

# BINUS University

<b>Academic Career:</b> <i>Undergraduate / <del>Master</del> / <del>Doctoral</del> *)</i>		<b>Class Program:</b> <i>International / Regular / <del>Smart Program</del> /  Global Class / <del>BINUS Online Learning</del> *)</i>	
<input type="checkbox"/> Mid Exam <input type="checkbox"/> Compact Term Exam <input checked="" type="checkbox"/> Final Exam <input type="checkbox"/> Others Exam : _____		<b>Term : Odd / <del>Even</del> / <del>Compact</del> *)</b> <b>Period (Only for BOL) : 1 / 2 *)</b>	
<input checked="" type="checkbox"/> Kemanggisan <input type="checkbox"/> Senayan <input type="checkbox"/> Semarang <input checked="" type="checkbox"/> Alam Sutera <input type="checkbox"/> Bandung <input checked="" type="checkbox"/> Bekasi <input type="checkbox"/> Malang		<b>Academic Year :</b>  <b>2022 / 2023</b>	
Exam Type* : <del>Onsite</del> / Online		Faculty / Dept. : School of Computer Science	
Day / Date** : Sabtu / 4 Februari 2023		Code - Course : COMP7116001 – Computer Vision COMP7116016 – Computer Vision	
Time** : 17.00		Code - Lecturer : Team Teaching	
Exam Specification*** : <input type="checkbox"/> Open Book <input type="checkbox"/> Open Notes <input type="checkbox"/> Close Book <input type="checkbox"/> Submit Project <input type="checkbox"/> Open E-Book <input type="checkbox"/> Oral Test		BULC (Only for BOL) : -	
Equipment*** : <input type="checkbox"/> Exam Booklet <input type="checkbox"/> Laptop <input type="checkbox"/> Drawing Paper – A3 <input type="checkbox"/> Calculator <input type="checkbox"/> Tablet <input type="checkbox"/> Drawing Paper – A2 <input type="checkbox"/> Dictionary <input type="checkbox"/> Smartphone <input type="checkbox"/> Notes		Class : All Classes	
Student ID *** : _____		Name *** : _____	
Signature *** : _____			
*) Strikethrough the unnecessary items      **) For Online Exam, this is the due date      ***) Only for Onsite Exam			
<p><b><i>Please insert the test paper into the exam booklet and submit both papers after the test.</i></b></p> <p><b><i>The penalty for CHEATING is DROP OUT!</i></b></p>			

## Learning Outcomes:

**LO 1 :** Describe various computational principles and standard image processing operators in computer vision

**LO 2 :** Explain the local features with their detectors and descriptors in computer vision **LO 3 :** Employ various features to find the correspondence between images and perform recognition in computer vision

**LO 4 :** Build various image recognition system in computer vision

**Nama:** Mario

**NIM:** 2440016905

**Kelas:** LA08 - LEC

Verified by,

Hidayaturrahman (D6423) and sent to Program on JAN 10, 2023

## I. Esai (100%)

1. Bayangkan Anda adalah seorang *freelancer* pembuat sistem berbasis *computer vision*. Pada suatu hari, seorang klien datang ke Anda untuk membuat sistem yang bisa mendeteksi apakah ada karakter berwajah merah atau tidak di gambar-gambar yang ada di folder pada tautan berikut:  
<https://drive.google.com/drive/folders/1mrf4cqjQB0OZYyu0EWINGahTihVPaMM?usp=sharing>

- a. [LO 1, LO 2 & LO 3, 45 poin] Jelaskan pendekatan dan algoritma-algoritma yang anda gunakan untuk memecahkan masalah yang diberikan oleh klien tersebut

Pertama-tama, kedua gambar yang diperoleh dari link google drive di atas akan disimpan kedalam sebuah variable. Mereka adalah “avengers-5-characters-shang-chi\_1.jpg” dan “delay-mcu.jpg”. Dengan melihat kedua gambar secara sekilas, dapat dilihat bahwa hanya ada 1 karakter berwajah merah, yaitu pahlawan bernama “Vision” yang beridiri di paling kiri dalam gambar kiri (“avengers-5-characters-shang-chi\_1.jpg”). Maka, algoritma-algoritma yang akan saya kembangkan harus dapat mengklasifikasikan dia saja sebagai karakter berwajah merah.



Terdapat 2 algoritma utama yang digunakan, yaitu algoritma untuk proses pendeteksi wajah dan algoritma untuk mengukur “tingkat kemerahan” dari sebuah gambar. Algoritma pendeteksi wajah dibuat dengan bantuan beberapa fungsi-fungsi penting dari library OpenCv (versi 2) sedangkan algortima pengukur “tingkat kemerahan” dari sebuah gambar dibuat sendiri tanpa bantuan library apapun.

Untuk menghemat waktu, kebanyakan algoritma deteksi wajah biasa merubah sebuah gambar menjadi grayscale terlebih dahulu. Lalu, kita akan memerlukan file “haarcascade\_frontalface\_default.xml” yang berisi data-data fitur wajah untuk membantu mendeteksi wajah dan akan digunakan oleh fungsi library OpenCV2. Beberapa parameter-parameter akan ditentukan: scaleFactor untuk mengganti ukuran gambar grayscalenya, minNeighbors untuk menentukan sebuah area (dimana sebuah muka terdeteksi) harus memiliki berapa tetangga (yang juga mendeteksi keberadaan sebuah muka) untuk dikatakan sebagai wajah, minSize untuk menentukan ukuran minimal dari sebuah wajah di gambar.

Ketika sebuah wajah ditemukan yang memenuhi kriteria-kriteria parameter fungsi deteksi wajah di atas, maka akan kita beri sebuah kotak untuk mengelilingi area yang komputer anggap adalah sebuah muka. Dari sini, hampir semua lokasi wajah sudah ditentukan dengan benar. Hanya saja pahlawan bernama “Falcon” (di sebelah paling kanan dari gambar delay-mcu.jpg) tidak bisa terdeteksi mukanya karena kostumnya menutupi sebagian dari wajahnya, terutama pelindung matanya. Mungkin akan diperlukan algoritma lebih kompleks lagi agar wajah-wajah yang terhalang seperti itu masih bisa terdeteksi dengan benar. Namun untuk keperluan ujian ini, saya percaya sistem pendeteksi wajah ini sudah cukup.

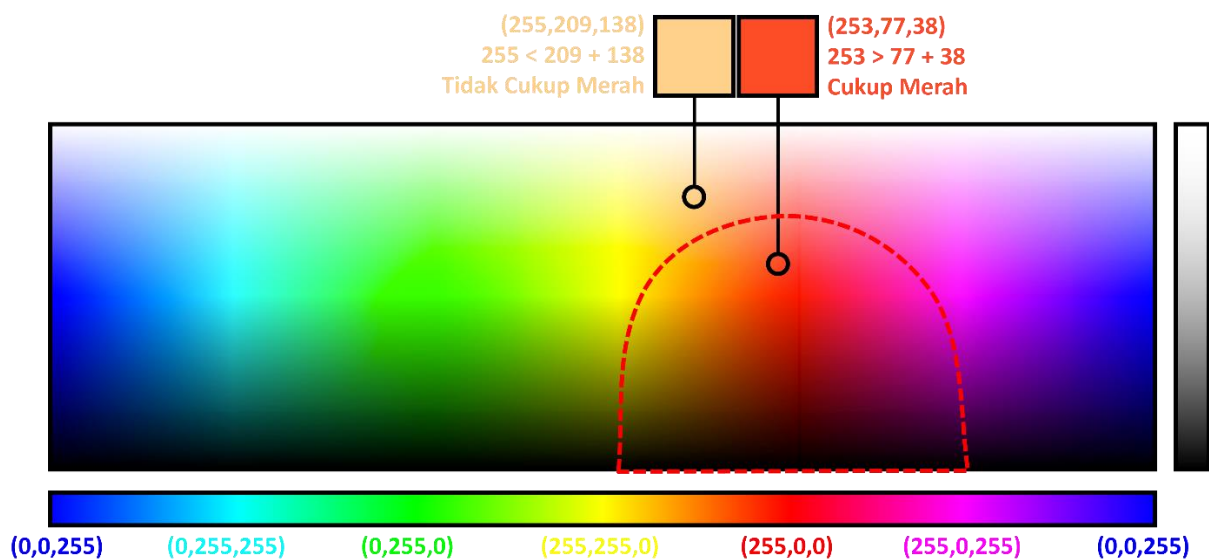
Verified by,

Hidayaturrahman (D6423) and sent to Program on JAN 10, 2023



Dari algoritma pendeteksi wajah sebelumnya, kita sekarang sudah tahu lokasi dari setiap wajah, dimana setiap wajah adalah sebuah sub-image (kotak) yang di ambil dari image lebih besar. Lalu, untuk setiap wajah, akan dijalankan sebuah algoritma untuk mengecek tingkat kemerahan dari muka tersebut.

Nilai RGB (0-255) untuk setiap piksel dari gambar wajah akan di cek. Jika nilai merah lebih besar daripada jumlah dari nilai hijau dan biru (merah > hijau + biru), maka piksel tersebut tergolong “cukup merah”. Diagram dibawah ini dibuat menggunakan Microsoft PowerPoint untuk membantu penjelasan, dimana semua warna yang di dalam bentuk setengah lingkaran merah akan digolong “cukup merah”. Ada 2 contoh warna piksel yang dipilih sebagai demonstrasi.



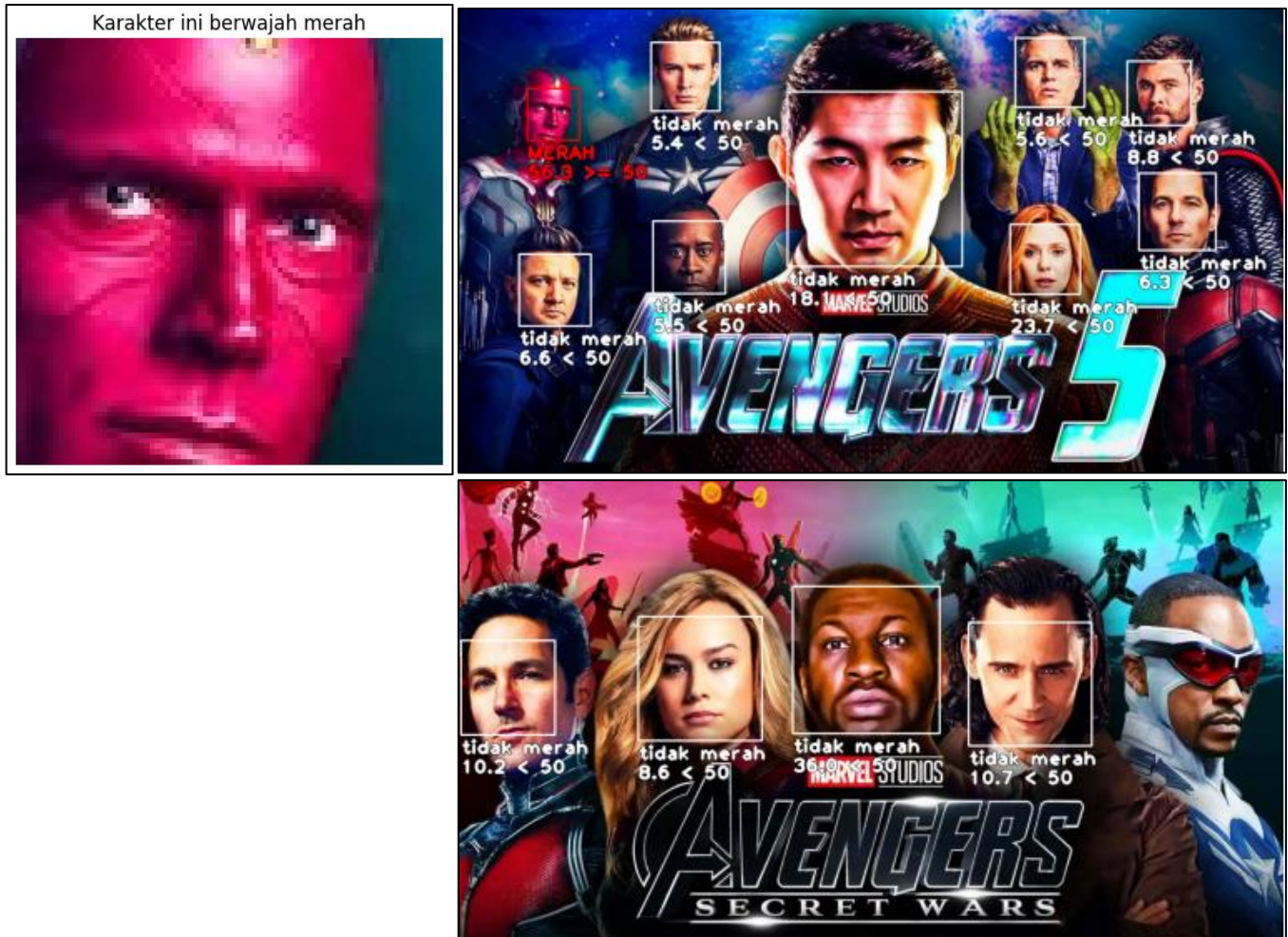
Untuk setiap piksel yang tergolong “cukup merah”, nilai “tingkat kemerahan” dari wajah tersebut bertambah sebanyak nilai merah - (nilai hijau + nilai biru). Setelah proses ini diulang untuk setiap piksel dari sebuah gambar wajah, nilai “tingkat kemerahan” dari wajah tersebut akan dihitung nilai rata-ratanya dengan cara membagi nilai tersebut dengan total jumlah piksel dari gambar wajah tersebut.

Pada akhirnya, jika sebuah gambar memiliki rata-rata “tingkat kemerahan” yang memenuhi sebuah nilai ambang minimum (untuk kasus ini, saya menyetel nilai ambang minimum sebagai 50), maka karakter di gambar itu dianggap memiliki wajah merah dan diberi box berwarna merah serta teks merah: “**MERAH {nilai rata-rata tingkat kemerahan} >= {nilai ambang minimum}**”. Jika tidak, maka karakter di gambar itu dianggap tidak memiliki wajah merah dan diberi box berwarna putih serta teks putih: “**tidak merah {nilai rata-rata tingkat kemerahan} < {nilai ambang minimum}**”. Hasil dari kedua gambar “avengers-5-characters-shang-chi\_1.jpg” dan “delay-mcu.jpg” setelah diberi kotak-kotak dan teks-teks akan ditampilkan. Selain itu, gambar dari wajah pahlawan Vision saja (yaitu satu-satunya karakter dengan wajah yang dianggap merah oleh algoritma ini) juga akan ditampilkan.

Verified by,

Hidayaturrahman (D6423) and sent to Program on JAN 10, 2023





Secara keseluruhan, program ini sudah berjalan sesuai dengan keinginan klien saya.

- b. **[LO 4, 25 poin]** Buatlah kode berdasarkan pendekatan yang anda ajukan di poin a menggunakan python notebook.

Code serta penjelasan code untuk bagain ini dapat dilihat pada file bernama "2440016905-Mario-CodeUASComputerVision.ipynb"

Semua gambar-gambar yang saya tampilkan di poin a adalah hasil screenshot dari python notebook (kecuali gambar penjelasan warna yang dibuat menggunakan PowerPoint).

Verified by,

*Hidayaturrahman (D6423) and sent to Program on JAN 10, 2023*

2. **[LO 1, LO 2 & LO 3, 30 poin]** Jelaskan dengan menggunakan kata-kata Anda sendiri mengenai proyek akhir (AOL) Anda serta kontribusi Anda dalam proyek tersebut.

Anggota kelompok saya terdiri dari:

2401962402 - Lawysen

2440016905 - Mario (Saya sendiri)

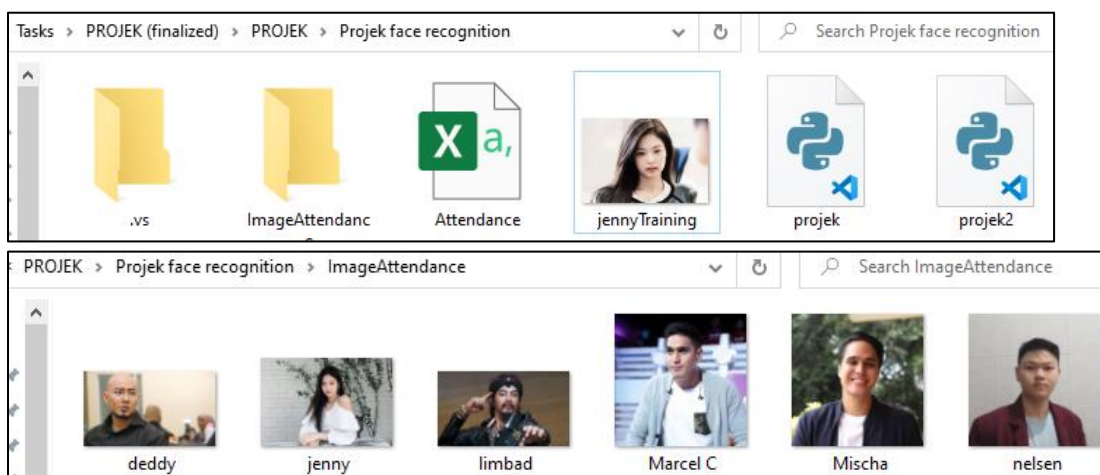
2401954205 - Nelsen

Beberapa file-file penting terkait proyek akhir (AOL) kelompok saya yang dimasukan dalam file zip ini:

- Folder yang mengandung proyek sebenarnya → “Projek face recognition”
  - File Python yang digunakan untuk memahami cara kerja library face\_recognition → “projek.py”
  - File Python untuk menjalankan program pencatatan dan deteksi muka → “projek2.py”
  - File Excel untuk menyimpan informasi absensi secara otomatis → “Attendance.xlsx”
  - 1 Gambar Latihan yang digunakan untuk proyek.py → “jennyTraining.jpg”
  - Folder yang mengandung semua gambar Latihan untuk proyek → “ImageAttendance”
- Presentasi kelompok → “Presentation Project Computer Vision.pdf”
- Laporan proyek akhir dengan format makalah IEEE → “Report Project Computer Vision IEEE.pdf”

Pada awalnya, kelompok kami bersama-sama bertukar pikiran memikirkan proyek apa yang ingin kita lakukan. Setelah banyak-banyak mencari dari sumber online, kami sepakat untuk membuat sebuah sistem deteksi dan absensi wajah otomatis menggunakan library Python face\_recognition. Sebagian besar dari penjelasan cara kerja dari proyek sudah dapat dilihat dari file Python (melalui teks yang dijadikan komen), file laporan, dan link presentasi. Maka, yang akan saya ketik untuk Ujian akhir ini akan lebih fokus dengan proses perkembangan proyek daripada penjelasan algoritma.

Untuk kedua file Python, kami menggunakan beberapa library-library yang sering kali digunakan: cv2, numpy, os, dan datetime. Namun, terdapat satu library baru yang sangat penting untuk mengaktifkan berbagai macam fungsi-fungsi pengenalan wajah yang bernama “face\_recognition”. Proses instalasinya lumayan rumit. Saya dan Lawysen berusaha mengikuti berbagai macam tutorial pengunduhan serta instalasi sendiri namun tidak berhasil membuatnya bekerja dengan benar di Python. Pada akhirnya, Nelsen berhasil menginstal library tersebut dengan benar. Setelah itu, kami mengadakan beberapa sesi kerja sama melalui Discord (saling berbicara dan menampilkan layar untuk melihat code bersama-sama). Kami juga mencari beberapa gambar-gambar yang akan digunakan dalam program ini.



Verified by,

Hidayaturrahman (D6423) and sent to Program on JAN 10, 2023

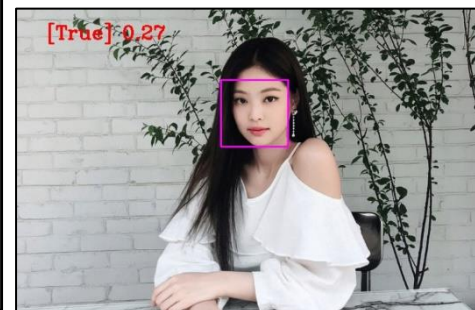
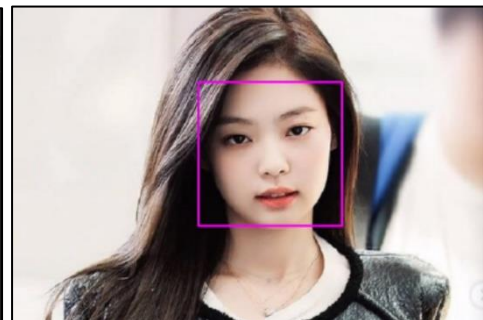
Pertama-tama, kami membuat file “projek.py” untuk mengetes dan memahami cara penggunaan library face\_recognition. Dengan proses coba-coba dan referensi dari dokumentasi library face\_recognition, kami perlahan-lahan berhasil membuat sebuah program yang melihat sebuah gambar latihan (referensi muka orang yang ingin kita acari) dan mendeteksi wajah yang sama pada gambar yang berbeda. Secara singkat, proses-proses yang dilakukan adalah:

- Membaca dan menyimpan 1 gambar latihan dan 1 gambar uji.
- Melalui algoritma kompleks, akan dicari dan disimpan informasi lokasi dari wajah-wajah yang ditemukan untuk setiap gambar.
- Menggambarkan kotak pada lokasi wajah yang terdeteksi.
- Membandingkan kemiripan dari kedua wajah dari kedua gambar dan menentukan (serta di ketik dalam gambar) apakah kedua muka tersebut memiliki identitas yang sama.
- Menampilkan hasil dari kedua gambar setelah diberi kotak dan teks.

```

projek.py
projek.py > ...
1 import cv2
2 import numpy as np
3 import face_recognition
4
5 # ambil gambar dari file yang uda di sediain
6 imgTrain = face_recognition.load_image_file('./jennyTraining.jpg')
7 # ubah warna gambar tersebut menjadi rgb
8 imgTrain = cv2.cvtColor(imgTrain, cv2.COLOR_BGR2RGB)
9 imgTest = face_recognition.load_image_file('./ImageAttendance/jenny.jpg')
10 imgTest = cv2.cvtColor(imgTest, cv2.COLOR_BGR2RGB)
11
12 # mencari lokasi wajah
13 faceLoc = face_recognition.face_locations(imgTrain)[0]
14 encodeTrain = face_recognition.face_encodings(imgTrain)[0]
15 # memberikan kotak kepada lokasi wajah yang telah di temukan
16 cv2.rectangle(imgTrain, (faceLoc[3], faceLoc[0]), (faceLoc[1], faceLoc[2]), (255, 0, 255), 2)
17
18 faceLocTest = face_recognition.face_locations(imgTest)[0]
19 encodeTest = face_recognition.face_encodings(imgTest)[0]
20 cv2.rectangle(imgTest, (faceLocTest[3], faceLocTest[0]), (faceLocTest[1], faceLocTest[2]), (255, 0, 255), 2)
21
22 #Ini untuk mengecek apakah wajahnya itu sama atau tidak
23 results = face_recognition.compare_faces([encodeTrain], encodeTest)
24 #Semakin pendek jaraknya maka semakin mirip mukanya
25 faceDis = face_recognition.face_distance([encodeTrain], encodeTest)
26
27 # memberikan tulisan pada gambar apakah sama(true) atau tidak(false)
28 # dan jarak kemiripan antara gambar test dengan Train
29 cv2.putText(imgTest, f'{results} {round(faceDis[0], 2)}', (50,50), cv2.FONT_HERSHEY_COMPLEX, 1, (0,0,255), 2)
30
31 # menampilkan gambar Train dan test
32 cv2.imshow('Train', imgTrain)
33 cv2.imshow('Test', imgTest)
34 cv2.waitKey(0)

```



Setelah itu, kami membuat file “projek2.py” untuk membuat sistem deteksi wajah dan pencatatan absensi otomatis. Kami mencari berbagai macam referensi online dan melihat kembali pelajaran-pelajaran dari asisten lab untuk membuat program ini. Secara singkat, proses-proses yang dilakukan adalah:

- Membaca dan menyimpan semua gambar dari folder “ImageAttendance”.
- Membuat sebuah fungsi untuk mencari dan menyimpan informasi lokasi dari wajah-wajah yang ditemukan untuk setiap gambar.
- Membuat sebuah fungsi untuk mencatat nama dari wajah yang terdeteksi kedalam file “Attendance.csv”.
- Sampai program dihentikan, tampilan dari webcam akan terus diambil dan ditampilkan tanpa henti.
- Membandingkan kemiripan muka yang terdeteksi dari webcam dengan muka-muka dari gambar-gambar sebelumnya. Jika ditemukan sebuah muka yang cukup mirip dengan muka dari webcam, maka akan digambarkan kotak dan nama pada lokasi wajah serta dicatat nama dan waktu deteksi pada file excel.

Verified by,

Hidayaturrahman (D6423) and sent to Program on JAN 10, 2023

```

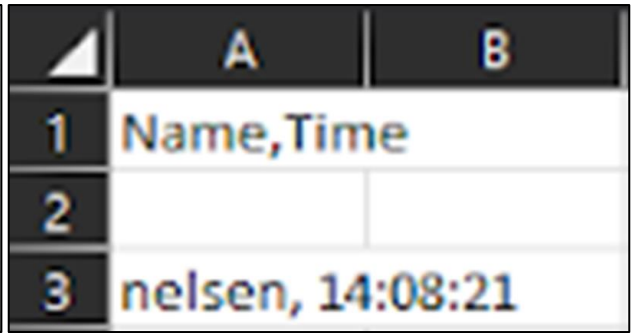
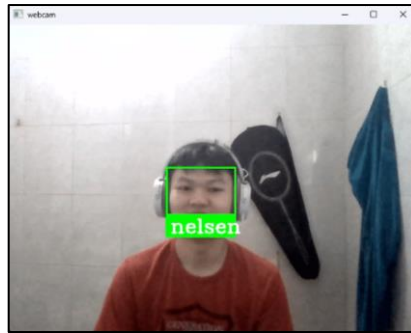
projek2.py X
projek2.py > ...
1 import cv2
2 import numpy as np
3 import face_recognition
4 import os
5 from datetime import datetime
6
7 #buat nampung gambar
8 path = 'ImageAttendance'
9 images = []
10 classNames = []
11 myList = os.listdir(path)
12
13 #Masukkan semua gambar dan class ke dalam list
14 for cls in myList:
15     curImg = cv2.imread(path+'/'+cls)
16     images.append(curImg)
17     classNames.append(os.path.splitext(cls)[0])
18
19 #Encoding Function yang menghitung semua encodings dari gambar yang ada
20 def findEncodings(images):
21     encodeList = []
22     for img in images:
23         img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
24         encode = face_recognition.face_encodings(img)[0]
25         encodeList.append(encode)
26     return encodeList
27
28 # membuat sebuah fungsi yang berfungsi untuk mengambil nama dan waktu untuk di tulis kedalam database
29 # dalam kasus ini kami menggunakan file csv atau excel
30 def markAttendance(name):
31     with open('Attendance.csv', 'r+') as f:
32         myDataList = f.readlines()
33         nameList = []
34         for line in myDataList:
35             #Untuk ngambil namanya saja
36             entry = line.split(',')
37             nameList.append(entry[0])
38
39             #Untuk ngambil waktu saat attendance
40             if name not in nameList:
41                 now = datetime.now()
42                 dtString = now.strftime('%H:%M:%S')
43                 f.writelines(f'\n{name},{dtString}')
44
45 # menyimpan data gambar (nama) kedalam variable encodeListKnown
46 encodeListKnown = findEncodings(images)

```

```

47
48 #Untuk mengambil image dari webcam
49 cap = cv2.VideoCapture(0)
50
51 # terus melakukan pengecekan selama webcam menyala
52 while True:
53     success, img = cap.read()
54     #Mereduce size dari gambar agar proses dapat berjalan lebih cepat
55     imgS = cv2.resize(img, (0,0), None, 0.25, 0.25)
56     imgS = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
57
58     #Mencari lokasi muka di dalam gambar pada webcam
59     faceCurFrame = face_recognition.face_locations(imgS)
60     #Mengencode gambar pada webcam
61     encodesCurFrame = face_recognition.face_encodings(imgS, faceCurFrame)
62
63     # melakukan pengecekan antara gambar dari webcam dengan gambar yang telah kita simpan sebelumnya
64     for encodeFace, faceLoc in zip(encodesCurFrame, faceCurFrame):
65         matches = face_recognition.compare_faces(encodeListKnown, encodeFace)
66         faceDis = face_recognition.face_distance(encodeListKnown, encodeFace)
67
68         #Mencari nilai index yang terkecil
69         matchIndex = np.argmin(faceDis)
70
71     #Menampilkan nama dari index yang terkecil
72     if matches[matchIndex]:
73         # name = classNames[matchIndex].upper()
74         name = classNames[matchIndex]
75         y1,x2,y2,x1 = faceLoc
76         #Untuk menyesuaikan dengan gambar yang telah di resize
77         # y1,x2,y2,x1 = y1*4,x2*4,y2*4,x1*4
78         #Menandai wajah dengan kotak
79         cv2.rectangle(img, (x1,y1), (x2,y2), (0,255,0), 2)
80         cv2.rectangle(img, (x1,y2-35), (x2,y2), (0,255,0), cv2.FILLED)
81         #Menampilkan nama dari wajah yang terdeteksi
82         cv2.putText(img, name, (x1+6, y2-6), cv2.FONT_HERSHEY_COMPLEX, 1, (255,255,255), 2)
83         #Masukkan data attendance ke csv
84         markAttendance(name)
85
86 #Untuk menampilkan webcam
87 cv2.imshow('webcam', img)
88 cv2.waitKey(1)

```



Tentunya, sebelum kami mulai membuat presentasi atau laporan kami, setiap anggota mengecek kembali kelancaran / kebenaran dari program pythonnya. Setiap kali ada kesalahan error dalam programnya ketika dijalankan, kami mencoba untuk melakukan proses debugging secara bersama-sama hingga semua anggota dapat menjalani programnya tanpa masalah.

Setelah semua file dari folder “Projek face recognition” sudah selesai dibuat, kami lanjut untuk membuat presentasi dan laporan yang harus dikumpul untuk keperluan dokumentasi serta bukti pemahaman kami terkait projek yang kami buat. Saya cukup pandai dalam segi desain visual. Maka, saya yang membantu mendesain mayoritas dari tampilan, warna, tata letak, dan pemformatan untuk teks serta gambar dalam presentasi ini. Kelompok saya bersama-sama membuat isi dari halaman-halaman presentasi menggunakan website bernama Canva. Kami menentukan teks dan gambar apa saja yang harus dimasukkan ke dalam slide dan apa saja yang harus kita hafal. Setelah itu, kami membagi seksi-seksi yang akan dibahas saat presentasi nanti. Bagian saya adalah slide 12-18. Dosen kurang lebih sudah puas dengan proyeknya, hanya saja beberapa bagian teori perlu dijelaskan dengan lebih baik lagi. Misalnya, apa yang dilakukan oleh fungsi faceLog dan encodeTrain. Setelah perbaikan-perbaikannya dilakukan, kami menyimpan presentasi tersebut menjadi sebuah file pdf.

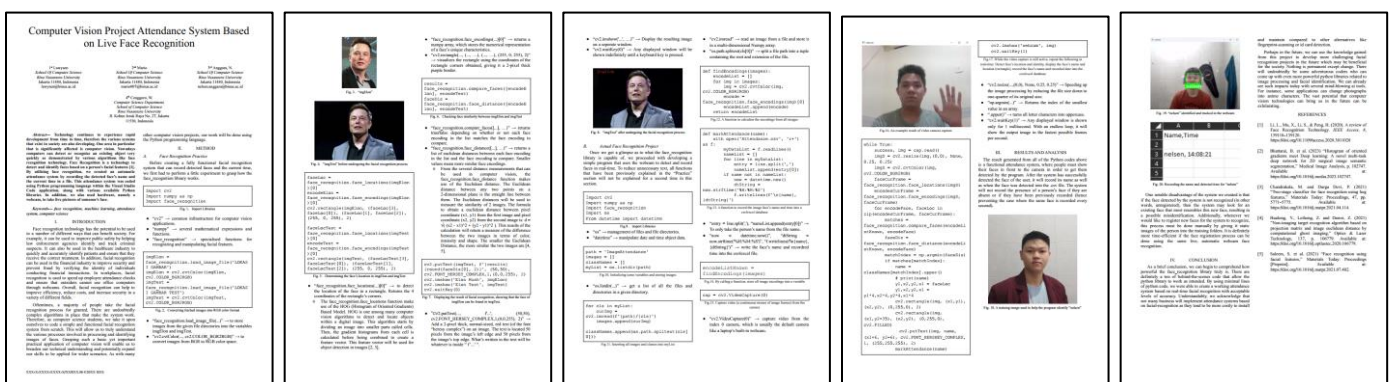
Verified by,

Hidayaturrahman (D6423) and sent to Program on JAN 10, 2023





Setelah presentasi selesai dan saran perbaikan dari dosen diimplementasikan, kami lanjut membuat laporan nya menggunakan Google Docs. Saya mengisi laporan tersebut dengan setiap gambar, code, dan penjelasan singkat dari setiap fungsi atau baris code lainnya. Menggunakan arahan dari Lawysen dan Nelsen, saya mengetik isi dari laporannya, mulai dari bagian abstrak (di awal laporan) hingga kesimpulan. Karena kemahiran saya dengan bahasa Inggris, saya dipercaya untuk menerjemahkan dan mengoreksi semua teks kami baik untuk laporan. Saya juga yang memastikan bahwa tata letak teks, gambar, dan bagian-bagian dari code rapih sesuai format IEEE. Sesuai dengan permintaan dosen, kami juga mengambil referensi dari beberapa jurnal dan makalah lainnya untuk menjelaskan beberapa hal dengan lebih detail, seperti fungsi “face\_recognition.face\_locations” dan “face\_recognition.face\_distance”. Setelah laporannya selesai dibuat, kami menyimpannya menjadi sebuah file pdf.



Setelah kami yakin dengan kebenaran dari program, laporan, dan presentasi proyek (meninjau semuanya untuk terakhir kalinya), kami mengumpulkan semua file-file penting untuk proyek kepada dosen.

-- Selamat Mengerjakan --

Verified by,

Hidayaturrahman (D6423) and sent to Program on JAN 10, 2023