=(30,30), scale=1.06)

for(x,y,w,h) in people_rects[0]:
    cv2.rectangle(image, (x,y), (x+w, y+h), (0,255,0),2)

cv2.imshow("X-men", image)
cv2.waitKey(0)
print(f'Found {len(people_rects[0])} people!')
```



Found 3 people!

Process finished with exit code 0

## 5. Rozpoznawanie osób na wideo

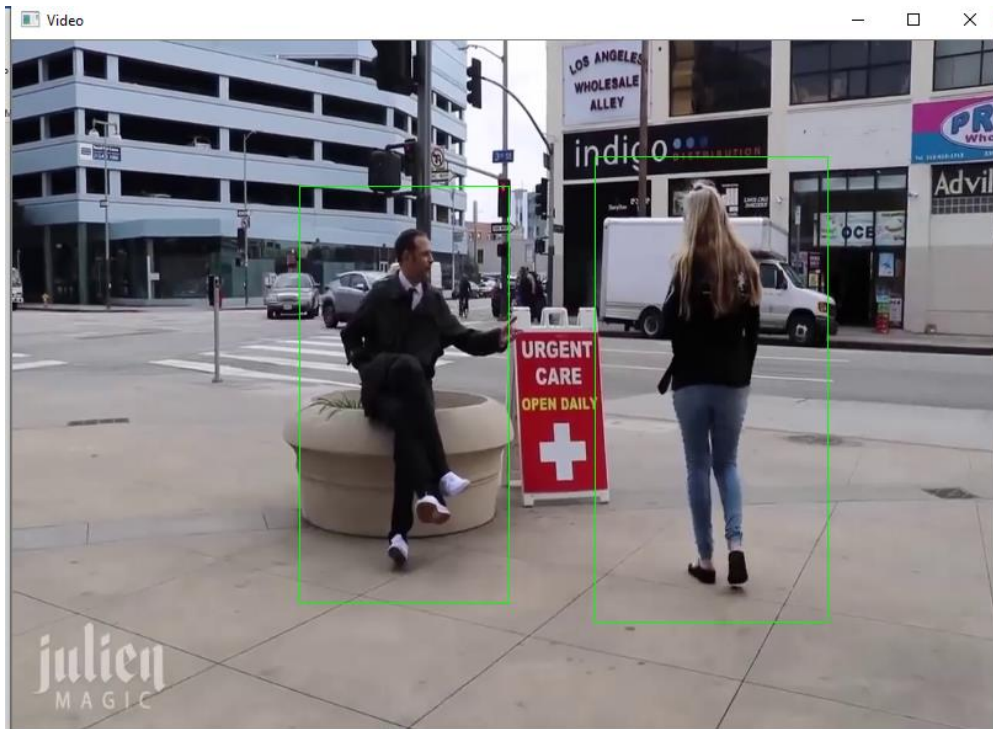
Zastosowanie klasyfikatorów HOG

```
hog = cv2.HOGDescriptor()
hog.setSVMDetector(cv2.HOGDescriptor_getDefaultPeopleDetector())
cv2.startWindowThread()
cap = cv2.VideoCapture('video_with_people.mp4')

while True:
    ret, frame = cap.read()

    frame = cv2.resize(frame, (800, 560))
    gray_filter = cv2.cvtColor(frame, cv2.COLOR_RGB2GRAY)
    boxes, weights = hog.detectMultiScale(frame, winStride=(8,8))
    boxes = numpy.array([x,y,x+w,y+h] for (x,y,w,h) in boxes)

    for(xa,ya,xb,yb) in boxes:
        cv2.rectangle(frame, (xa,ya), (xb,yb), (0,255,0), 1)
    cv2.imshow("Video", frame)
    if(cv2.waitKey(1) & 0xFF==ord('q')):
        break
cap.release()
cv2.destroyAllWindows()
```



### Zadania:

1. Na **własnym** zdjęciu proszę **wykryć swoją** twarz.
2. Na **zdjęciu**, na którym państwo jest z kolegami z klasy (przyjaciółmi, rodziną), **znaleźć twarz, oczy, nos, uśmiech. Policzyć liczbę osób na zdjęciu.**
3. Utwórz film, który odsłania Twoją (!) Twarz z kamery laptopa przez co najmniej 30 sekund. Uzyskaj dostęp do wideo z kamery laptopa za pomocą cv2. VideoCapture (). **Wykryć swoją twarz na podstawie obrazu wideo (na żywo lub nagranego) z kamery.**
4. Przetwórz plik wideo tak, aby **podkreślić pieszych** i, jeśli to możliwe, ich twarze. **Policzyć ilość wykrytych osób na nagraniu.** Możesz pobrać plik z YouTube i wyciąć wideo przez co najmniej 30 sekund.
5. W trakcie obrony pracy laboratoryjnej należy umieć wyjaśnić główne działania i algorytmy zastosowane w pracy