UJIAN TENGAH SEMESTER

KOMPUTER GRAFIK 1

“FACE RECOGNITION SEDERHANA MENGGUNAKAN LIBRARY OPEN CV BAHASA PHYTON”

Kelompok 4 :

NAMA : NIM :

1. Deny Bagus Septian 201011400381

2. Faisal Irsyad Noer 201011400873

3. Haerudin 201011400070

4. Firman Noer Rizal 201011401399

**penggunaan library OpenCV dalam bahasa Python untuk melakukan deteksi dan pengenalan wajah.**

**Berikut source code nya :**

import cv2

# Menginisialisasi detektor wajah

face\_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade\_frontalface\_default.xml')

# Menginisialisasi recognizer wajah

recognizer = cv2.face.LBPHFaceRecognizer\_create()

# Mengimpor data pelatihan wajah

recognizer.read('train\_data.yml')

# Inisialisasi kamera

cap = cv2.VideoCapture(0)

while True:

# Membaca frame dari kamera

ret, frame = cap.read()

# Mengkonversi frame menjadi skala abu-abu

gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

# Mendeteksi wajah dalam frame

faces = face\_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5, minSize=(30, 30))

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

# Menggambar kotak di sekitar wajah yang terdeteksi

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

# Mengenali wajah

roi\_gray = gray[y:y+h, x:x+w]

id\_, confidence = recognizer.predict(roi\_gray)

# Menampilkan hasil pengenalan

if confidence <= 100:

# Nama pengguna berdasarkan ID

# Ganti dengan database atau logika pengenalan wajah yang sesuai

name = "Pengguna " + str(id\_)

else:

name = "Tidak Dikenali"

cv2.putText(frame, name, (x, y-10), cv2.FONT\_HERSHEY\_SIMPLEX, 0.9, (0, 255, 0), 2)

# Menampilkan frame hasil

cv2.imshow('Face Recognition', frame)

# Tombol 'q' untuk keluar dari program

if cv2.waitKey(1) & 0xFF == ord('q'):

break

# Melepaskan sumber daya

cap.release()

cv2.destroyAllWindows()

note : Pastikan Anda sudah menginstal library OpenCV untuk menjalankan kode ini. Kode tersebut menggunakan kamera untuk mendeteksi wajah dalam frame video secara real-time dan melakukan pengenalan menggunakan metode pengenalan wajah LBPH (Local Binary Patterns Histograms). Namun, perlu diingat bahwa ini hanya contoh sederhana, dan dalam pengembangan aplikasi pengenalan wajah yang lebih kompleks, dibutuhkan lebih banyak langkah seperti pelatihan model dan manajemen database wajah yang lebih canggih.

**Artikel & Tutorial langkah – langkah nya :**

Pengenalan wajah merupakan salah satu aplikasi yang menarik dalam bidang komputer vision. Dengan menggunakan library OpenCV, sebuah library populer dalam pengolahan citra dan penglihatan mesin, kita dapat mengimplementasikan sistem pengenalan wajah yang sederhana namun efektif. Dalam artikel ini, kita akan menjelaskan langkah-langkah dasar untuk membuat sistem pengenalan wajah dengan menggunakan library OpenCV dalam bahasa Python.

Langkah 1: Persiapan Awal

Pertama, pastikan Anda telah menginstal library OpenCV pada sistem Python Anda. Anda dapat menginstalnya dengan menggunakan pip melalui perintah berikut:

pip install opencv-python

Selain itu, pastikan Anda memiliki dataset gambar wajah untuk melatih sistem pengenalan. Dataset ini harus berisi gambar-gambar wajah yang sudah diberi label sesuai dengan identitas individu.

Langkah 2: Deteksi Wajah

Langkah pertama dalam sistem pengenalan wajah adalah mendeteksi wajah dalam gambar atau video. OpenCV menyediakan detektor wajah siap pakai yang dapat kita gunakan. Kita akan menggunakan Haar Cascade Classifier, yang sudah dilatih untuk mendeteksi fitur wajah.

Source code :

import cv2

# Menginisialisasi detektor wajah

face\_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade\_frontalface\_default.xml')

# Membaca gambar atau video

img = cv2.imread('image.jpg')

gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

# Mendeteksi wajah

faces = face\_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5, minSize=(30, 30))

# Menandai wajah yang terdeteksi

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

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

# Menampilkan gambar dengan wajah yang terdeteksi

cv2.imshow('Deteksi Wajah', img)

cv2.waitKey(0)

cv2.destroyAllWindows()

Langkah 3: Pelatihan Sistem

Setelah mendeteksi wajah, langkah berikutnya adalah melatih sistem pengenalan. Dalam contoh ini, kita akan menggunakan metode pengenalan wajah LBPH (Local Binary Patterns Histograms) yang sederhana namun efektif.

Source code :

import cv2

import os

# Menginisialisasi detektor wajah

face\_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade\_frontalface\_default.xml')

# Menginisialisasi recognizer wajah

recognizer = cv2.face.LBPHFaceRecognizer\_create()

# Membaca dataset pelatihan

def read\_dataset(dataset\_path):

image\_paths = [os.path.join(dataset\_path, f) for f in os.listdir(dataset\_path)]

face\_samples = []

ids = []

for image\_path in image\_paths:

image = cv2.imread(image\_path)

gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

faces = face\_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=

5, minSize=(30, 30))

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

face\_samples.append(gray[y:y+h, x:x+w])

ids.append(1) # Ganti dengan ID sesuai dengan identitas individu

return face\_samples, ids

# Melatih sistem pengenalan

dataset\_path = 'dataset'

faces, ids = read\_dataset(dataset\_path)

recognizer.train(faces, np.array(ids))

# Menyimpan model pelatihan

recognizer.save('train\_data.yml')

Pastikan Anda memiliki dataset yang berisi gambar-gambar wajah yang sudah diberi label dengan benar sesuai dengan identitas individu. Kemudian, modifikasi kode untuk mengatur ID yang sesuai dengan setiap identitas dalam dataset.

Langkah 4: Pengenalan Wajah

Setelah melatih sistem pengenalan, kita dapat menggunakan model yang sudah dilatih untuk mengenali wajah dalam gambar atau video.

Source code :

import cv2

# Menginisialisasi detektor wajah

face\_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade\_frontalface\_default.xml')

# Menginisialisasi recognizer wajah

recognizer = cv2.face.LBPHFaceRecognizer\_create()

# Mengimpor data pelatihan wajah

recognizer.read('train\_data.yml')

# Membaca gambar atau video

img = cv2.imread('image.jpg')

gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

# Mendeteksi wajah

faces = face\_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5, minSize=(30, 30))

# Mengenali wajah

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

roi\_gray = gray[y:y+h, x:x+w]

id\_, confidence = recognizer.predict(roi\_gray)

if confidence <= 100:

name = "Pengguna " + str(id\_)

else:

name = "Tidak Dikenali"

cv2.putText(img, name, (x, y-10), cv2.FONT\_HERSHEY\_SIMPLEX, 0.9, (0, 255, 0), 2)

# Menampilkan gambar dengan hasil pengenalan wajah

cv2.imshow('Pengenalan Wajah', img)

cv2.waitKey(0)

cv2.destroyAllWindows()

Pastikan Anda sudah melatih model dengan menggunakan dataset yang benar sebelum menjalankan kode ini. Ganti `image.jpg` dengan path gambar yang ingin Anda kenali.

Dalam artikel ini, kita telah menjelaskan langkah-langkah dasar untuk membuat sistem pengenalan wajah menggunakan library OpenCV dalam bahasa Python. Anda dapat mengembangkan dan menyesuaikan sistem ini dengan menambahkan fitur-fitur tambahan seperti pengenalan multiwajah, pengenalan waktu nyata, dan lain-lain. Semoga artikel ini dapat membantu Anda memulai dalam mengimplementasikan sistem pengenalan wajah yang efektif menggunakan OpenCV dan Python.

**Tutorial install Opencv-phyton :**

Berikut adalah langkah-langkah untuk menginstal OpenCV dengan Python menggunakan pip:

Langkah 1: Pastikan Python sudah terinstal

Pastikan Anda memiliki instalasi Python yang valid di sistem Anda. Anda dapat memeriksa apakah Python sudah terinstal dengan membuka terminal atau command prompt dan mengetikkan perintah berikut:

python --version

Jika Python sudah terinstal, Anda akan melihat versi Python yang terpasang. Jika tidak ada, silakan instal Python terlebih dahulu sebelum melanjutkan.

Langkah 2: Instalasi OpenCV dengan pip

Setelah memastikan Python terinstal, langkah selanjutnya adalah menginstal OpenCV menggunakan pip. Buka terminal atau command prompt, lalu jalankan perintah berikut:

pip install opencv-python

Perintah di atas akan menginstal paket OpenCV untuk Python. Jika Anda ingin menginstal versi yang mendukung fitur tambahan seperti pengolahan video atau pengenalan wajah, Anda juga dapat mencoba perintah berikut:

pip install opencv-python-headless

Langkah 3: Verifikasi instalasi

Setelah selesai menginstal, Anda dapat memverifikasi instalasi OpenCV dengan membuka terminal atau command prompt, lalu menjalankan perintah berikut:

Setelah masuk ke interpreter Python, ketikkan perintah berikut untuk memastikan OpenCV terinstal dengan benar:

import cv2

print(cv2.\_\_version\_\_)

Jika versi OpenCV ditampilkan tanpa kesalahan, itu berarti OpenCV berhasil terinstal di Python Anda.

Sekarang Anda telah berhasil menginstal OpenCV untuk Python menggunakan pip. Anda dapat memulai pengembangan aplikasi komputer vision menggunakan OpenCV dalam bahasa Python.