Jurusan Teknologi Informasi Politeknik Negeri Malang

Tugas Minggu-16 : Feature Matching, Face Detection, Face Tracking

Mata Kuliah Pengolahan Citra dan Visi Komputer

Pengampu: Dr. Eng CAHYA RAHMAD., ST.,M.KOM.

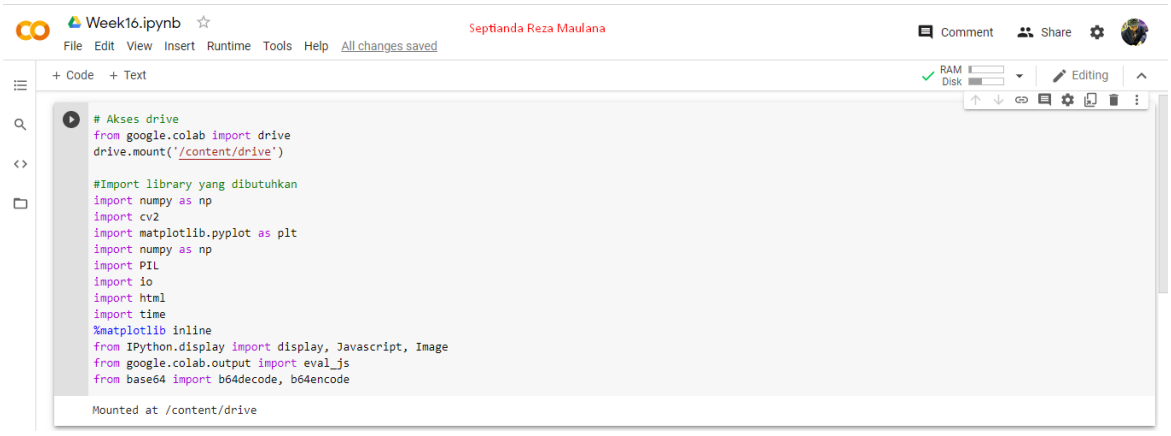
Juni 2021

Tujuan

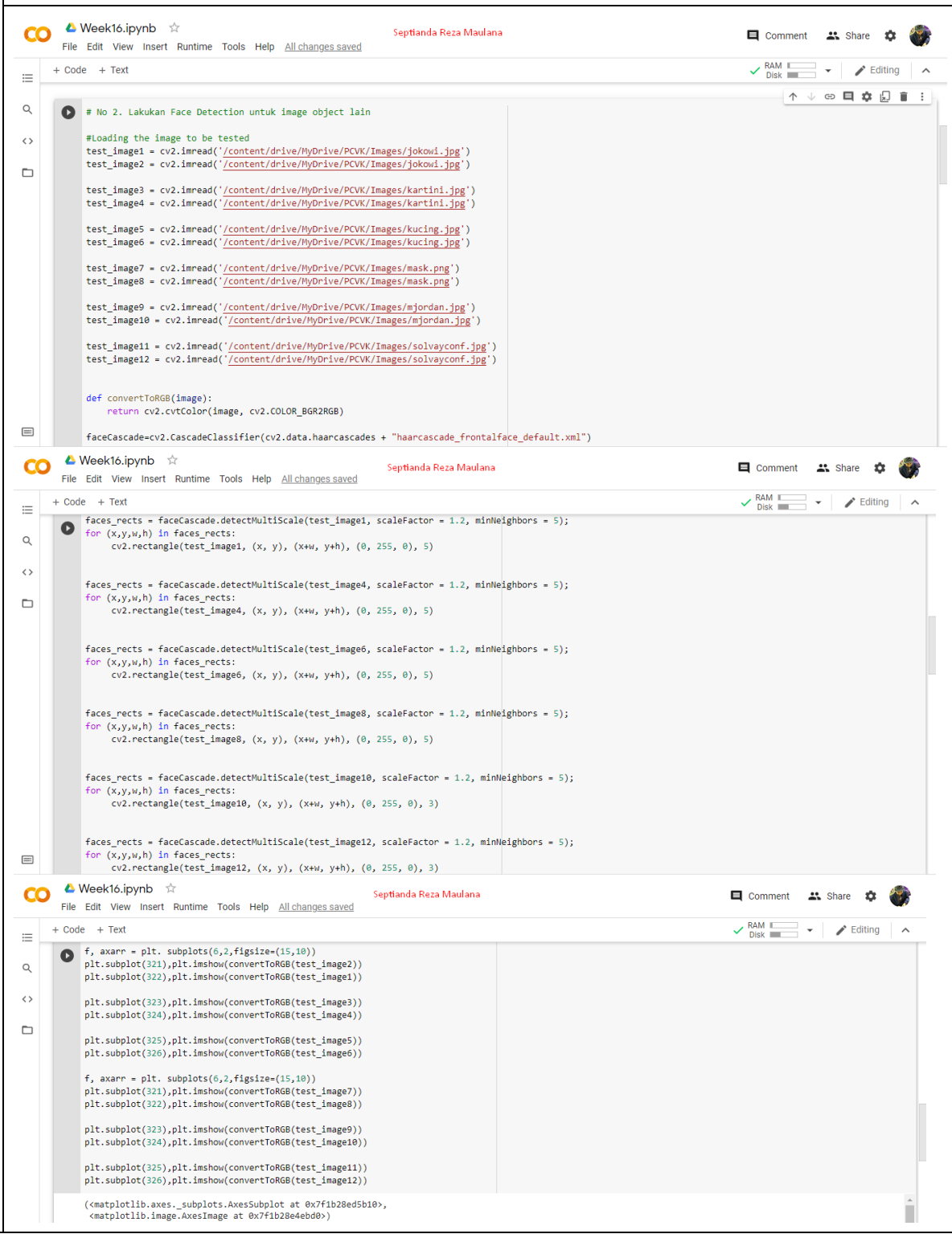
1. Mahasiswa mampu memahami konsep Feature Matching.
2. Mahasiswa mampu memahami konsep face detection.
3. Mahasiswa dapat mengimplementasikan beberapa metode dalam proses Feature Matching dan Face Detection menggunakan Python pada Google Colab.

Tugas Praktikum

Langkah	Keterangan
1	Buka https://colab.research.google.com/ . Setelah dipastikan bahwa google Colab terhubung dengan Github Anda, buat notebook baru dan beri nama “Week12.ipynb”. Kemudian import beberapa library dan akses folder yang ada di Drive Anda dengan seperti biasa.



2	Lakukan Face Detection untuk image object lain yang tersedia pada (/images/facedet). Tampilkan seperti pada contoh berikut.



Week16.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Septianda Reza Maulana

Comment Share Editing

+ Code + Text

```
# No 2. Lakukan Face Detection untuk image object lain

#Loading the image to be tested
test_image1 = cv2.imread('/content/drive/MyDrive/PCVK/Images/jokowi.jpg')
test_image2 = cv2.imread('/content/drive/MyDrive/PCVK/Images/jokowi.jpg')

test_image3 = cv2.imread('/content/drive/MyDrive/PCVK/Images/kartini.jpg')
test_image4 = cv2.imread('/content/drive/MyDrive/PCVK/Images/kartini.jpg')

test_image5 = cv2.imread('/content/drive/MyDrive/PCVK/Images/kucing.jpg')
test_image6 = cv2.imread('/content/drive/MyDrive/PCVK/Images/kucing.jpg')

test_image7 = cv2.imread('/content/drive/MyDrive/PCVK/Images/mask.png')
test_image8 = cv2.imread('/content/drive/MyDrive/PCVK/Images/mask.png')

test_image9 = cv2.imread('/content/drive/MyDrive/PCVK/Images/mjordan.jpg')
test_image10 = cv2.imread('/content/drive/MyDrive/PCVK/Images/mjordan.jpg')

test_image11 = cv2.imread('/content/drive/MyDrive/PCVK/Images/solvayconf.jpg')
test_image12 = cv2.imread('/content/drive/MyDrive/PCVK/Images/solvayconf.jpg')


def convertToRGB(image):
    return cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

faceCascade=cv2.CascadeClassifier(cv2.data.haarcascades + "haarcascade_frontalface_default.xml")
```

Week16.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Septianda Reza Maulana

Comment Share Editing

+ Code + Text

```
faces_rects = faceCascade.detectMultiScale(test_image1, scaleFactor = 1.2, minNeighbors = 5);
for (x,y,w,h) in faces_rects:
     cv2.rectangle(test_image1, (x, y), (x+w, y+h), (0, 255, 0), 5)


faces_rects = faceCascade.detectMultiScale(test_image4, scaleFactor = 1.2, minNeighbors = 5);
for (x,y,w,h) in faces_rects:
     cv2.rectangle(test_image4, (x, y), (x+w, y+h), (0, 255, 0), 5)


faces_rects = faceCascade.detectMultiScale(test_image6, scaleFactor = 1.2, minNeighbors = 5);
for (x,y,w,h) in faces_rects:
     cv2.rectangle(test_image6, (x, y), (x+w, y+h), (0, 255, 0), 5)


faces_rects = faceCascade.detectMultiScale(test_image8, scaleFactor = 1.2, minNeighbors = 5);
for (x,y,w,h) in faces_rects:
     cv2.rectangle(test_image8, (x, y), (x+w, y+h), (0, 255, 0), 5)


faces_rects = faceCascade.detectMultiScale(test_image10, scaleFactor = 1.2, minNeighbors = 5);
for (x,y,w,h) in faces_rects:
     cv2.rectangle(test_image10, (x, y), (x+w, y+h), (0, 255, 0), 3)


faces_rects = faceCascade.detectMultiScale(test_image12, scaleFactor = 1.2, minNeighbors = 5);
for (x,y,w,h) in faces_rects:
     cv2.rectangle(test_image12, (x, y), (x+w, y+h), (0, 255, 0), 3)
```

Week16.ipynb

File Edit View Insert Runtime Tools Help All changes saved

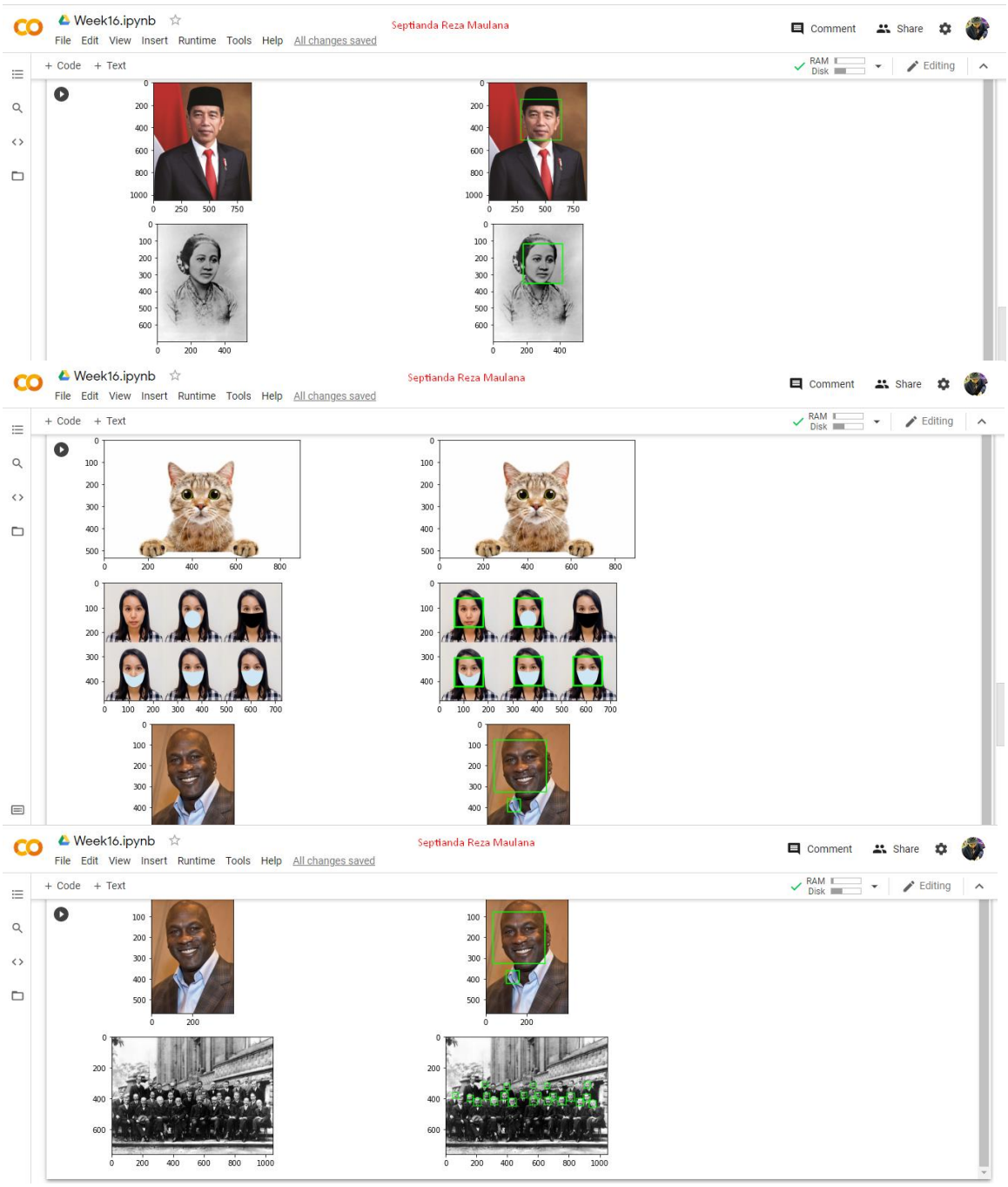
Septianda Reza Maulana

Comment Share Editing

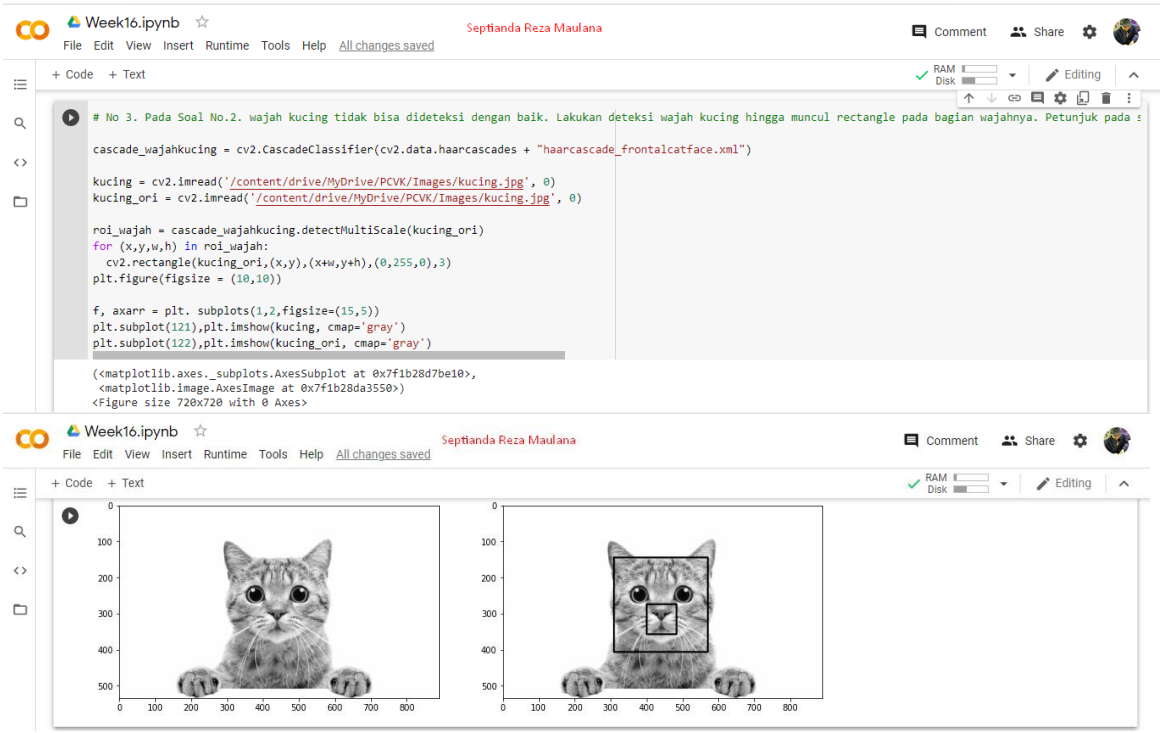
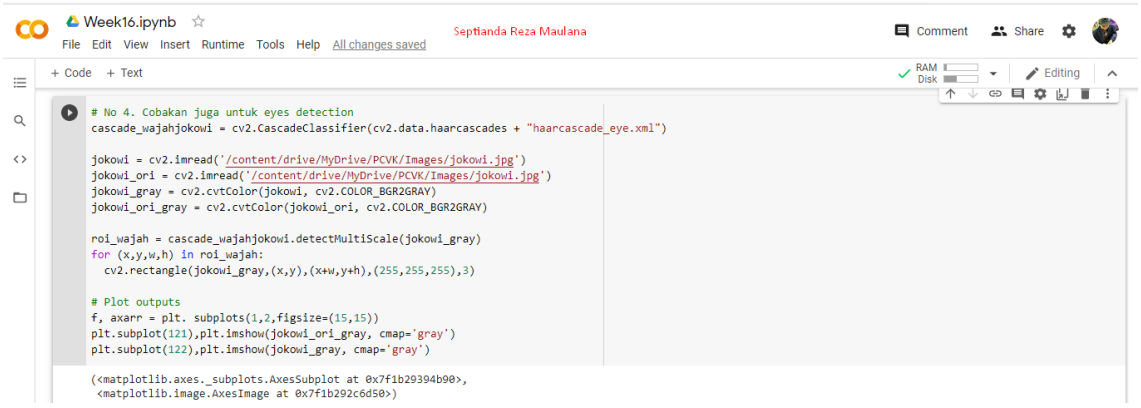
+ Code + Text

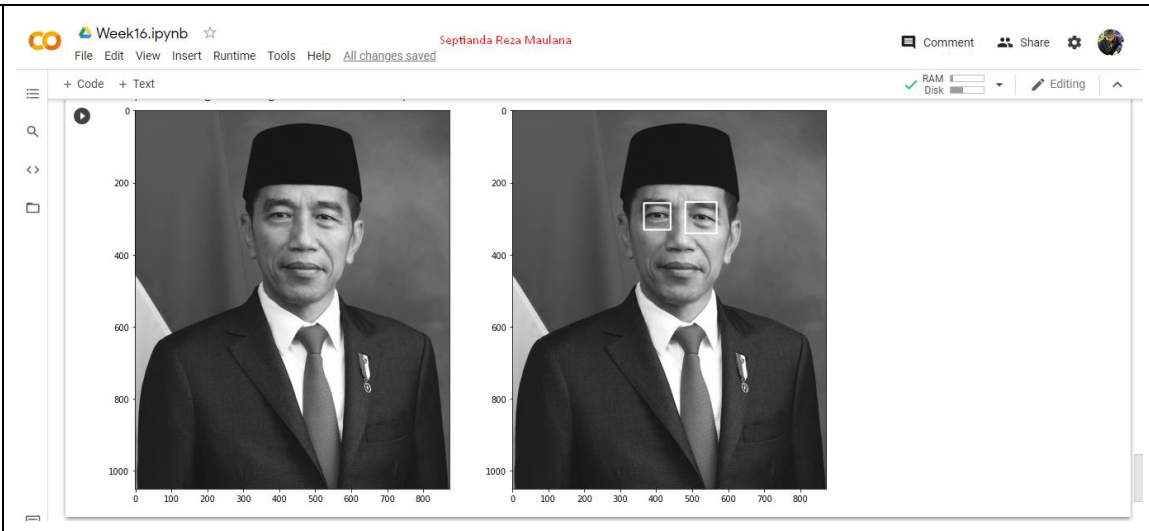
```
f, axarr = plt. subplots(6,2,figsize=(15,10))
plt.subplot(321),plt.imshow(convertToRGB(test_image2))
plt.subplot(322),plt.imshow(convertToRGB(test_image1))

plt.subplot(323),plt.imshow(convertToRGB(test_image3))
plt.subplot(324),plt.imshow(convertToRGB(test_image4))

plt.subplot(325),plt.imshow(convertToRGB(test_image5))
plt.subplot(326),plt.imshow(convertToRGB(test_image6))

f, axarr = plt. subplots(6,2,figsize=(15,10))
plt.subplot(321),plt.imshow(convertToRGB(test_image7))
plt.subplot(322),plt.imshow(convertToRGB(test_image8))

plt.subplot(323),plt.imshow(convertToRGB(test_image9))
plt.subplot(324),plt.imshow(convertToRGB(test_image10))

plt.subplot(325),plt.imshow(convertToRGB(test_image11))
plt.subplot(326),plt.imshow(convertToRGB(test_image12))
```

```
(<matplotlib.axes._subplots.AxesSubplot at 0x7f1b28ed5b10>,
 <matplotlib.image.AxesImage at 0x7f1b28e4ebd0>)
```

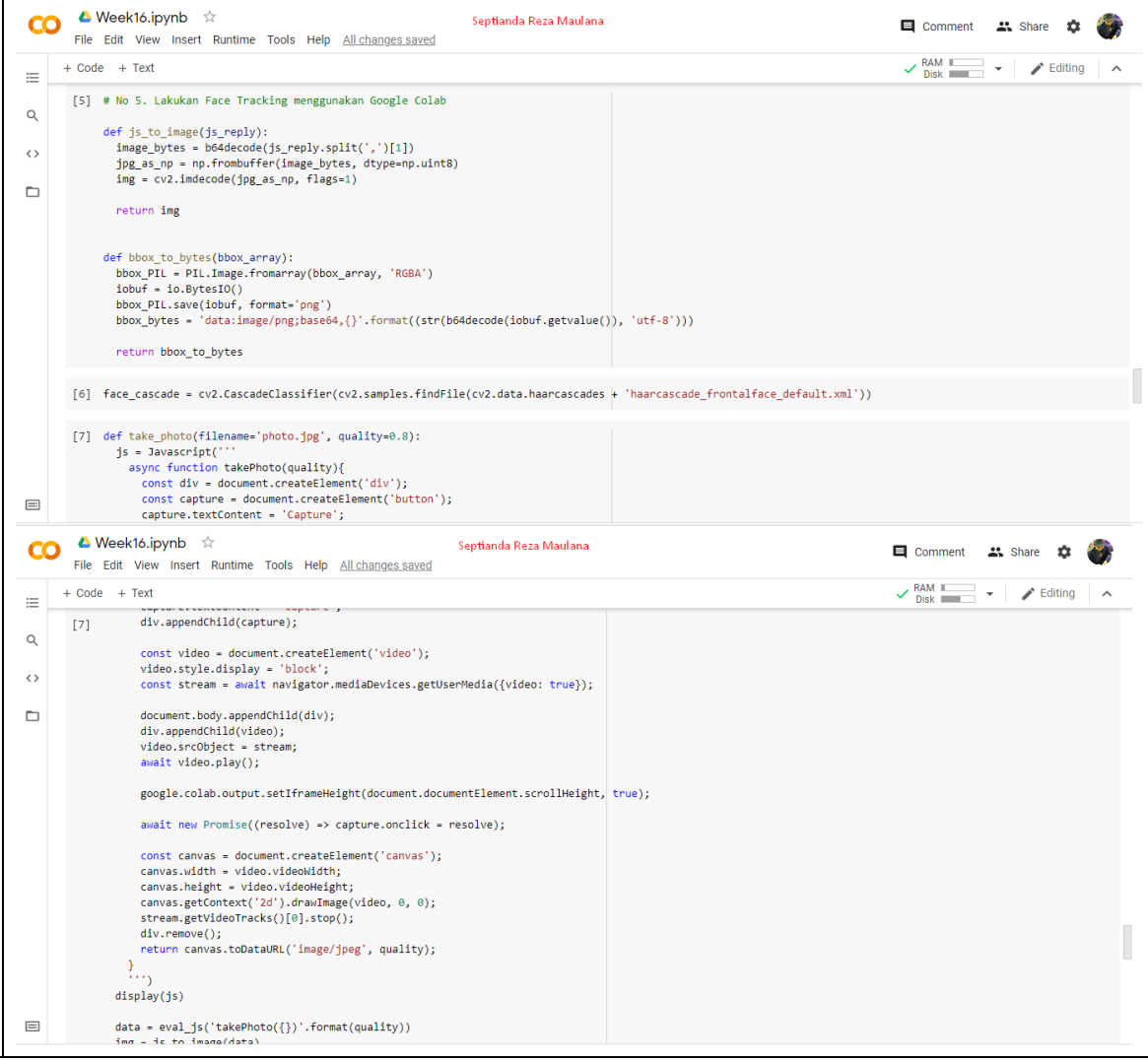


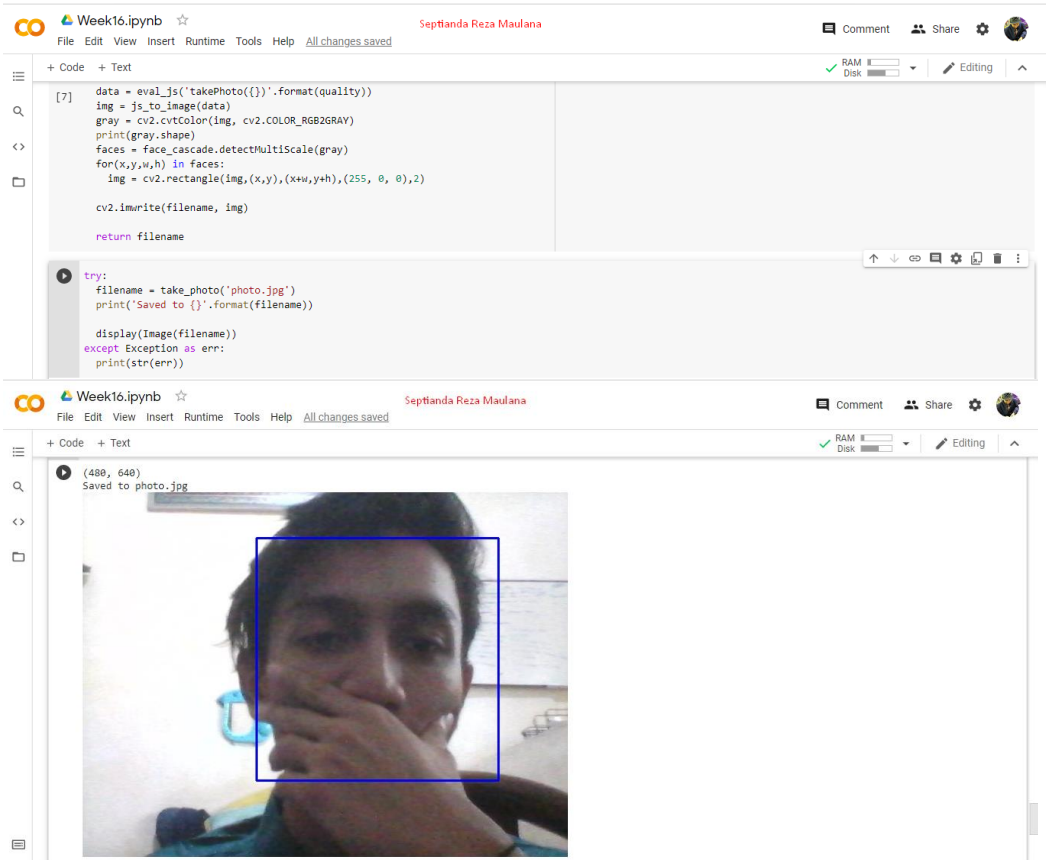
Perhatikan pada hasil face detection diatas. Secara keseluruhan, face detection dapat dilakukan dengan baik, bahkan untuk image berupa gambar bukan foto, wajah bermasker, atau wajah yang berukuran kecil (solvay).

3	Pada Soal No.2. wajah kucing tidak bisa dideteksi dengan baik. Lakukan deteksi wajah kucing hingga muncul rectangle pada bagian wajahnya. Petunjuk pada soal ini, perhatikan pretrained features yang telah disediakan OpenCV. Gunakan xml yang ada jika memang telah disediakan. Jika belum ada, coba cari dengan searching melalui search engines.
	
4	Cobakan juga untuk eyes detection.
	



5	Lakukan Face Tracking menggunakan Google Colab. Petunjuk, Tutorial selengkapnya tentang akses kamera dan FaceDetection pada google colab dapat dilihat di link berikut: https://www.youtube.com/watch?v=YjWh7QvVH60 .





Lakukan Blurring pada bagian wajah yang terdeteksi. Berikut contoh keluarannya. Petunjuk: anda dapat menggunakan cv.medianBlur untuk melakukan Blurring

6