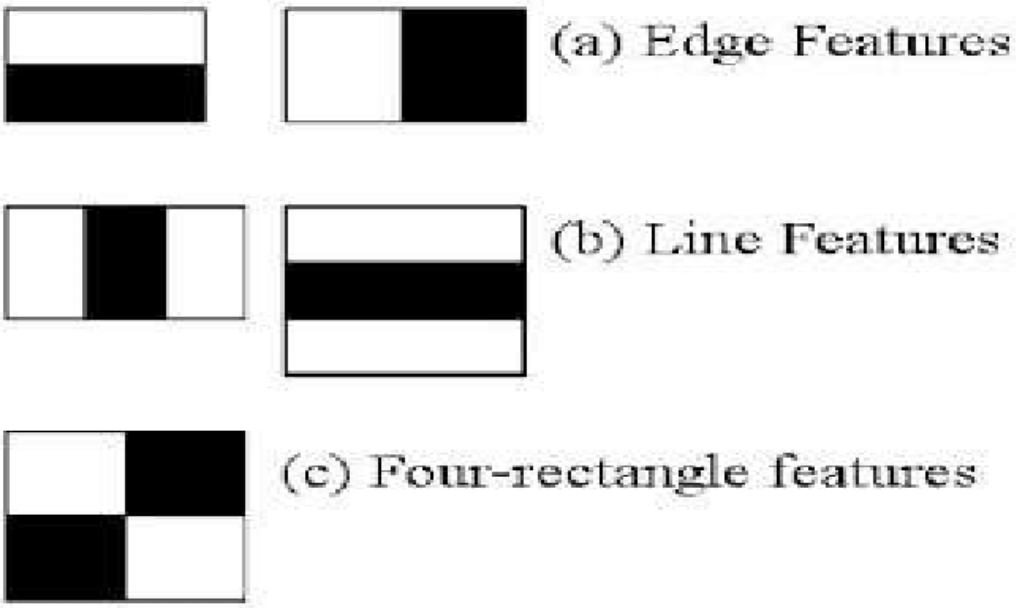
Nama : Dena Cahya Setia Putri
Kelas : JumaTec

Object Detection with Haar Cascade

Deteksi Objek menggunakan Haar feature-based cascade classifiers adalah metode deteksiobjek yang diajukan pada tahun 2001 oleh Paul Viola dan Michael Jones dalam paper, kemudian pada tahun 2002 diteruskan oleh Rainer Lienhart dalam paper.

Haar Feature adalah fitur yang didasarkan pada Wavelet Haar. Wavelet Haar adalah gelombang tunggal bujur sangkar (satu interval tinggi dan satu interval rendah). Untuk duadimensi, satu terang dan satu gelap. Setiap Haar-like feature terdiri dari gabungan kotak- kotak hitam dan putih.



OpenCV menyediakan Cascade Classifier yang dapat diimplementasikan denganmenggunakan fungsi cv2.CascadeClassifier().

Jenis-jenis pre-trained model Cascade Classifier dapat ditemukan pada link github berikut.

Referensi tambahan :

1. Cascade Classifier
2. Cascade Classifier Training

Download File

Download pre-trained model Haar Cascade dari Google Drive

```
!gdown https://drive.google.com/uc?id=1PIX8seCgYcyR9R6vw1o88-v2QZ7YYH_J
```

```
Downloading...
From: https://drive.google.com/uc?id=1PIX8seCgYcyR9R6vw1o88-v2QZ7YYH_JTo:
/content/haarcascades.zip
     0% 0.00/1.46M [00:00<?, ?B/s] 100% 1.46M/1.46M [00:00<00:00,
189MB/s]
```

Unzip pre-train model Haar Cascade yang sudah di download

```
!unzip '/content/haarcascades.zip'
```

```
Archive:  /content/haarcascades.zip
replace haarcascade_eye.xml? [y]es, [n]o, [A]ll, [N]one, [r]ename:
```

Download gambar foto keluarga yang akan digunakan untuk object detection

```
!wget --no-check-certificate \

https://res.klook.com/images/fl_lossy.progressive,q_65/c_fill,w_1350,h
_1080,f_auto/w_80,x_15,y_15,g_south_west,l_klook_water/activities/
qupuw4hgkhwuaijbjlvp/FamilyPortraitExperiencedariZ-anddiSeoul.jpg \
      -O Potrait.jpg
```

Download library tambahan dari OpenCV

```
!wget --no-check-certificate \
      https://raw.githubusercontent.com/computationalcore/introduction-
to-opencv/master/utils/common.py \
      -O common.py
```

Import Library

Mengimport library yang dibutuhkan yaitu :

1. OpenCV
2. Numpy
3. Matplotlib dan pylab

```
import cv2
import common
import numpy as np

%matplotlib inline
from matplotlib import pyplot as plt
import pylab
pylab.rcParams['figure.figsize'] = (10.0, 8.0)
```

```
# these imports let you use opencv
import cv2 #opencv itself
import common #some useful opencv functions
```

```
import numpy as np # matrix manipulations

#the following are to do with this interactive notebook code
%matplotlib inline
from matplotlib import pyplot as plt # this lets you draw inlinepictures in the notebooks
import pylab # this allows you to control figure size pylab.rcParams['figure.figsize'] = (10.0, 8.0) # this controls figuresize in the notebook
```

Read Image

Membaca dan menampilkan gambar yang sudah di download

```
base_image = cv2.imread('Potrait.jpg')
grey = cv2.cvtColor(base_image, cv2.COLOR_BGR2GRAY)
rgb = cv2.cvtColor(base_image, cv2.COLOR_BGR2RGB)
plt.imshow(rgb)
```

base_image = cv2.imread('Potrait.jpg')
grey = cv2.cvtColor(base_image, cv2.COLOR_BGR2GRAY)rgb = cv2.cvtColor(base_image, cv2.COLOR_BGR2RGB) plt.imshow(rgb)

Face Detection

Face detection menggunakan pre-trained model haarcascade_frontalface_default.xml untuk mendeteksi wajah pada gambar.

```
test_image = cv2.imread('Potrait.jpg')
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
faces = face_cascade.detectMultiScale(grey, 1.3, 5)
for (x,y,w,h) in faces:
     cv2.rectangle(test_image,(x,y),(x+w,y+h),(0,0,255),2)
face_output = cv2.cvtColor(test_image, cv2.COLOR_BGR2RGB)
plt.imshow(face_output)
```

test_image = cv2.imread('Potrait.jpg')face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')faces = face_cascade.detectMultiScale(grey, 1.3, 5)
for (x,y,w,h) **in** faces: cv2.rectangle(test_image,(x,y),(x+w,y+h),(0,0,255),2)
face_output = cv2.cvtColor(test_image, cv2.COLOR_BGR2RGB)plt.imshow(face_output)

Smile Detection

Smile detection menggunakan pre-trained model haarcascade_smile.xml untuk mendeteksi senyum pada gambar.

```
test_image = cv2.imread('Potrait.jpg')
smile_cascade = cv2.CascadeClassifier('haarcascade_smile.xml')
smiles = smile_cascade.detectMultiScale(grey, 1.3, 20)
for (x,y,w,h) in smiles:
     cv2.rectangle(test_image,(x,y),(x+w,y+h),(0,255,0),2)
smile_output = cv2.cvtColor(test_image, cv2.COLOR_BGR2RGB)
plt.imshow(smile_output)
```

test_image = cv2.imread('Potrait.jpg')
smile_cascade = cv2.CascadeClassifier('haarcascade_smile.xml') smiles = smile_cascade.detectMultiScale(grey, 1.3, 20) *#1.3 20***for** (x,y,w,h) **in** smiles:
cv2.rectangle(test_image,(x,y),(x+w,y+h),(0,255,0),2) smile_output = cv2.cvtColor(test_image, cv2.COLOR_BGR2RGB)plt.imshow(smile_output)

Pada gambar di atas terdapat False Positive, Untuk memperbaikinya deteksi dilakukan didalam face, dengan melanjutkan kode diatas, berikut kode untuk melakukan smile detection

```
test_image = cv2.imread('Potrait.jpg')
for (x,y,w,h) in faces:
  for (x_s,y_s,w_s,h_s) in smiles:
    if ((x <= x_s) and (y <= y_s) and (x+w >= x_s+w_s) and (y+h >= y_s+h_s)):
      cv2.rectangle(test_image, (x_s,y_s),(x_s+w_s,y_s+h_s),(0,255,0),2)
smile_output = cv2.cvtColor(test_image, cv2.COLOR_BGR2RGB)
plt.imshow(smile_output)
```

test_image = cv2.imread('Potrait.jpg')
for (x,y,w,h) **in** faces:
for (x_s,y_s,w_s,h_s) **in** smiles:
if ((x <= x_s) **and** (y <= y_s) **and** (x+w >= x_s+w_s) **and** (y+h >=y_s+h_s)):
cv2.rectangle(test_image, (x_s,y_s),(x_s+w_s,y_s+h_s),(0,255,0),2)
smile_output = cv2.cvtColor(test_image, cv2.COLOR_BGR2RGB)plt.imshow(smile_output)

Eye Detection

Eye Detection menggunakan pre-trained model haarcascade_eye.xml untuk mendeteksimata pada gambar.

```
test_image = cv2.imread('Potrait.jpg')
eyes_cascade = cv2.CascadeClassifier('haarcascade_eye.xml')
eyes = eyes_cascade.detectMultiScale(grey, 1.1, 25)
for (x,y,w,h) in eyes:
     cv2.rectangle(test_image,(x,y),(x+w,y+h),(255,0,0),2)
eye_output = cv2.cvtColor(test_image, cv2.COLOR_BGR2RGB)
plt.imshow(eye_output)
```

test_image = cv2.imread('Potrait.jpg')
eyes_cascade = cv2.CascadeClassifier('haarcascade_eye.xml')eyes =
eyes_cascade.detectMultiScale(grey, 1.1, 25)
for (x,y,w,h) **in** eyes: cv2.rectangle(test_image,(x,y),(x+w,y+h),(255,0,0),2)
eye_output = cv2.cvtColor(test_image, cv2.COLOR_BGR2RGB)plt.imshow(eye_output)

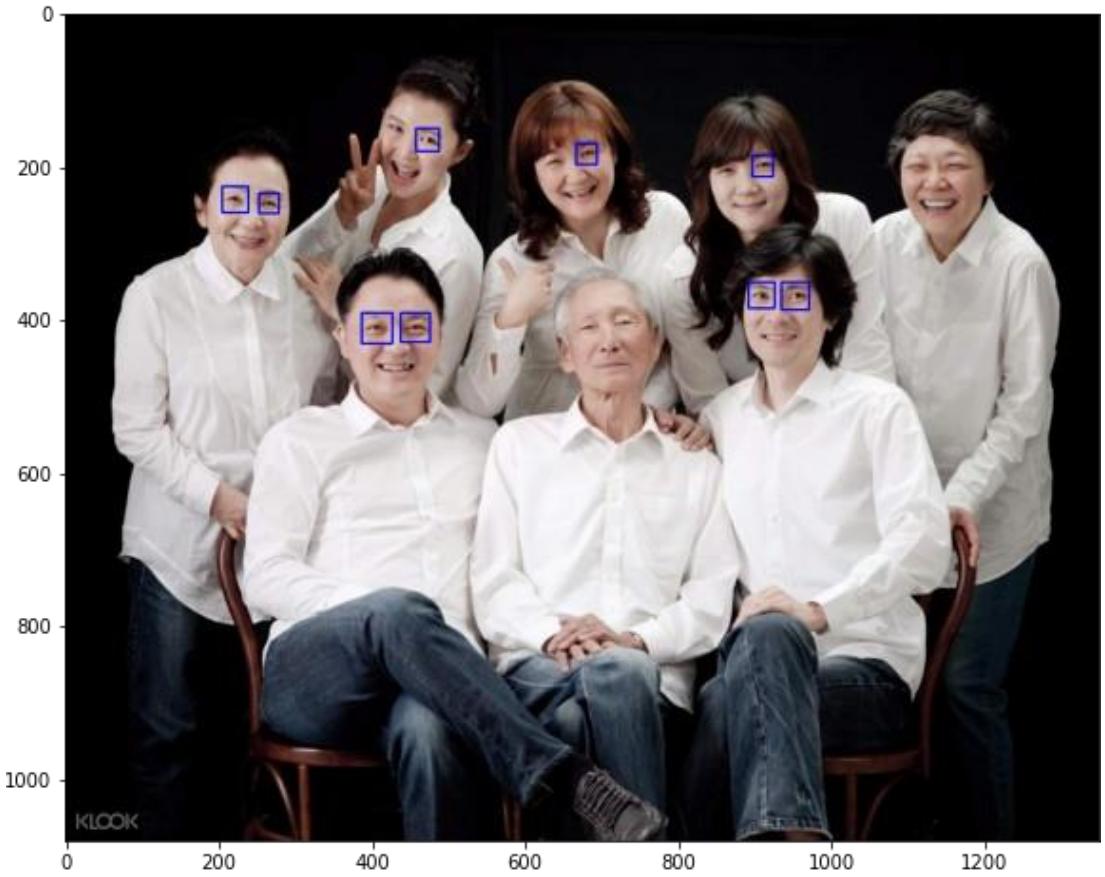
<matplotlib.image.AxesImage at 0x7f9e40fb79d0>

Apabila terdapat False Positive, untuk memperbaikinya deteksi dilakukan di dalam face,dengan melanjutkan kode diatas, berikut kode untuk melakukan eye detection

```
test_image = cv2.imread('Potrait.jpg')
for (x,y,w,h) in faces:
  for (x_s,y_s,w_s,h_s) in eyes:
    if ((x <= x_s) and (y <= y_s) and (x+w >= x_s+w_s) and (y+h >= y_s+h_s)):
      cv2.rectangle(test_image, (x_s,y_s),(x_s+w_s,y_s+h_s),(255,0,0),2)
eye_output = cv2.cvtColor(test_image, cv2.COLOR_BGR2RGB)
plt.imshow(eye_output)
```

```
test_image = cv2.imread('Potrait.jpg')
for (x,y,w,h) in faces:
    for (x_s,y_s,w_s,h_s) in eyes:
        if ((x <= x_s) and (y <= y_s) and (x+w >= x_s+w_s) and (y+h >=y_s+h_s)):
            cv2.rectangle(test_image, (x_s,y_s),(x_s+w_s,y_s+h_s),(255,0,0),2)
eye_output = cv2.cvtColor(test_image, cv2.COLOR_BGR2RGB)plt.imshow(eye_output)
```

```
<matplotlib.image.AxesImage at 0x7f9e40ba9950>
```



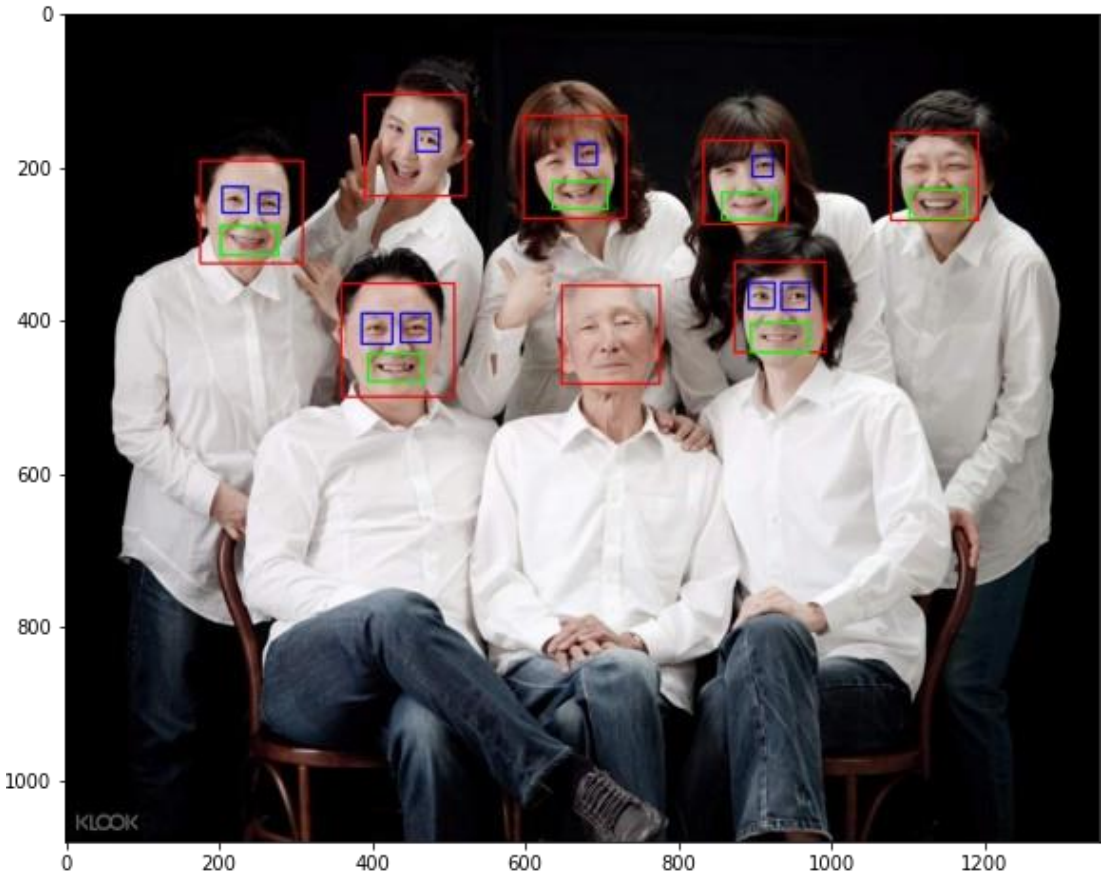
Menggabungkan Semuanya

Sekarang tinggal menggabungkan deteksi wajah, deteksi senyuman, dan deteksi mata.

```
test_image = cv2.imread('Potrait.jpg')
for (x,y,w,h) in faces:
  cv2.rectangle(test_image,(x,y),(x+w,y+h),(255,0,0),2)
  for (x_s,y_s,w_s,h_s) in eyes:
    if ((x <= x_s) and (y <= y_s) and (x+w >= x_s+w_s) and (y+h >= y_s+h_s)):
      cv2.rectangle(test_image, (x_s,y_s),(x_s+w_s,y_s+h_s),(255,255,255),2)
  for (x_s,y_s,w_s,h_s) in smiles:
    if ((x <= x_s) and (y <= y_s) and (x+w >= x_s+w_s) and (y+h >= y_s+h_s)):
      cv2.rectangle(test_image, (x_s,y_s),(x_s+w_s,y_s+h_s),(0,255,0),2)
head_output = cv2.cvtColor(test_image, cv2.COLOR_BGR2RGB)
plt.imshow(head_output)
```

```
test_image = cv2.imread('Potrait.jpg')
for (x,y,w,h) in faces: cv2.rectangle(test_image,(x,y),(x+w,y+h),(0,0,255),2)
    for (x_s,y_s,w_s,h_s) in smiles:
        if ((x <= x_s) and (y <= y_s) and (x+w >= x_s+w_s) and (y+h >=y_s+h_s)):
            cv2.rectangle(test_image, (x_s,y_s),(x_s+w_s,y_s+h_s),(0,255,0),2)
    for (x_s,y_s,w_s,h_s) in eyes:
        if ((x <= x_s) and (y <= y_s) and (x+w >= x_s+w_s) and (y+h >=y_s+h_s)):
            cv2.rectangle(test_image, (x_s,y_s),(x_s+w_s,y_s+h_s),(255,0,0),2)
head_output = cv2.cvtColor(test_image, cv2.COLOR_BGR2RGB)plt.imshow(head_output)
```

```
<matplotlib.image.AxesImage at 0x7f9e408b5e10>
```



Latihan

Nomer 1 : Face Detection

Download gambar dari link di bawah ini menggunakan wget.

https://web.kominfo.go.id/sites/default/files/kominfo-setkab-jokowi-Kabinet-Indonesia- Maju-1.jpg

Buatlah deteksi wajah dari gambar tersebut menggunakan pre-trained model HaarCascade.

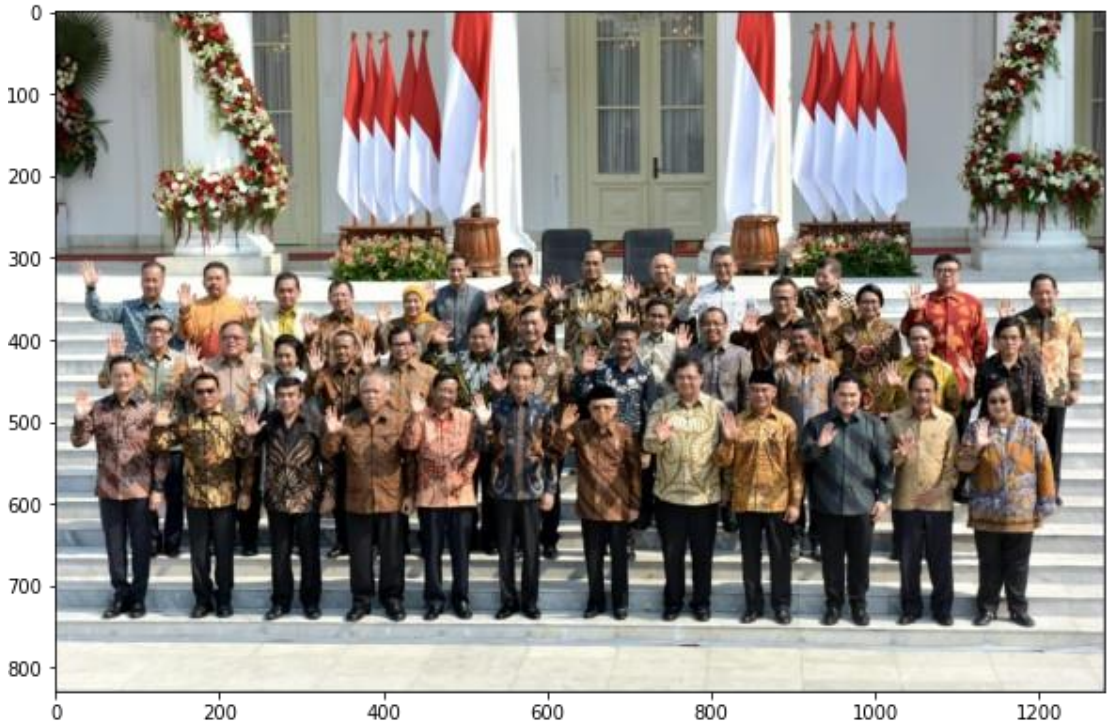
Objective : Dapat mendeteksi semua wajah dari gambar tersebut

Hint : Rubah nilai scale factor dan min neighbors hingga mendapatkan hasil terbaik.

!gdown https://web.kominfo.go.id/sites/default/files/kominfo-setkab-jokowi-Kabinet-Indonesia-Maju-1.jpg

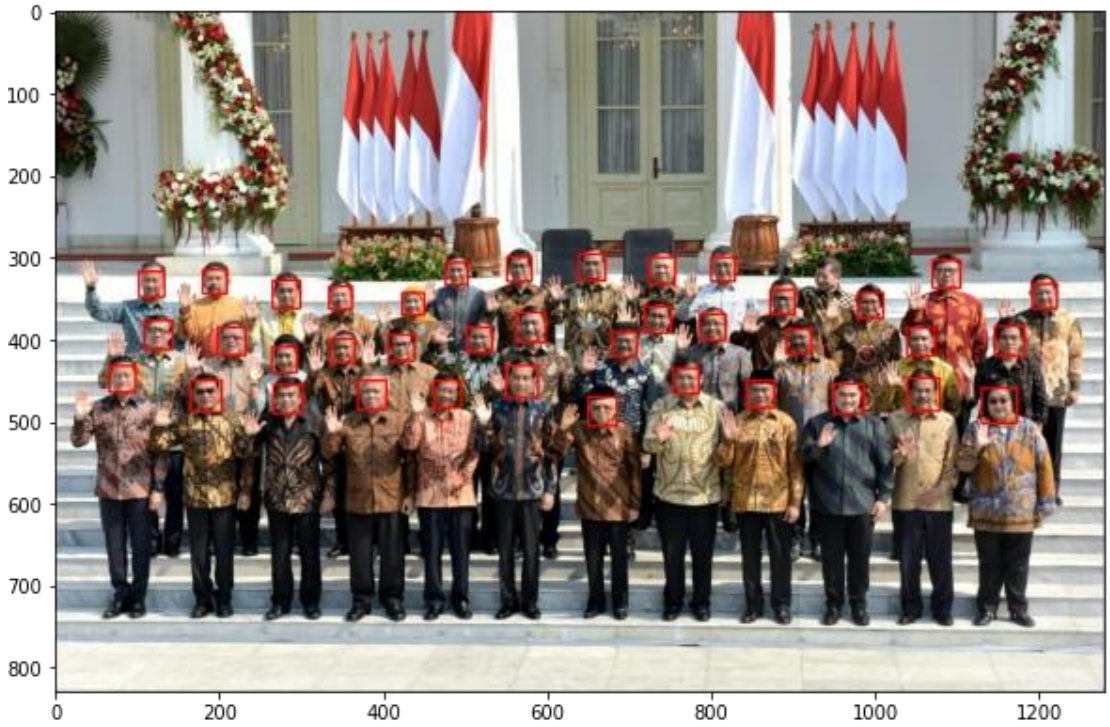
Downloading...
From: https://web.kominfo.go.id/sites/default/files/kominfo-setkab-jokowi-Kabinet-Indonesia-Maju-1.jpg

```
To: /content/kominfo-setkab-jokowi-Kabinet-Indonesia-Maju-1.jpg
      0% 0.00/216k [00:00<?, ?B/s] 100% 216k/216k [00:00<00:00, 24.9MB/s]

!wget --no-check-certificate \ https://web.kominfo.go.id/sites/default/files/kominfo-
        setkab-
jokowi-Kabinet-Indonesia-Maju-1.jpg

--2022-10-13 05:07:43--
https://web.kominfo.go.id/sites/default/files/kominfo-setkab-jokowi-Kabinet-Indonesia-
Maju-1.jpg
Resolving web.kominfo.go.id (web.kominfo.go.id)... 45.60.36.49Connecting to
web.kominfo.go.id (web.kominfo.go.id)| 45.60.36.49|:443... connected.
HTTP request sent, awaiting response... 200 OKLength: 216172 (211K)
[image/jpeg]
Saving to: ‘kominfo-setkab-jokowi-Kabinet-Indonesia-Maju-1.jpg.10’

                    kominfo-s         0%[                                 ]               0 --.-KB/s kominfo-
setkab-joko 100%[===================>] 211.11K   --.-KB/s          in0.02s

2022-10-13 05:07:43 (12.0 MB/s) - ‘kominfo-setkab-jokowi-Kabinet-Indonesia-Maju-
1.jpg.10’ saved [216172/216172]


import cv2 #opencv itself
import common #some useful opencv functions
import numpy as np # matrix manipulations

#the following are to do with this interactive notebook code
%matplotlib inline
from matplotlib import pyplot as plt # this lets you draw inlinepictures in the notebooks
import pylab # this allows you to control figure size pylab.rcParams['figure.figsize'] = (10.0,
8.0) # this controls figuresize in the notebook

base_image = cv2.imread('/content/kabinet Indonesia maju.jpg')grey =
cv2.cvtColor(base_image, cv2.COLOR_BGR2GRAY)
rgb = cv2.cvtColor(base_image, cv2.COLOR_BGR2RGB)
plt.imshow(rgb)

<matplotlib.image.AxesImage at 0x7f9e2fb6b690>
```




```
test_image = cv2.imread('/content/kabinet Indonesia maju.jpg')face_cascade =
cv2.CascadeClassifier('haarcascade_frontalface_default.xml') faces =
face_cascade.detectMultiScale(grey, 1.1, 5)
for (x,y,w,h) in faces: cv2.rectangle(test_image,(x,y),(x+w,y+h),(0,0,255),2)
face_output = cv2.cvtColor(test_image, cv2.COLOR_BGR2RGB)plt.imshow(face_output)
```

<matplotlib.image.AxesImage at 0x7f9e2f2e9350>



Nomer 2 : Plate Number Detection

Download gambar dari link di bawah ini menggunakan wget.

https://russiabusinesstoday.com/wp-content/uploads/2018/01/russia-cars.jpg

Buatlah deteksi plat nomor mobil dari gambar tersebut menggunakan pre-trained modelHaar Cascade.

Objective : Dapat mendeteksi plat nomor dari 3 mobil terdepan

Hint : Rubah nilai scale factor dan min neighbors hingga mendapatkan hasil terbaik.

```
!gdown https://russiabusinesstoday.com/wp-
content/uploads/2018/01/russia-cars.jpg
```

```
Downloading...
From:
https://russiabusinesstoday.com/wp-content/uploads/2018/01/russia-cars.jpg
To: /content/russia-cars.jpg
      0% 0.00/139k [00:00<?, ?B/s] 100% 139k/139k [00:00<00:00, 33.8MB/s]
```

```
!wget --no-check-certificate \
       https://russiabusinesstoday.com/wp-content/uploads/2018/01/russia-cars.jpg
```

```
--2022-10-13 04:49:46--
https://russiabusinesstoday.com/wp-content/uploads/2018/01/russia-
```

cars.jpg
Resolving russiabusinesstoday.com (russiabusinesstoday.com)...172.67.204.49, 104.21.22.104, 2606:4700:3033::6815:1668, ...
Connecting to russiabusinesstoday.com (russiabusinesstoday.com)|172.67.204.49|:443... connected.
HTTP request sent, awaiting response... 200 OKLength: 139042 (136K) [image/jpeg]
Saving to: ‘russia-cars.jpg.4’

russia-cars.jpg.4 0%[ ] 0 --.-KB/s russia-cars.jpg.4 100%[==========>] 135.78K --.-KB/s in 0.003s

2022-10-13 04:49:46 (47.7 MB/s) - ‘russia-cars.jpg.4’ saved[139042/139042]

```
base_plat = cv2.imread('/content/russia-cars.jpg')grey = cv2.cvtColor(base_plat, cv2.COLOR_BGR2GRAY)rgb = cv2.cvtColor(base_plat, cv2.COLOR_BGR2RGB) plt.imshow(rgb)
```

<matplotlib.image.AxesImage at 0x7f9e2feadb10>


```
test_plat = cv2.imread('/content/russia-cars.jpg')plat_cascade = cv2.CascadeClassifier('/content/haarcascade_russian_plate_number.xml')plats = plat_cascade.detectMultiScale(grey, 1.1, 5)
for (x,y,w,h) in plats: cv2.rectangle(test_plat,(x,y),(x+w,y+h),(0,0,255),2)
```

plat_output = cv2.cvtColor(test_plat, cv2.COLOR_BGR2RGB)plt.imshow(plat_output)

<matplotlib.image.AxesImage at 0x7f9e2fe27e90>