MODULE 12

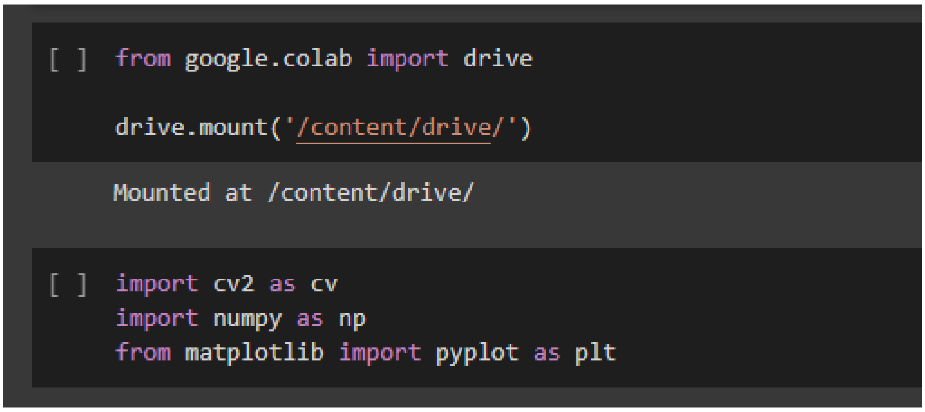
FACE DETECTION METHOD

Arranged By:

Rajendra Rakha Arya Prabaswara
(1941720080/20)

PROGRAM STUDI D-IV TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG

1. **Go to https://colab.research.google.com/ . After making sure that Google Colab is connected with your Github, create a new notebook and name it "Week1 2 .ipynb". Then import some libraries and access the folders in your Drive as usual .**

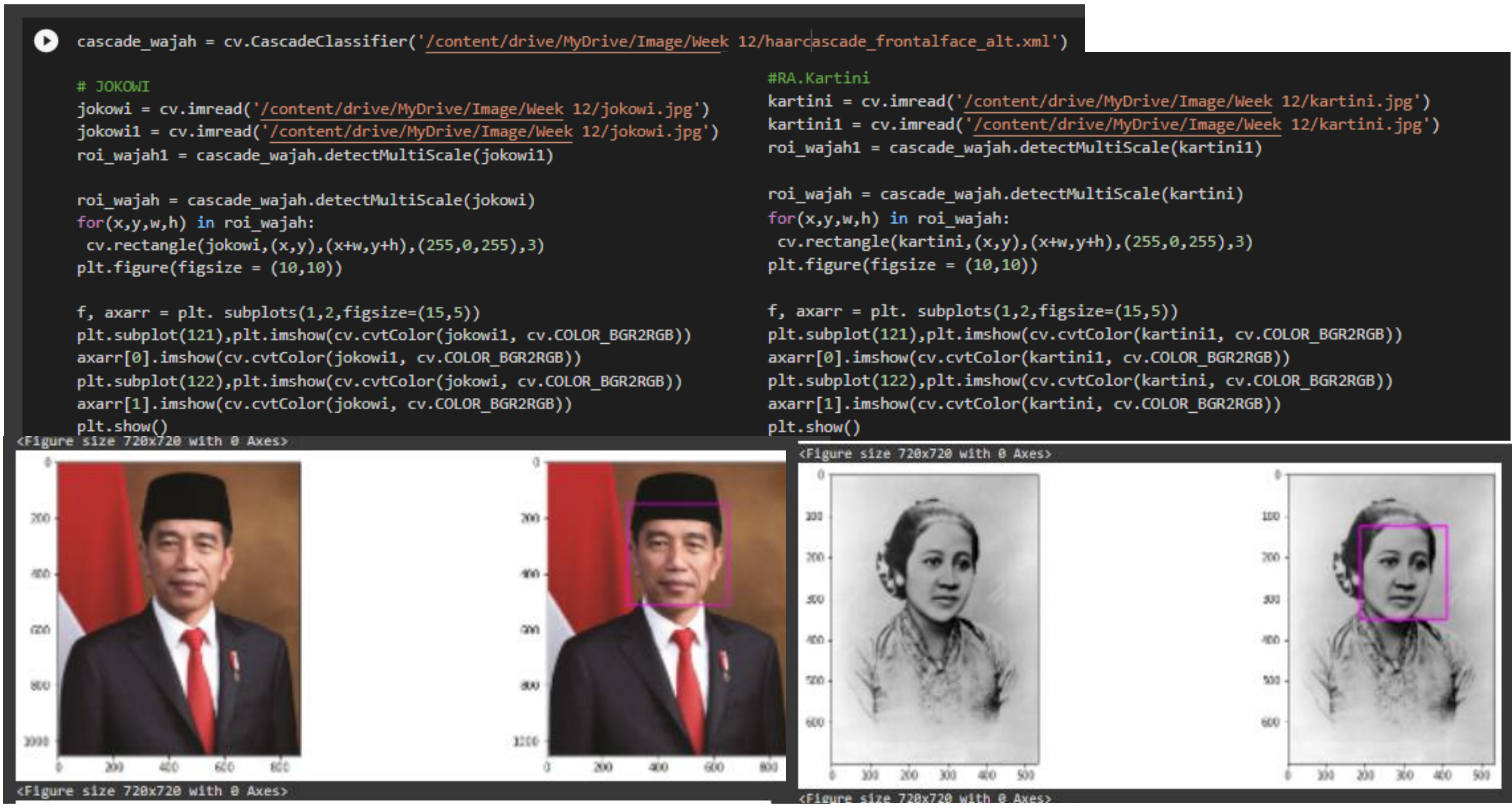


```
[ ]  from google.colab import drive

     drive.mount('/content/drive/')

     Mounted at /content/drive/

[ ]  import cv2 as cv
     import numpy as np
     from matplotlib import pyplot as plt
```

FULL CODE

https://github.com/Rjndrkha/PCVK_Genap_2022

2. **Perform Face Detection for other image objects available at (/ images / facedet) as shown in the following example. Pay attention to the face detection results above. Overall, face detection can be done well, even for images in the form of non-photo images, masked faces, or small faces (solvay).**

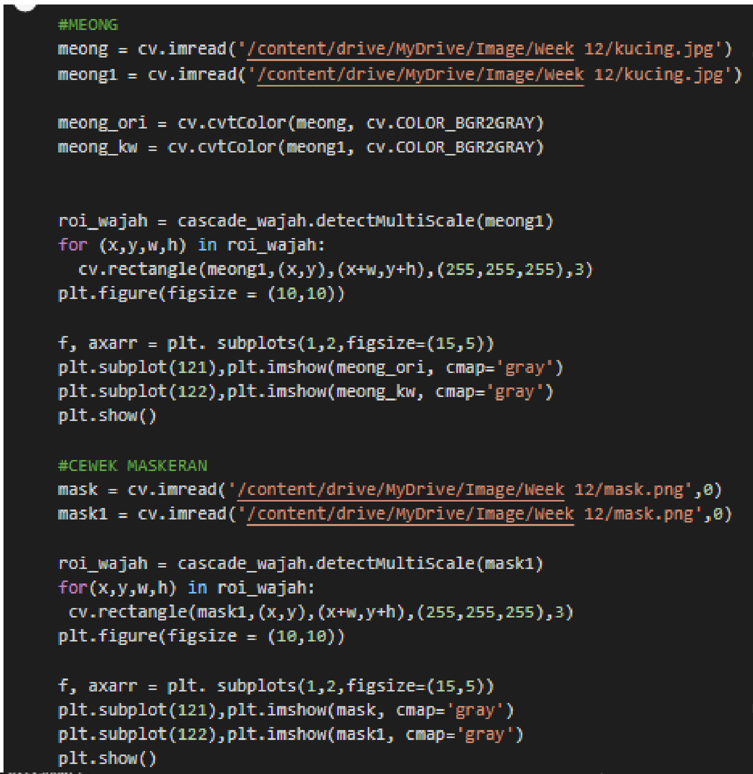


```
cascade_wajah = cv.CascadeClassifier('/content/drive/MyDrive/Image/Week 12/haarcascade_frontalface_alt.xml')

# JOKOWI
jokowi = cv.imread('/content/drive/MyDrive/Image/Week 12/jokowi.jpg')
jokowi1 = cv.imread('/content/drive/MyDrive/Image/Week 12/jokowi.jpg')
roi_wajah1 = cascade_wajah.detectMultiScale(jokowi1)

roi_wajah = cascade_wajah.detectMultiScale(jokowi)
for(x,y,w,h) in roi_wajah:
 cv.rectangle(jokowi,(x,y),(x+w,y+h),(255,0,255),3)
plt.figure(figsize = (10,10))

f, axarr = plt. subplots(1,2,figsize=(15,5))
plt.subplot(121),plt.imshow(cv.cvtColor(jokowi1, cv.COLOR_BGR2RGB))
axarr[0].imshow(cv.cvtColor(jokowi1, cv.COLOR_BGR2RGB))
plt.subplot(122),plt.imshow(cv.cvtColor(jokowi, cv.COLOR_BGR2RGB))
axarr[1].imshow(cv.cvtColor(jokowi, cv.COLOR_BGR2RGB))
plt.show()
<Figure size 720x720 with 0 Axes>
```

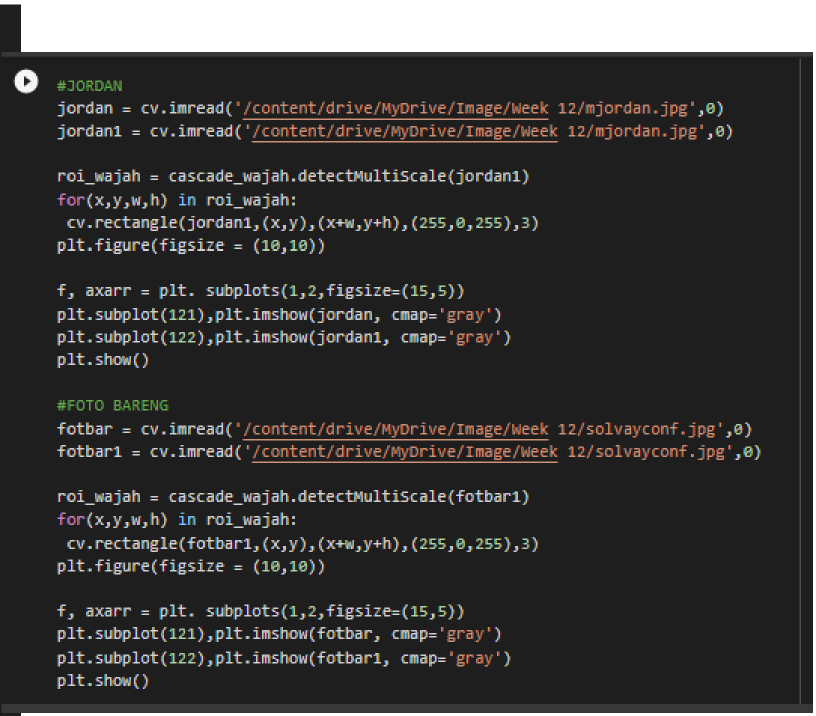
```
<Figure size 720x720 with 0 Axes>
```

```
#RA.Kartini
kartini = cv.imread('/content/drive/MyDrive/Image/Week 12/kartini.jpg')
kartini1 = cv.imread('/content/drive/MyDrive/Image/Week 12/kartini.jpg')
roi_wajah1 = cascade_wajah.detectMultiScale(kartini1)

roi_wajah = cascade_wajah.detectMultiScale(kartini)
for(x,y,w,h) in roi_wajah:
 cv.rectangle(kartini,(x,y),(x+w,y+h),(255,0,255),3)
plt.figure(figsize = (10,10))

f, axarr = plt. subplots(1,2,figsize=(15,5))
plt.subplot(121),plt.imshow(cv.cvtColor(kartini1, cv.COLOR_BGR2RGB))
axarr[0].imshow(cv.cvtColor(kartini1, cv.COLOR_BGR2RGB))
plt.subplot(122),plt.imshow(cv.cvtColor(kartini, cv.COLOR_BGR2RGB))
axarr[1].imshow(cv.cvtColor(kartini, cv.COLOR_BGR2RGB))
plt.show()
<Figure size 720x720 with 0 Axes>
```

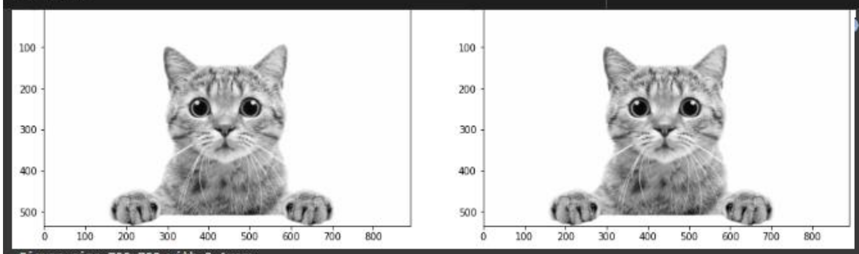
```
<Figure size 720x720 with 0 Axes>
```



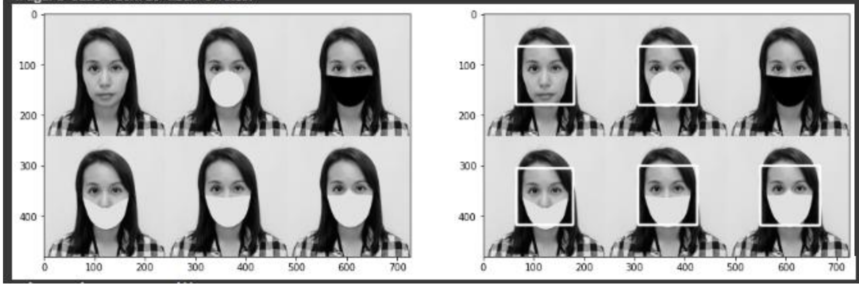
```
#MEONG
meong = cv.imread('/content/drive/MyDrive/Image/Week 12/kucing.jpg')
meong1 = cv.imread('/content/drive/MyDrive/Image/Week 12/kucing.jpg')

meong_ori = cv.cvtColor(meong, cv.COLOR_BGR2GRAY)
meong_kw = cv.cvtColor(meong1, cv.COLOR_BGR2GRAY)


roi_wajah = cascade_wajah.detectMultiScale(meong1)
for (x,y,w,h) in roi_wajah:
  cv.rectangle(meong1,(x,y),(x+w,y+h),(255,255,255),3)
plt.figure(figsize = (10,10))

f, axarr = plt. subplots(1,2,figsize=(15,5))
plt.subplot(121),plt.imshow(meong_ori, cmap='gray')
plt.subplot(122),plt.imshow(meong_kw, cmap='gray')
plt.show()

#CEWEK MASKERAN
mask = cv.imread('/content/drive/MyDrive/Image/Week 12/mask.png',0)
mask1 = cv.imread('/content/drive/MyDrive/Image/Week 12/mask.png',0)

roi_wajah = cascade_wajah.detectMultiScale(mask1)
for(x,y,w,h) in roi_wajah:
 cv.rectangle(mask1,(x,y),(x+w,y+h),(255,255,255),3)
plt.figure(figsize = (10,10))

f, axarr = plt. subplots(1,2,figsize=(15,5))
plt.subplot(121),plt.imshow(mask, cmap='gray')
plt.subplot(122),plt.imshow(mask1, cmap='gray')
plt.show()
```



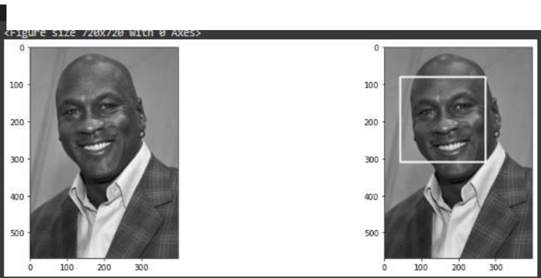
```
#JORDAN
jordan = cv.imread('/content/drive/MyDrive/Image/Week 12/mjordan.jpg',0)
jordan1 = cv.imread('/content/drive/MyDrive/Image/Week 12/mjordan.jpg',0)

roi_wajah = cascade_wajah.detectMultiScale(jordan1)
for(x,y,w,h) in roi_wajah:
 cv.rectangle(jordan1,(x,y),(x+w,y+h),(255,0,255),3)
plt.figure(figsize = (10,10))

f, axarr = plt. subplots(1,2,figsize=(15,5))
plt.subplot(121),plt.imshow(jordan, cmap='gray')
plt.subplot(122),plt.imshow(jordan1, cmap='gray')
plt.show()

#FOTO BARENG
fotbar = cv.imread('/content/drive/MyDrive/Image/Week 12/solvayconf.jpg',0)
fotbar1 = cv.imread('/content/drive/MyDrive/Image/Week 12/solvayconf.jpg',0)

roi_wajah = cascade_wajah.detectMultiScale(fotbar1)
for(x,y,w,h) in roi_wajah:
 cv.rectangle(fotbar1,(x,y),(x+w,y+h),(255,0,255),3)
plt.figure(figsize = (10,10))

f, axarr = plt. subplots(1,2,figsize=(15,5))
plt.subplot(121),plt.imshow(fotbar, cmap='gray')
plt.subplot(122),plt.imshow(fotbar1, cmap='gray')
plt.show()
```



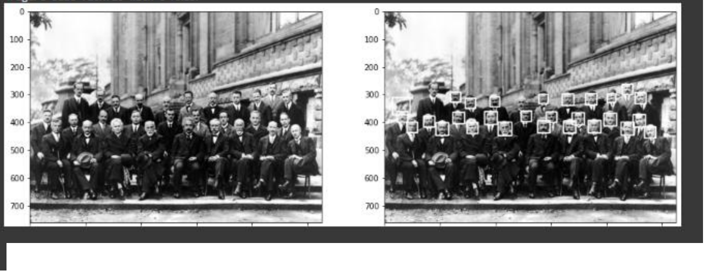
```
<Figure size 720x720 with 0 Axes>
```



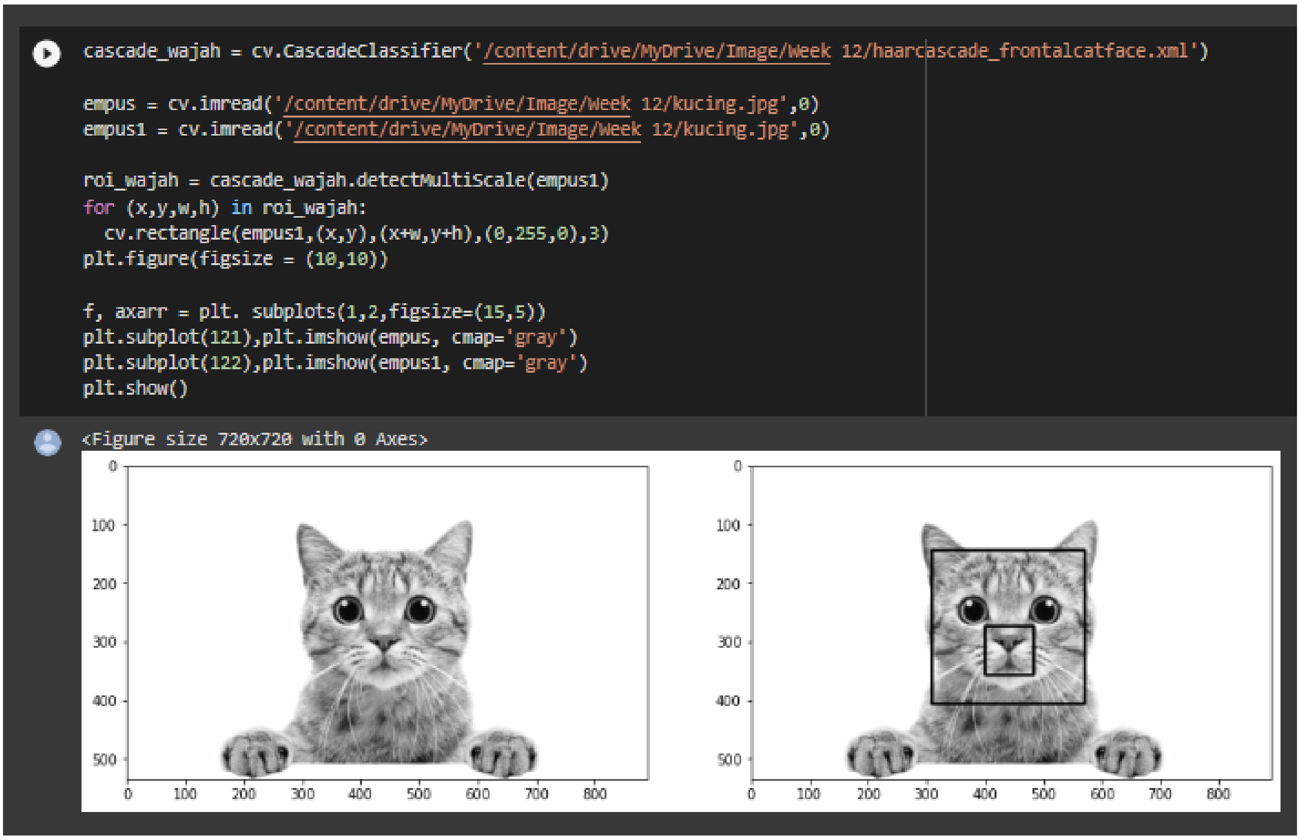
```
<Figure size 720x720 with 0 Axes>
```



```
<Figure size 720x720 with 0 Axes>
```



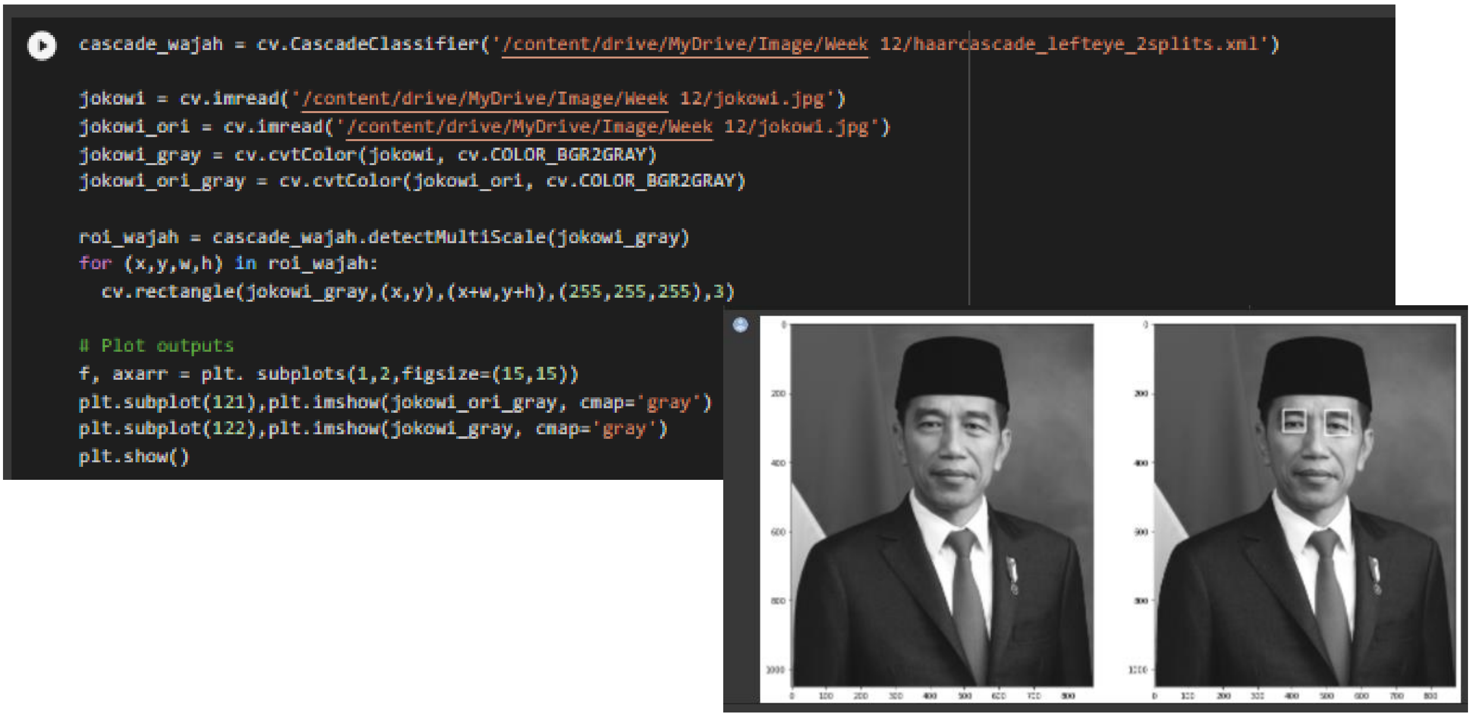
3. **In Problem No.2. cat faces cannot be detected properly. Detect the cat's face until a rectangle appears on the face. For instructions on this problem, pay attention to the pretrained features that OpenCV has provided. Use the existing xml if one is provided. If you don't have it, try searching by searching through search engines.**



```
cascade_wajah = cv.CascadeClassifier('/content/drive/MyDrive/Image/Week 12/haarcascade_frontalcatface.xml')

empus = cv.imread('/content/drive/MyDrive/Image/Week 12/kucing.jpg',0)
empus1 = cv.imread('/content/drive/MyDrive/Image/Week 12/kucing.jpg',0)

roi_wajah = cascade_wajah.detectMultiScale(empus1)
for (x,y,w,h) in roi_wajah:
  cv.rectangle(empus1,(x,y),(x+w,y+h),(0,255,0),3)
plt.figure(figsize = (10,10))

f, axarr = plt. subplots(1,2,figsize=(15,5))
plt.subplot(121),plt.imshow(empus, cmap='gray')
plt.subplot(122),plt.imshow(empus1, cmap='gray')
plt.show()
```

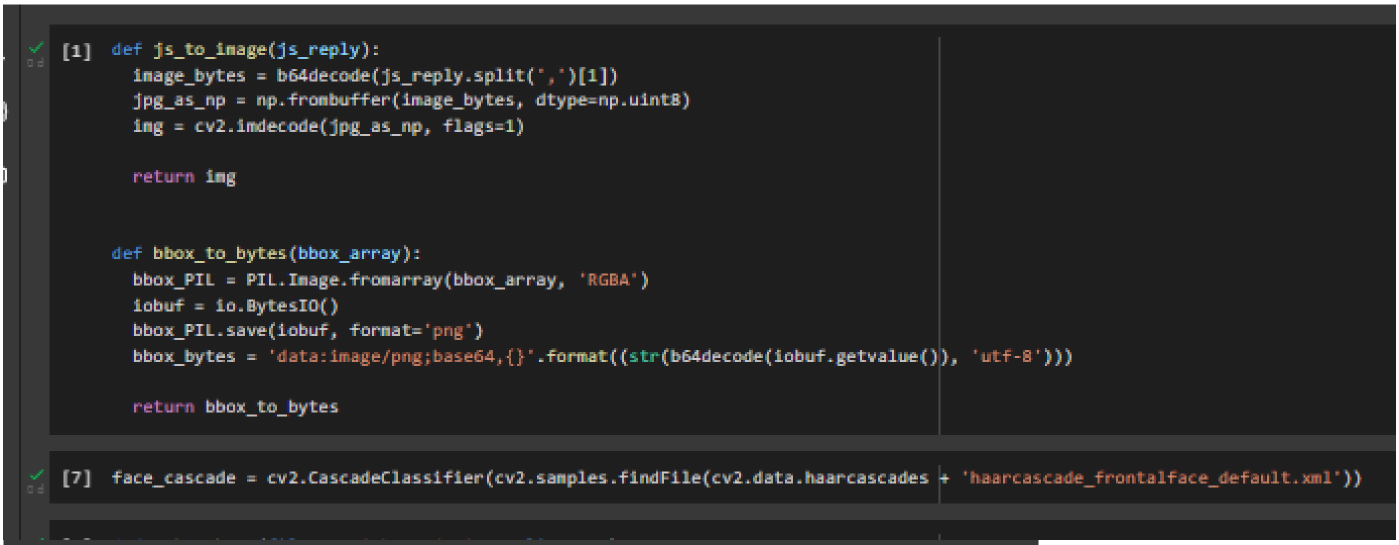
```
<Figure size 720x720 with 0 Axes>
```

4. **Also try it for an eyes detection.**



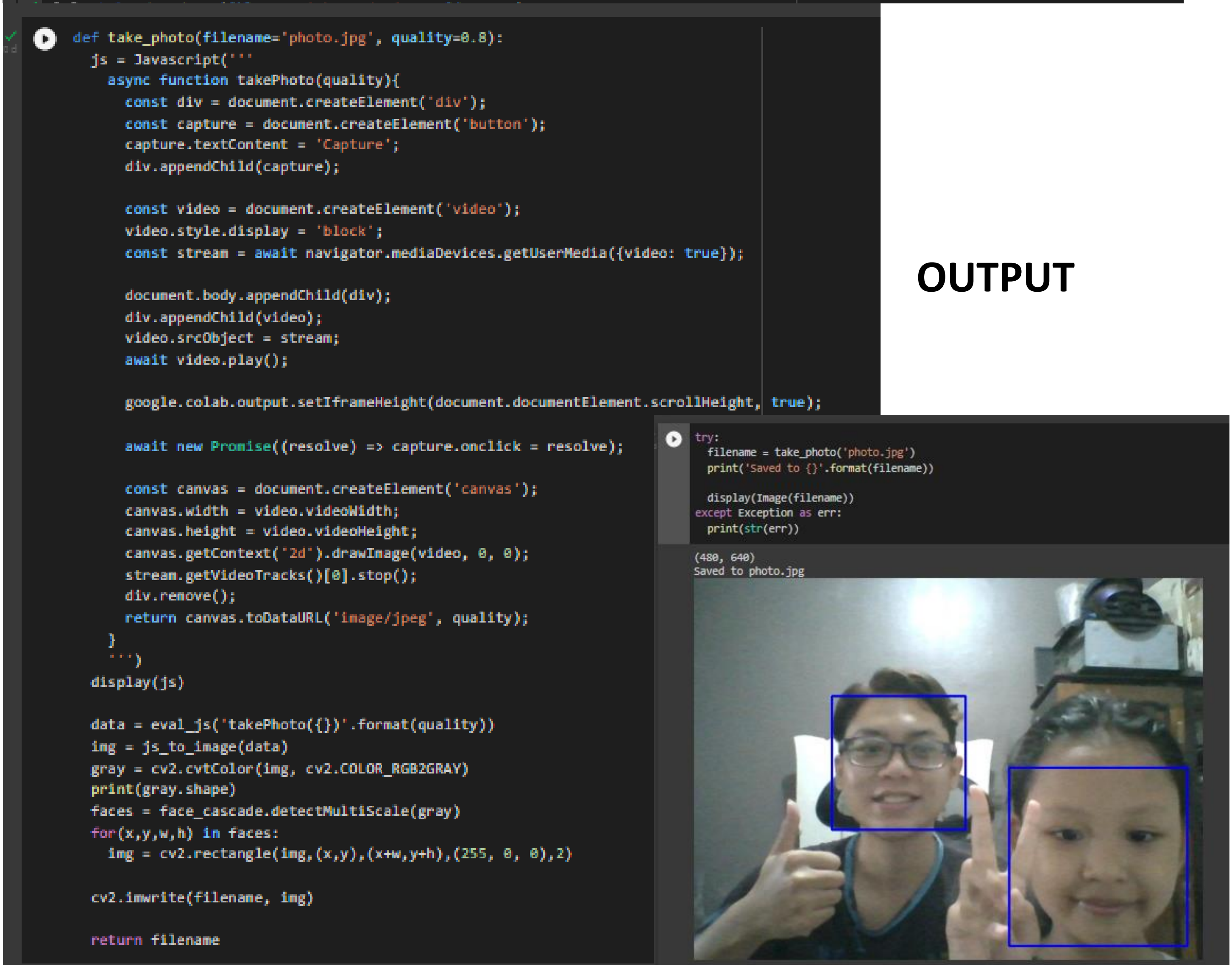
```
cascade_wajah = cv.CascadeClassifier('/content/drive/MyDrive/Image/Week 12/haarcascade_lefteye_2splits.xml')

jokowi = cv.imread('/content/drive/MyDrive/Image/Week 12/jokowi.jpg')
jokowi_ori = cv.imread('/content/drive/MyDrive/Image/Week 12/jokowi.jpg')
jokowi_gray = cv.cvtColor(jokowi, cv.COLOR_BGR2GRAY)
jokowi_ori_gray = cv.cvtColor(jokowi_ori, cv.COLOR_BGR2GRAY)

roi_wajah = cascade_wajah.detectMultiScale(jokowi_gray)
for (x,y,w,h) in roi_wajah:
  cv.rectangle(jokowi_gray,(x,y),(x+w,y+h),(255,255,255),3)

# Plot outputs
f, axarr = plt. subplots(1,2,figsize=(15,15))
plt.subplot(121),plt.imshow(jokowi_ori_gray, cmap='gray')
plt.subplot(122),plt.imshow(jokowi_gray, cmap='gray')
plt.show()
```

5. **Implement Face Tracking using Google Colab. Instructions, complete tutorials on camera access and FaceDetection on google colab can be seen at the following link: https://www.youtube.com/watch?v=YjWh7QvVH60**



```
[1] def js_to_image(js_reply):
      image_bytes = b64decode(js_reply.split(',')[1])
      jpg_as_np = np.frombuffer(image_bytes, dtype=np.uint8)
      img = cv2.imdecode(jpg_as_np, flags=1)

      return img


    def bbox_to_bytes(bbox_array):
      bbox_PIL = PIL.Image.fromarray(bbox_array, 'RGBA')
      iobuf = io.BytesIO()
      bbox_PIL.save(iobuf, format='png')
      bbox_bytes = 'data:image/png;base64,{}'.format((str(b64decode(iobuf.getvalue()), 'utf-8')))

      return bbox_to_bytes
```

```
[7] face_cascade = cv2.CascadeClassifier(cv2.samples.findFile(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml'))
```



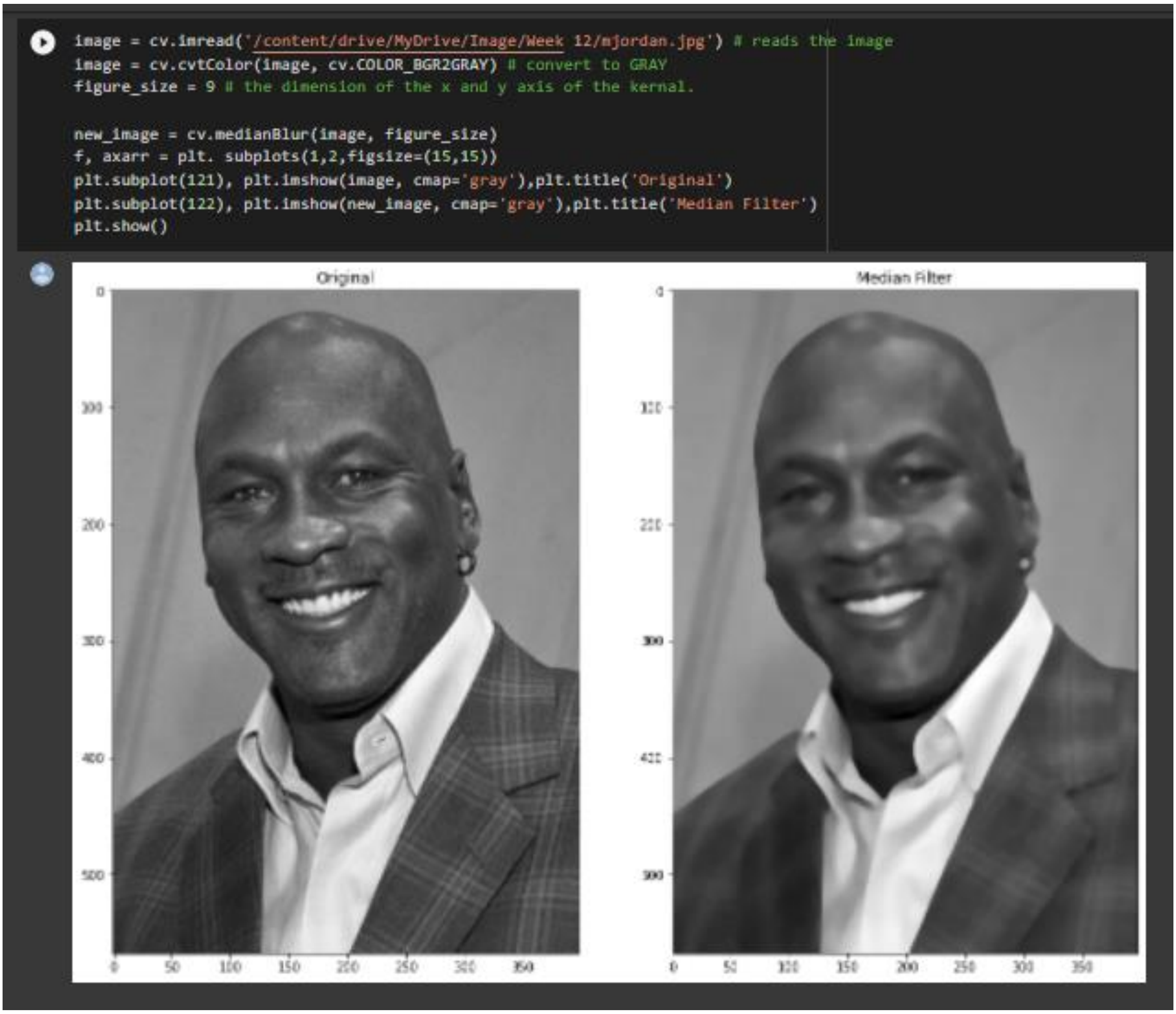
```
def take_photo(filename='photo.jpg', quality=0.8):
  js = Javascript('''
    async function takePhoto(quality){
      const div = document.createElement('div');
      const capture = document.createElement('button');
      capture.textContent = 'Capture';
      div.appendChild(capture);

      const video = document.createElement('video');
      video.style.display = 'block';
      const stream = await navigator.mediaDevices.getUserMedia({video: true});

      document.body.appendChild(div);
      div.appendChild(video);
      video.srcObject = stream;
      await video.play();

      google.colab.output.setIframeHeight(document.documentElement.scrollHeight, true);

      await new Promise((resolve) => capture.onclick = resolve);

      const canvas = document.createElement('canvas');
      canvas.width = video.videoWidth;
      canvas.height = video.videoHeight;
      canvas.getContext('2d').drawImage(video, 0, 0);
      stream.getVideoTracks()[0].stop();
      div.remove();
      return canvas.toDataURL('image/jpeg', quality);
    }
    ''')
  display(js)

  data = eval_js('takePhoto({})'.format(quality))
  img = js_to_image(data)
  gray = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)
  print(gray.shape)
  faces = face_cascade.detectMultiScale(gray)
  for(x,y,w,h) in faces:
    img = cv2.rectangle(img,(x,y),(x+w,y+h),(255, 0, 0),2)

  cv2.imwrite(filename, img)

  return filename
```

OUTPUT

```
try:
  filename = take_photo('photo.jpg')
  print('Saved to {}'.format(filename))

  display(Image(filename))
except Exception as err:
  print(str(err))
```

```
(480, 640)
Saved to photo.jpg
```

6. **Blur the detected part of the face. Here's an example of the output. Hint: you can use cv.medianBlur to give a blur effect.**



```
image = cv.imread('/content/drive/MyDrive/Image/Week 12/mjordan.jpg') # reads the image
image = cv.cvtColor(image, cv.COLOR_BGR2GRAY) # convert to GRAY
figure_size = 9 # the dimension of the x and y axis of the kernal.

new_image = cv.medianBlur(image, figure_size)
f, axarr = plt. subplots(1,2,figsize=(15,15))
plt.subplot(121), plt.imshow(image, cmap='gray'),plt.title('Original')
plt.subplot(122), plt.imshow(new_image, cmap='gray'),plt.title('Median Filter')
plt.show()
```