

Laporan Penugasan Teknologi Berkembang
Computer Vision (CV)



Disusun Oleh:
Ihsan Kamil Al Khozi
5026211117

Departemen Sistem Informasi
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
2024

Computer Vision (CV)

Computer vision adalah bidang kecerdasan buatan (AI) yang memungkinkan komputer untuk menafsirkan dan memahami informasi visual dari lingkungan sekitarnya. Bidang ini melibatkan pengembangan algoritma dan model yang memungkinkan komputer untuk memahami dan menafsirkan informasi visual dari gambar atau video. Tujuan dari computer vision adalah untuk memungkinkan komputer meniru dan meningkatkan kemampuan penglihatan manusia.

Key Elements Dalam Computer Vision

Berikut merupakan beberapa key elements dalam Computer Vision:

| Fitur | Penjelasan |
|---------------------|--|
| Image Recognition | Proses mengenali dan mengklasifikasikan gambar atau objek di dalam gambar berdasarkan pola yang telah dilatih oleh model. Misalnya, sebuah sistem dapat mengenali apakah sebuah gambar berisi kucing, anjing, atau benda lain dengan mengacu pada data pelatihan sebelumnya. |
| Object Detection | Bertujuan untuk menemukan dan mengidentifikasi lokasi spesifik objek dalam gambar. Misalnya, sistem dapat mendeteksi dan memberikan bounding box di sekitar mobil atau orang dalam sebuah gambar. |
| Image Segmentation | Teknik memisahkan gambar menjadi beberapa bagian (region) berdasarkan fitur tertentu, seperti warna atau tekstur, untuk lebih fokus pada objek tertentu. |
| Facial Recognition | Teknologi yang memungkinkan sistem mengenali dan mengidentifikasi wajah individu dari gambar atau video. Sistem facial recognition biasanya digunakan untuk verifikasi identitas, keamanan, atau pelacakan individu dalam sistem pengawasan. |
| Gesture Recognition | Proses mengidentifikasi gerakan tangan atau tubuh dari gambar atau video untuk menerjemahkan isyarat |

| | |
|-----------------|---|
| | tersebut menjadi perintah yang dapat dipahami komputer. Aplikasi ini banyak digunakan dalam kontrol antarmuka tanpa sentuhan atau augmented reality (AR). |
| Visual Tracking | Proses melacak pergerakan objek dalam video dari satu frame ke frame berikutnya. Ini sangat penting dalam aplikasi seperti pengawasan video, sistem navigasi kendaraan otonom, atau augmented reality, di mana objek atau individu perlu dipantau secara real-time sepanjang waktu. |

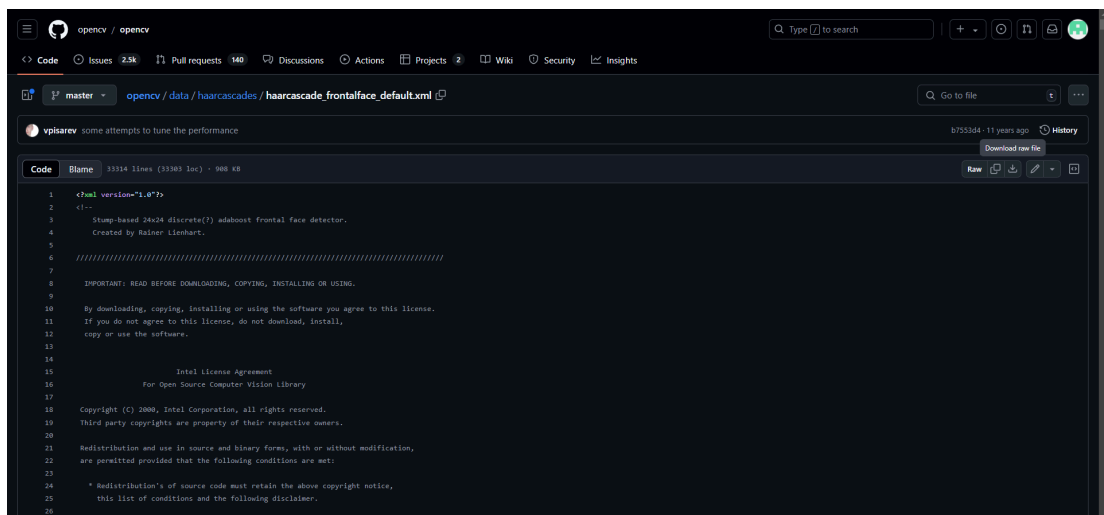
Percobaan Computer Vision - Face Recognition Simulation

Pada percobaan CV ini, saya menggunakan beberapa hal di bawah ini:

- Python dengan versi 3.13.0
- Library Python "OpenCV"
- File Haar Cascade

Berikut adalah langkah - langkah untuk melakukan percobaan Face Recognition:

1. Mengunduh file Haar Cascade



```

1  <?xml version="1.0"?>
2  <!--
3      Stump-based 24x24 discrete(?) adaboost frontal face detector.
4      Created by Rainer Lienhart.
5
6      =====
7
8      IMPORTANT: READ BEFORE DOWNLOADING, COPYING, INSTALLING OR USING.
9
10     By downloading, copying, installing or using the software you agree to this license.
11     If you do not agree to this license, do not download, install,
12     copy or use the software.
13
14
15     Intel License Agreement
16     For Open Source Computer Vision Library
17
18     Copyright (C) 2000, Intel Corporation, all rights reserved.
19     Third party copyrights are property of their respective owners.
20
21     Redistribution and use in source and binary forms, with or without modification,
22     are permitted provided that the following conditions are met:
23
24     * Redistribution's of source code must retain the above copyright notice,
25       this list of conditions and the following disclaimer.
26

```

File tersebut berfungsi sebagai model yang digunakan oleh algoritma deteksi objek berbasis Haar Cascade di OpenCV. Secara khusus, file ini berisi serangkaian fitur yang telah dilatih untuk mengenali pola tertentu dalam gambar, seperti wajah manusia.

2. Menulis script code untuk membuat dataset wajah secara otomatis

```

face_create_dataset.py > ...
1  import cv2
2  import os
3
4  faceCascade = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
5  cap = cv2.VideoCapture(0)
6  dataset_path = "dataset/"
7
8  if not os.path.exists(dataset_path):
9      os.mkdir(dataset_path)
10
11  person_id = 1 # id for person that we will detect
12  count = 0 # count for image name id
13  while True:
14      _, frame = cap.read()
15      gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
16      faces = faceCascade.detectMultiScale(frame, scaleFactor=1.1,
17                                          minNeighbors=5, minSize=(30, 30))
18
19      for (x, y, w, h) in faces:
20          cv2.rectangle(frame, (x, y), (x+w, y+h), (255, 0, 0), 2)
21          count += 1
22          cv2.imwrite(dataset_path + "Person-" + str(person_id)
23                    + "-" + str(count) + ".jpg", gray[y:y+h, x:x+w])
24
25          cv2.imshow("Camera", frame)
26
27          if cv2.waitKey(1) == ord('q'):
28              break
29          elif count == 30: # stop when 30 photos have been taken
30              break
31
32  cap.release()
33  cv2.destroyAllWindows()

```

Ketika kode ini dijalankan, maka sistem akan secara otomatis mengambil wajah dari seseorang.

3. Menulis script code untuk melatih model yang akan digunakan

```

face_training.py > ...
1  import cv2
2  import numpy as np
3  import os
4
5  def checkDataset(directory="dataset/"):
6      if os.path.exists(directory) and len(os.listdir(directory)) != 0:
7          return True
8      return False
9
10 def organizeDataset(path="dataset/"):
11     imagePath = [os.path.join(path, p) for p in os.listdir(path)]
12     faces = []
13     ids = np.array([], dtype="int")
14     for i in imagePath:
15         img = cv2.cvtColor(cv2.imread(i), cv2.COLOR_BGR2GRAY)
16         id = int(i.split("/")[-1].split(".")[0].split("-")[1])
17         face = faceCascade.detectMultiScale(img)
18         for (x, y, w, h) in face:
19             faces.append(img[y:y+h, x:x+w])
20             ids = np.append(ids, id)
21     return faces, ids
22
23 if not checkDataset():
24     print("Dataset not found")
25 else:
26     recognizer = cv2.face.LBPHFaceRecognizer.create()
27     faceCascade = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
28
29     # train faces
30     print("Training faces...")
31     faces, ids = organizeDataset()
32     recognizer.train(faces, ids)
33     print("Training finished!")
34
35     # save model
36     recognizer.write("face-model.yml")
37     print("Model saved as 'face-model.yml'")

```

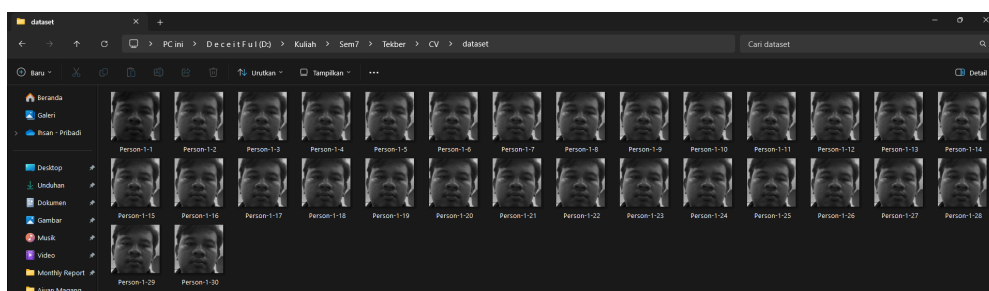
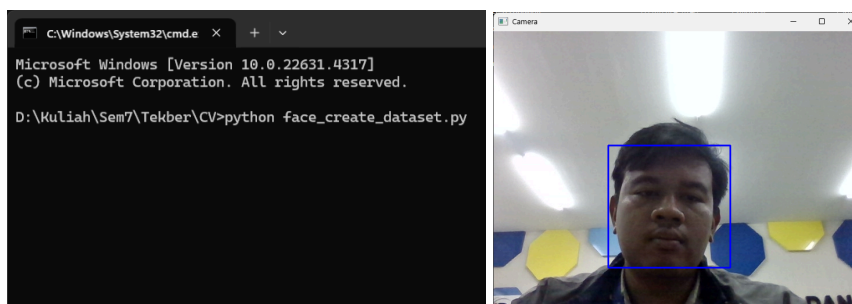
Setelah membuat dataset wajah, selanjutnya data tersebut akan dilatih dengan menggunakan kode ini. Model yang dihasilkan setelah hasil dari pelatihan data tadi akan dinamai dengan **face-model.yml**

4. Menulis script code untuk melakukan face recognition

```
face_recognition.py > ...
1 import cv2
2
3 recognizer = cv2.face.LBPHFaceRecognizer.create()
4 recognizer.read("face-model.yml") # face model from face_training.py
5 faceCascade = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
6 font = cv2.FONT_HERSHEY_COMPLEX
7
8 id = 0
9 names = ['None', 'Ihsan']
10 cap = cv2.VideoCapture(0)
11
12 while True:
13     _, frame = cap.read()
14     gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
15     faces = faceCascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5, minSize
16
17     for (x, y, w, h) in faces:
18         cv2.rectangle(frame, (x, y), (x+w, y+h), (255, 0, 0), 2)
19         id, confidence = recognizer.predict(gray[y:y+h, x:x+w])
20
21         if confidence < 100:
22             id = names[id]
23         else:
24             id = "unknown"
25         confidence = "{:.0f}%".format(round(100 - confidence))
26
27         cv2.putText(frame, str(id), (x+5, y-5), font, 1, (255, 0, 0), 1)
28         cv2.putText(frame, str(confidence), (x+5, y+h-5), font, 1, (255, 255, 0), 1)
29
30     cv2.imshow("Camera", frame)
31     if cv2.waitKey(1) == ord('q'):
32         break
33
34 cap.release()
35 cv2.destroyAllWindows()
```

Setelah melatih modelnya, maka kita akan mencoba melakukan face recognition. Script code ini digunakan untuk melakukan face recognition.

5. Menjalankan file face_create_dataset.py



Ketika file tersebut dijalankan, maka sistem akan secara otomatis mengambil gambar wajah kita sebanyak 30 kali melalui kamera. Hasil dari pengambilan wajah tersebut akan disimpan di folder bernama **dataset**.

6. Menjalankan file face_training.py

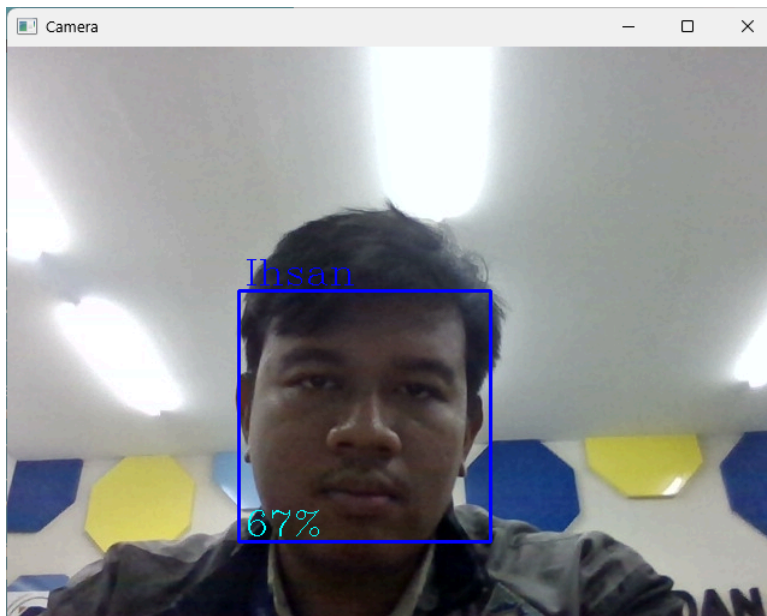
```
D:\Kuliah\Sem7\Tekber\CV>python face_training.py
Training faces...
[ERROR:0@0.856] global ocl.cpp:1006 cv::ocl::OpenCLExecutionContext::Impl::getInitializedExecutionContext OpenCL: Can't
create default OpenCL queue
Training finished!
Model saved as 'face-model.yml'

D:\Kuliah\Sem7\Tekber\CV>
```

| Nama | Tanggal diubah | Jenis | Ukuran |
|-------------------------------------|------------------|------------------|----------|
| dataset | 17/10/2024 17:11 | Folder file | |
| face_create_dataset | 17/10/2024 17:08 | Python File | 1 KB |
| face_recognition | 17/10/2024 16:48 | Python File | 2 KB |
| face_training | 17/10/2024 16:45 | Python File | 2 KB |
| haarcascade_frontalface_default.xml | 17/10/2024 17:05 | File XML | 909 KB |
| face-model | 17/10/2024 17:18 | Yaml Source File | 4.138 KB |

Ketika file tersebut dijalankan, maka sistem akan secara otomatis melakukan training terhadap dataset wajah yang telah kita ambil tadi. Hasil training model tersebut akan disimpan sebagai **face-model.yml**

7. Menjalankan file face_recognition.py



Setelah file dijalankan, maka kamera akan menunjukkan hasil dari face recognition yang telah dilakukan beserta label nama dan tingkat akurasinya.