



UNIVERSITAS NEGERI JAKARTA

FAKULTAS TEKNIK

PROGRAM STUDI SISTEM & TEKNOLOGI INFORMASI

RENCANA TUGAS MAHASISWA

MATA KULIAH	Pengolahan Citra				
KODE MK	1519600069	SKS	3	SEMESTER	117
DOSEN PENGAMPU	Irma Permata Sari, S.Pd., M.Eng				

BENTUK TUGAS	WAKTU Pengerjaan
Mandiri	1 Minggu

JUDUL TUGAS

Tugas-1: Adding shapes and text to image

SUB CAPAIAN PEMBELAJARAN MATA KULIAH

DESKRIPSI TUGAS

Melakukan deteksi wajah dan membuat frame pada wajah yg terdeteksi menggunakan opencv pada python.

Tujuan Tugas

Praktikum ini bertujuan agar mahasiswa memahami:

1. Memahami Operasi Dasar Pengolahan Citra.
2. Memahami mengenai cara membuat shape dan text menggunakan opencv.
3. Memahami mengenai cara mendeteksi wajah menggunakan opencv.

Dasar Teori

OpenCV (Open Source Computer Vision Library) adalah sebuah library berisi fungsi-fungsi pemrograman yang ditujukan untuk keperluan computer vision secara realtime. Untuk menjalankan program opencv ini, perlu menginstall python versi terbaru. Kemudian menginstall opencv melalui pip pada python. Selain itu bisa juga menggunakan pycharm untuk lebih memudahkan menginstall package yang diperlukan.

Pembahasan

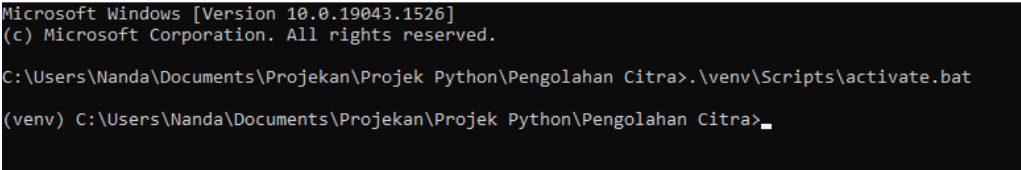
Praktikum ini menggunakan Visual Studio Code sebagai IDE nya dan juga menggunakan Python untuk bahasa pemrogramannya serta OpenCV untuk library-nya. Berikut ini langkah-langkah pembuatan programnya.

A. Instalasi

1. Pastikan python sudah terinstall.
2. Buat virtual environment menggunakan perintah berikut.
3. Kemudian masuk kedalam virtual environment dengan perintah berikut.

python -m venv .\venv

.\venv\Scripts\activate.bat



```
Microsoft Windows [Version 10.0.19043.1526]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Nanda\Documents\Projekan\Projek Python\Pengolahan Citra>.\venv\Scripts\activate.bat
(venv) C:\Users\Nanda\Documents\Projekan\Projek Python\Pengolahan Citra>_
```

4. Install opencv melalui command prompt menggunakan perintah berikut.

pip install opencv-python

5. Jika sudah lakukan perintah berikut untuk melihat apakah opencv sudah terinstall

pip list

```
(venv) C:\Users\Nanda\Documents\Projekan\Projek Python\Pengolahan Citra>pip list
Package      Version
-----
numpy        1.23.3
opencv-python 4.6.0.66
pip          22.2.2
setuptools   57.4.0

(venv) C:\Users\Nanda\Documents\Projekan\Projek Python\Pengolahan Citra>
```

B. Source Code

1. Import opencv kedalam file python

```
# Untuk import library opencv
import cv2
```

2. Import module pengenalan wajah, untuk modulnya sendiri bisa didownload pada akun resmi githun opencv melalui link berikut.

<https://github.com/opencv/opencv/tree/4.x/data/haarcascades>

```
# Untuk mengimport module pengenalan wajah
faceCascade = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
```

3. Inisialisasi webcam, disesuaikan dengan webcam setiap laptop masing-masing. Untuk webcam laptop saya menggunakan settingan tersebut.

```
# Untuk inisialisasi webcam, disesuaikan dengan webcam setiap laptop
cap = cv2.VideoCapture(2, cv2.CAP_DSHOW)
```

4. Membuat looping untuk memutar setiap frame

```
# Untuk melakukan perulangan secara terus-menerus
while True:
```

5. Memasukan data frame dan status webcam kedalam variable berikut.

```
# Untuk mendapatkan frame dan status webcam
ret, frame = cap.read()
```

6. Kemudian ubah menjadi warna abu-abu/gray.

```
# Untuk mengubah warna menjadi abu-abu / gray
imgGray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```

7. Setelah diubah warnanya, lanjut melakukan deteksi wajah menggunakan sintaks berikut. Parameter **scaleFactor** direkomendasikan antara 1.05 sampai 1.4, lebih kecil lebih jelas facedetectionnya, namun semakin lambat videonya karena beban cpu semakin berat, begitupun sebaliknya. Parameter **minNeighbors** direkomendasikan antara 3 – 6, semakin tinggi nilainya semakin sedikit mendeteksi wajah, namun kualitas deteksi bagus, begitu sebaliknya.

```
faces = faceCascade.detectMultiScale(  
    image=imgGray,  
    scaleFactor=1.09,  
    minNeighbors=5  
)
```

8. Selanjutnya membuat frame atau bingkai untuk wajah yang terdeteksi menggunakan sintaks berikut.

```
# Untuk membuat frame wajah sesuai data titik koordinat diatas  
for (x,y,w,h) in faces:  
    # Membuat frame wajah  
    cv2.rectangle(  
        img=frame,  
        pt1=(x, y-30),  
        pt2=(x+w, y+h),  
        color=(141, 15, 245),  
        thickness=2,  
    )  
    # Membuat filled rectangle untuk background text  
    cv2.rectangle(  
        img=frame,  
        pt1=(x,y),  
        pt2=(x+w, y-30),  
        color=(141, 15, 245),  
        thickness=-1,  
    )  
    # Membuat text yang akan ditampilkan  
    cv2.putText(  
        img=frame,  
        text="Nanda",  
        org=(x, y-10),  
        fontFace=cv2.FONT_HERSHEY_DUPLEX,  
        fontScale=0.5,  
        thickness=1,  
        color=(255,255,255)  
    )
```

9. Menampilkan frame yang telah melalui proses pengolahan.

```
# Untuk menampilkan frame yang sudah selesai diolah  
cv2.imshow("Video", frame)
```

10. Terakhir membuat sintaks untuk mencegah close dengan tombol silang dan hanya bisa close dengan tombol q pada keyboard.

```
# Untuk mencegah close dengan tombol silang dan hanya bisa close dengan tombol q pada keyboard  
if cv2.waitKey(1) & 0xFF == ord('q'):  
    break
```

Hasil

Berikut ini adalah hasil output dari tugas yang diberikan:



Kesimpulan

Kesimpulan dari praktikum ini adalah untuk memahami Operasi Dasar Pengolahan Citra, cara membuat shape dan text menggunakan opencv, dan cara mendeteksi wajah menggunakan opencv pada Python.

Listing Program

Berikut source code lengkap dari tugas yang telah diberikan:

```
# Untuk import library opencv
import cv2

# Untuk mengimport module pengenalan wajah
faceCascade = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")

# Untuk inialisasi webcam, disesuaikan dengan webcam setiap laptop
cap = cv2.VideoCapture(2, cv2.CAP_DSHOW)

# Untuk melakukan perulangan secara terus-menerus
while True:
    # Untuk mendapatkan frame dan status webcam
    ret, frame = cap.read()

    # Untuk mengubah warna menjadi abu-abu / gray
    imgGray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # Untuk melakukan deteksi wajah dan mendapatkan data titik koordinat

    # Parameter scaleFactor direkomendasikan antara 1.05 sampai 1.4,
    # lebih kecil lebih jelas facedetectionnya,
    # namun semakin lambat videonya karena beban cpu semakin berat, begitupun sebaliknya

    # Parameter minNeighbors direkomendasikan antara 3 - 6
    # Semakin tinggi nilainya semakin sedikit mendeteksi wajah
    # Namun kualitas deteksi bagus, begitu sebaliknya

    # Sesuaikan settingan ini dengan kebutuhan

    faces = faceCascade.detectMultiScale(
        image=imgGray,
        scaleFactor=1.09,
        minNeighbors=5
    )

    # Untuk membuat frame wajah sesuai data titik koordinat diatas
    for (x,y,w,h) in faces:
        # Membuat frame wajah
        cv2.rectangle(
            img=frame,
            pt1=(x, y-30),
            pt2=(x+w, y+h),
            color=(141, 15, 245),
            thickness=2,
        )
        # Membuat filled rectangle untuk background text
        cv2.rectangle(
            img=frame,
            pt1=(x,y),
            pt2=(x+w, y-30),
            color=(141, 15, 245),
            thickness=-1,
        )
        # Membuat text yang akan ditampilkan
        cv2.putText(
            img=frame,
            text="Nanda",
            org=(x, y-10),
            fontFace=cv2.FONT_HERSHEY_DUPLEX,
            fontScale=0.5,
            thickness=1,
            color=(255,255,255)
        )

    # Untuk menampilkan frame yang sudah selesai diolah
    cv2.imshow("Video", frame)

    # Untuk mencegah close dengan tombol silang dan hanya bisa close dengan tombol q pada keyboard
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
```