

# **PENGANTAR TUGAS**

## **COMPUTER VISION**



Dosen Pengampu : Bana Handaga, Ir. M.T., Ph.D

Disusun oleh:

Nama : Huan Wendy Ariono

Nim : L200180086

Program Studi : Informatika

**PRODI TEKNIK INFORMATIKA**  
**FAKULTAS KOMUNIKASI DAN INFORMATIKA**  
**UNIVERSITAS MUHAMMADIYAH SURAKARTA**

**2020/2021**

## **KATA PENGANTAR**

Puji dan syukur penyusun panjatkan kehadirat Allah SWT karena dengan Rahmat dan Karunia-Nya, penyusun dapat menyelesaikan pengantar tugas Mata Kuliah Computer Vision. Pembuatan pengantar tugas ini bertujuan untuk memenuhi tugas Mata Kuliah Computer Vision. Penyusun juga berharap semoga pengantar tugas ini dapat memenuhi standar penilaian bapak dosen.

Penyusun menyadari bahwa pembuatan pengantar tugas ini tidak hanya mengandalkan gagasan pribadi melainkan banyak mendapat referensi dari berbagai pihak baik secara langsung maupun tidak langsung. Penyusun menyadari bahwa pengantar tugas ini jauh dari kata sempurna. Atas perhatiannya saya ucapkan terimakasih

## **DAFTAR ISI**

HALAMAN SAMPUL.....	0
KATA PENGANTAR.....	1
DAFTAR ISI.....	2
BAB 1 PENDAHULUAN .....	3
BAB 2 PEMBAHASAN.....	4
BAB 3 PENUTUP.....	9

## **BAB I**

### **PENDAHULUAN**

LBPH adalah salah satu algoritma pengenalan objek dengan membagi gambar menjadi beberapa bagian kemudian menerjemahkannya menjadi bilangan biner representatifnya dalam bentuk histogram (saya terjemahkan bebas dari sumber aslinya di sini). Dalam LBPH setidaknya ada 4 parameter yang harus kita pahami, yakni 'Radius' (radius pusat pixel acuan), 'Neighbors' (tetangga pixel acuan), 'Grid X', dan 'Grid Y' (pembagian gambar). Untuk grid X dan Y, semakin banyak hasilnya akan semakin baik/presisi. Beruntunglah kita ada sekelompok komunitas opensource Computer Vision yang mau membagi source-nya ke publik, yakni OpenCV. Dalam openCV sudah ada class yang sudah mengenkapsulasi semua yang kita butuhkan dalam algoritma LBPH ini, yakni class `cv::face::LBPHFaceRecognizer`. Dengan class ini kita dapat dengan mudah mengklasifikasikan objek yang terdeteksi sesuai dengan dataset yang kita buat sebelumnya, sehingga tujuan akhir pengenalan objek dapat tercapai

## BAB 2

### PEMBAHASAN

#### Link Gdrive Tugas

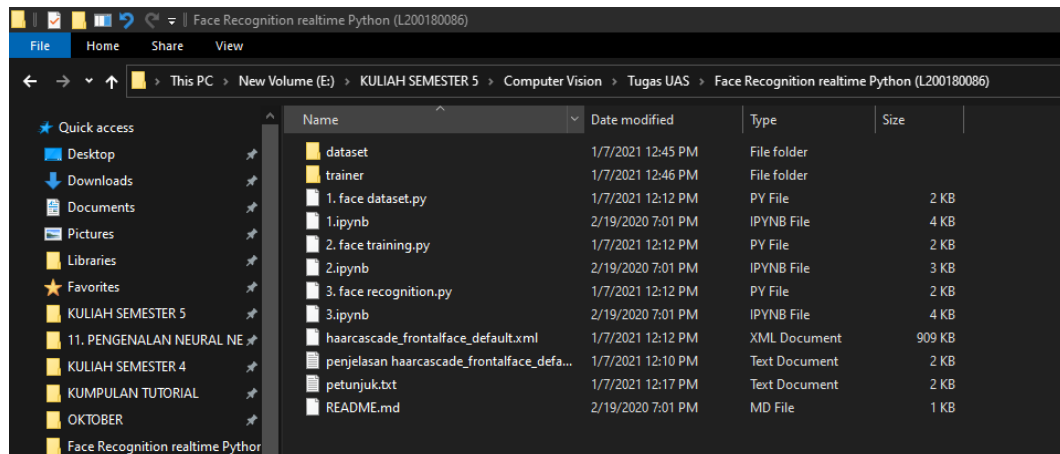
[https://drive.google.com/file/d/10uQi\\_4lqBoBR1oymPbFl-yZ3Hz7OmJK/view?usp=sharing](https://drive.google.com/file/d/10uQi_4lqBoBR1oymPbFl-yZ3Hz7OmJK/view?usp=sharing)

#### TAHAPAN PENGENALAN WAJAH (FACE RECOGNITION) OPENCV – PYTHON

Untuk dapat mengenali wajah sesuai identitasnya menggunakan OpenCV dan Python, setidaknya Anda harus melakukan 3 tahap sebagai berikut:

1. Membuat dataset referensi wajah
2. Mentraining data
3. Melakukan pengenalan wajah (Face Recognition)

Tiga tahap di atas akan kita lakukan menggunakan script yang berbeda. Dibawah ini adalah script yang dibutuhkan. Jangan lupa kita juga harus membuat folder dataset dan trainer



#### 1. face dataset.py

Ambil gambar menggunakan skrip face dataset.py. Skrip akan menyimpan 30 gambar wajah Anda di folder dataset setelah Anda memasukkan nomor ID (HARUS berupa bilangan bulat dan inkremental (dimulai dengan 1 lalu 2, 3, ...))

disini saya membuat maksimal 2 wajah

Catatan: Pastikan wajah Anda berada di tengah. Jendela akan runtuh saat semua 30 gambar diambil.

run => face dataset.py

### **coding face dataset.py**

```
import cv2
import os

cam = cv2.VideoCapture(0)
cam.set(3, 640) # set video width
cam.set(4, 480) # set video height

#pastikan 'haarcascade_frontalface_default.xml' ada di folder yang
sama dengan kode ini
face_detector =
cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

# Untuk setiap orang, masukkan satu id wajah numerik (harus
memasukkan angka mulai dari 1, ini adalah label orang 1)
face_id = input('\n enter user id end press <return> ==>  ')

print("\n [INFO] Initializing face capture. Look the camera and
wait ...")
# Inisialisasi hitungan wajah pengambilan sampel individu
count = 0

#mulai deteksi wajah Anda dan ambil 30 gambar
while(True):

    ret, img = cam.read()
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = face_detector.detectMultiScale(gray, 1.3, 5)

    for (x,y,w,h) in faces:

        cv2.rectangle(img, (x,y), (x+w,y+h), (255,0,0), 2)
        count += 1

        # Simpan gambar yang diambil ke dalam folder dataset
        cv2.imwrite("dataset/User." + str(face_id) + '.' +
str(count) + ".jpg", gray[y:y+h,x:x+w])

        cv2.imshow('image', img)

    k = cv2.waitKey(100) & 0xff # Press 'ESC' for exiting video
    if k == 27:
        break
    elif count >= 30: # Take 30 face sample and stop video
        break

# Do a bit of cleanup
print("\n [INFO] Exiting Program and cleanup stuff")
cam.release()
cv2.destroyAllWindows()
```

## 2. face training.py

Skrip face training.py akan melatih model untuk mengenali semua wajah dari 30 gambar yang diambil menggunakan skrip face dataset.py, dan menyimpan keluaran pelatihan di file trainer.yml di dalam folder trainer

run => python training.py

### **coding python training.py**

```
import cv2
import numpy as np
from PIL import Image #pillow package
import os

# Path untuk database gambar wajah
path = 'dataset'

recognizer = cv2.face.LBPHFaceRecognizer_create()
detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml");

# berfungsi untuk mendapatkan gambar dan data label
def getImagesAndLabels(path):

    imagePaths = [os.path.join(path,f) for f in os.listdir(path)]
    faceSamples=[]
    ids = []

    for imagePath in imagePaths:

        PIL_img = Image.open(imagePath).convert('L') # convert it
to grayscale
        img_numpy = np.array(PIL_img,'uint8')

        id = int(os.path.split(imagePath)[-1].split(".")[1])
        faces = detector.detectMultiScale(img_numpy)

        for (x,y,w,h) in faces:
            faceSamples.append(img_numpy[y:y+h,x:x+w])
            ids.append(id)

    return faceSamples,ids

print ("\n [INFO] Training faces. It will take a few seconds. Wait
...")
faces,ids = getImagesAndLabels(path)
recognizer.train(faces, np.array(ids))

# simpan model kedalam trainer/trainer.yml
recognizer.write('trainer/trainer.yml') # simpan untuk mac bukan
Pi

# Cetak jumlah wajah yang dilatih dan akhiri program
print("\n [INFO] {0} faces trained. Exiting
Program".format(len(np.unique(ids))))
```

### 3. face recognition.py

face recognition.py adalah skrip utama. Anda perlu menambahkan nama setiap orang yang melihat fotonya diambil di skrip dataset.py. Program akan mengenali wajah sesuai dengan id yang diberikan dalam skrip dataset.py. Jika huan memiliki id 1, namanya akan muncul dalam daftar sebagai indeks 1 seperti ini names = ['None', 'huan'] # keep None dan menambahkan nama ke dalam daftar ini

run => face recognition.py

#### **coding python recognition.py**

```
import cv2
import numpy as np
import os

recognizer = cv2.face.LBPHFaceRecognizer_create()
recognizer.read('trainer/trainer.yml') #memuat model terlatih
cascadePath = "haarcascade_frontalface_default.xml"
faceCascade = cv2.CascadeClassifier(cascadePath);

font = cv2.FONT_HERSHEY_SIMPLEX

#membuat id counter, jumlah orang yang ingin Anda sertakan
id = 2 #dua orang

names = ['', 'Huan', 'Wendy'] #key in names, mulai dari tempat
kedua, kosongkan pertama

# Inisialisasi dan mulai pengambilan video waktu nyata
cam = cv2.VideoCapture(0)
cam.set(3, 640) # set video width
cam.set(4, 480) # set video height

# Tentukan ukuran jendela min untuk dikenali sebagai wajah
minW = 0.1*cam.get(3)
minH = 0.1*cam.get(4)

while True:

    ret, img =cam.read()

    gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)

    faces = faceCascade.detectMultiScale(
        gray,
        scaleFactor = 1.2,
        minNeighbors = 5,
        minSize = (int(minW), int(minH)),
    )

    for(x,y,w,h) in faces:

        cv2.rectangle(img, (x,y), (x+w,y+h), (0,255,0), 2)
```



```

        id, confidence = recognizer.predict(gray[y:y+h,x:x+w])

        # Check if confidence is less than 100 ==> "0" is perfect
match
        if (confidence < 100):
            id = names[id]
            confidence = " {0}%".format(round(100 - confidence))
        else:
            id = "unknown"
            confidence = " {0}%".format(round(100 - confidence))

        cv2.putText(img, str(id), (x+5,y-5), font, 1,
(255,255,255), 2)
        cv2.putText(img, str(confidence), (x+5,y+h-5), font, 1,
(255,255,0), 1)

    cv2.imshow('camera',img)

    k = cv2.waitKey(10) & 0xff # Press 'ESC' for exiting video
    if k == 27:
        break

# Do a bit of cleanup
print("\n [INFO] Exiting Program and cleanup stuff")
cam.release()
cv2.destroyAllWindows()

```

### **penjelasan file haarcascade\_frontalface\_default.xml**

OpenCV ialah open source library computer vision, yang memudahkan pemrograman deteksi wajah, face tracking, face recognition, kalman filtering dan berbagai metode artificial intelligent.

OpenCV menggunakan sebuah tipe face detector yang disebut Haar Cascade classifier. Gambar menunjukkan face detector berhasil bekerja pada sebuah gambar. Jika ada sebuah image (bias dari file /live video), face detector menguji tiap lokasi image dan mengklasifikasinya sebagai “wajah” atau “bukan wajah”. Klasifikasi dimisalkan sebuah skala fix untuk wajah, misal 50x50 pixel. Jika wajah pada image lebih besar atau lebih kecil dari pixel tersebut, classifier terus menerus jalan beberapa kali, untuk mencari wajah pada gambar tersebut.

Classifier menggunakan data yang disimpan pada file XML untuk memutuskan bagaimana mengklasifikasi tiap lokasi image. OpenCV menggunakan 4 data XML untuk deteksi wajah depan, dan 1 untuk wajah profile. Termasuk juga 3 file XML bukan wajah: 1 untuk deteksi full body, 1 untuk upper body, dan 1 untuk lower body. Anda harus memberitahukan classifier dimana menemukan file data yang akan anda gunakan. Salah satunya bernama haarcascade\_frontalface\_default.xml. Pada OpenCV.

### **BAB III**

#### **PENUTUP**

Setelah kita menjalankan semua program diatas dengan baik pastinya aplikasi deteksi wajah dapat berfungsi dengan baik. Jadi dapat disimpulkan Mengenali wajah atau gambar sangatlah mudah untuk dilakukan oleh manusia namun kadang kala manusia itu lupa mengenali wajah atau gambar karena beberapa faktor contohnya faktor usia. Namun untuk membantu tugas manusia dalam mengidentifikasi atau mengenali wajah seseorang dibutuhkan sistem yang dapat bekerja dalam mengenali wajah atau gambar. Sistem ini harus diberi perlakuan khusus, ketika kita memasukan inputan sistem ini dapat mengenali wajah atau gambar tersebut.

Sistem yang sekarang sedang dikembangkan adalah Face Recognition. Face recognition merupakan salah satu teknik biometric yang memungkinkan komputer atau mesin dapat mengenali wajah atau gambar. Melalui sistem pengenalan wajah ini, identitas seseorang dapat diketahui dengan mudah dengan memanfaatkan kamera yang diarahkan ke wajah membandingkan inputan citra wajah yang masuk dengan citra yang ada di database.