

LAPORAN TUGAS BESAR 2

IF2123 ALJABAR LINIER DAN GEOMETRI



Anggota kelompok Crowd:

Jonet Wira Murti	13518083
Lionnarta Savirandy	13518128
Stefanus Ardi Mulia	13517119

PROGRAM STUDI INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2019

BAB I

DESKRIPSI MASALAH

Di dalam tugas besar ini, pokok permasalahan yang diangkat adalah membuat sistem pengenalan wajah dengan menggunakan matrik kemiripan berbasis vektor. Input untuk sistem adalah sebuah citra wajah yang ditanyakan, sedangkan koleksi citra wajah di dalam basis data diasumsikan sudah tersedia. Fitur-fitur dari wajah diekstraksi, begitu pula fitur-fitur wajah di dalam basis data. Pencocokan kemiripan dilakukan antara vektor fitur citra wajah input dengan vektor-vektor fitur wajah di dalam basis data. Hasil pencocokan diurutkan dari yang paling mirip hingga yang kurang mirip dengan menggunakan nilai ambang T , yaitu T gambar dalam basis data yang paling mirip dengan citra wajah input. Program akan dibuat dengan Bahasa Python dengan memanfaatkan sejumlah library di Python, salah satunya OpenCV(*Computer Vision*). Fungsi untuk mengekstraksi fitur dari sebuah citra wajah sudah tersedia dalam library OpenCV, sehingga tinggal digunakan. Beberapa algoritma untuk ekstraksi fitur yaitu SURF, ORB, SIFT, BRIEF. Kebanyakan dari algoritma ini didasarkan pada image gradient. Pada tugas kali ini, kami akan menggunakan descriptor KAZE karena sudah tersedia di dalam pustaka OpenCV.

BAB II TEORI SINGKAT

A. Vektor

Vektor merupakan besaran yang memiliki besar atau panjang dan arah. Secara umum vektor direpresentasikan dalam gambar panah dengan arah vektor merupakan arah dari panah dan besar atau panjangnya merupakan panjang dari panah. Bagian ekor dari panah disebut titik pangkal, sedangkan kepala dari panah disebut titik ujung vektor. Vektor biasanya dinotasikan sebagai huruf kecil yang ditebalkan seperti : **a**, **b**, **c**, atau sebagai huruf kecil yang ditulis miring seperti : *a*, *b*, *c*.

Pada vektor terdefinisi operasi-operasi serta aturan-aturan. operasi-operasi ini diantaranya, operasi penjumlahan, operasi pengurangan, serta operasi perkalian skalar.

B. Norma Euclidean

Norma Euclidean atau norma dari sebuah vektor **v** didefinisikan sebagai panjang atau besar dari vektor tersebut dan dinotasikan seperti $\|\mathbf{v}\|$. Jika $\mathbf{v} = (v_1, v_2, \dots, v_n)$ adalah vektor pada R_n , maka norma dari **v** didefinisikan sesuai formula $\|\mathbf{v}\| = \sqrt{v_1^2 + v_2^2 + \dots + v_n^2}$.

C. Perkalian Titik

Perkalian titik adalah suatu operasi perkalian vektor yang kedua operandnya harus merupakan vektor juga. Jika **u** dan **v** adalah vektor pada R_n , maka perkalian titik antara **u** dan **v** didefinisikan berdasarkan formula $\mathbf{u} \cdot \mathbf{v} = \|\mathbf{u}\| \|\mathbf{v}\| \cos(\theta)$, dengan θ merupakan sudut yang dibentuk antara **u** dan **v**. Jika $\mathbf{u} = (u_1, u_2, u_3, \dots, u_n)$ dan $\mathbf{v} = (v_1, v_2, v_3, \dots, v_n)$ merupakan vektor pada R_n , maka perkalian titik antara **u** dan **v** dapat juga didefinisikan berdasarkan formula $\mathbf{u} \cdot \mathbf{v} = u_1v_1 + u_2v_2 + u_3v_3 + \dots + u_nv_n$.

D. Pengenalan Wajah

Pengenalan wajah merupakan suatu teknologi untuk mengidentifikasi wajah orang dengan mencocokkan fitur-fitur yang diekstraksi dari wajah yang diinput melalui gambar digital dengan data yang disimpan dalam basis data.

E. OpenCV

OpenCV merupakan suatu pustaka dengan sumber terbuka yang menangani permasalahan-permasalahan penglihatan komputer dan pembelajaran mesin.

F. KAZE Features

KAZE Features adalah metode pendeteksian dan deskripsi fitur 2D yang beroperasi dalam ruang skala nonlinear. *KAZE Features* melakukan deteksi dan deskripsi pada suatu citra pada ruang nonlinear dengan melakukan filter difusi nonlinear.

BAB III IMPLEMENTASI PROGRAM

A. Feature Extraction

Feature extraction pada program kami diimplementasikan dengan menggunakan KAZE *features* sebagai Berikut ini adalah potongan kode yang digunakan untuk mengekstrak fitur pada gambar. Pertama, *keypoints* dari gambar yang diekstrak akan ditampung pada sebuah array. kemudian, array yang berisi *keypoints* ini akan diurutkan berdasarkan nilai responsnya, kemudian diambil 32 nilai pertama. Untuk setiap *keypoints* di array, akan dibuat *descriptor*-nya. *descriptor* ini yang kemudian dijadikan sebagai vektor untuk dibandingkan menggunakan cosine similarity dan euclidean distance dengan vektor lainnya.

Berikut adalah implementasinya pada Python :

```
def extract_features(image_path, vector_size=32):
    """
    Feature extractor by Andrey Nikishaev modified for Python 3 by Ardi.
    https://medium.com/machine-learning-world/feature-extraction-and-simil
    """
    image = cv2.imread(image_path)
    try:
        alg = cv2.KAZE_create()
        kps = alg.detect(image)
        kps = sorted(kps, key=lambda x: -x.response)[:vector_size]
        kps, dsc = alg.compute(image, kps)
        dsc = dsc.flatten()
        needed_size = (vector_size * 64)
        if dsc.size < needed_size:
            dsc = np.concatenate([dsc, np.zeros(needed_size - dsc.size)])
    except cv2.error as e:
        print(f'Error: ', e)
        return None

    return dsc
```

B. Cosine Similarity

Cosine similarity merupakan algoritma yang kami gunakan untuk menganalisis kemiripan antara 2 gambar dengan memanfaatkan perkalian titik antara dua *descriptor* dan mengambil nilai cosinusnya. Jika cosinusnya makin mendekati 1, maka dua buah gambar semakin mirip.

Dalam implementasinya pada program kami, untuk setiap vektor *descriptor* dalam database akan dilakukan operasi perkalian titik dengan *descriptor* gambar yang dipilih pengguna, kemudian diambil T gambar yang paling mirip dengan gambar pilihan.

Berikut ini adalah implementasi program pada python :

```
def cos_cdist(self, vector):
    # getting cosine distance between search image and images database
    result = []
    for dsc in self.matrix:
        dot = 0
        dsc_squaresum = 0
        vec_squaresum = 0
        for i in range(0, len(dsc)):
            dot += dsc[i] * vector[i]
            dsc_squaresum += dsc[i] ** 2
            vec_squaresum += vector[i] ** 2

        cos = dot / (math.sqrt(dsc_squaresum) * math.sqrt(vec_squaresum))
        result.append(1 - cos)

    return result
```

C. Euclidean Distance

Euclidean distance merupakan algoritma yang kami gunakan untuk menganalisis kemiripan antara 2 gambar dengan memanfaatkan jarak euclidean antara 2 vektor, yaitu vektor *descriptor* gambar yang dipilih pengguna dan vektor *descriptor* untuk setiap gambar dalam database. Jika jarak euclidean antara 2 vektor semakin mendekati 0, maka kedua vektor tersebut semakin mirip.

Dalam implementasinya pada program kami, untuk setiap vektor *descriptor* dalam database akan dilakukan operasi jarak euclidean dengan vektor *descriptor* dari gambar yang dipilih pengguna, kemudian ditampilkan T gambar yang paling mirip dengan gambar pilihan.

Berikut ini adalah implementasi program pada python :

```
def euclidean_cdist(self, vector):
    # getting euclidean distance between search image and images database
    result = []
    for dsc in self.matrix:
        squaresum = 0
        for i in range(0, len(dsc)):
            squaresum += (vector[i] - dsc[i]) ** 2

        eucl_dist = math.sqrt(squaresum)
        result.append(eucl_dist)

    return result
```

D. GUI

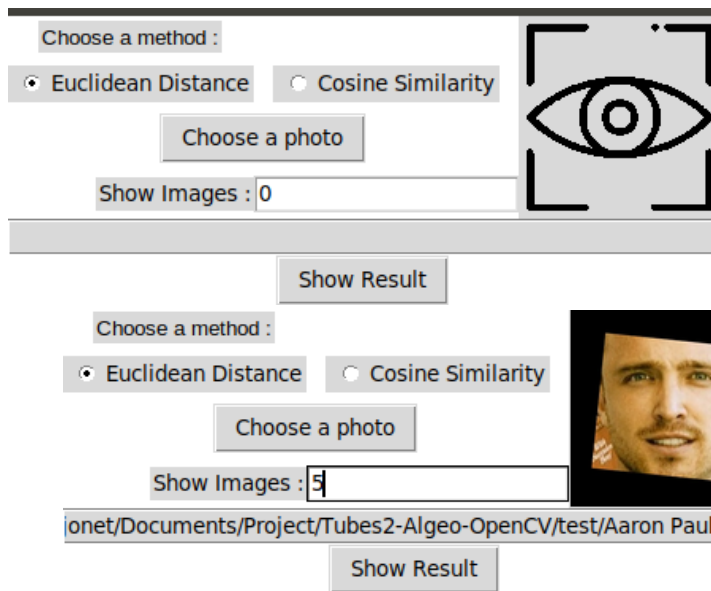
GUI (*graphical user interface*) merupakan suatu komponen interaksi seperti *icon* dan objek grafika lainnya untuk menolong pengguna berinteraksi dengan perangkat lunak komputer, seperti sistem operasi.

Dalam implementasi pada program ini, GUI kami akan meminta pengguna memilih metode yang ingin digunakan untuk pencocokan. Setelah itu pengguna dapat memilih foto sebagai bahan uji dan memasukkan nilai ambang sebagai jumlah foto dengan tingkat kecocokan tertinggi dengan masukan. Setelah itu pengguna dapat melihat hasil setelah menekan tombol "Show Result" dan menampilkan foto-foto dalam basis data yang sesuai.

BAB IV EKSPERIMEN

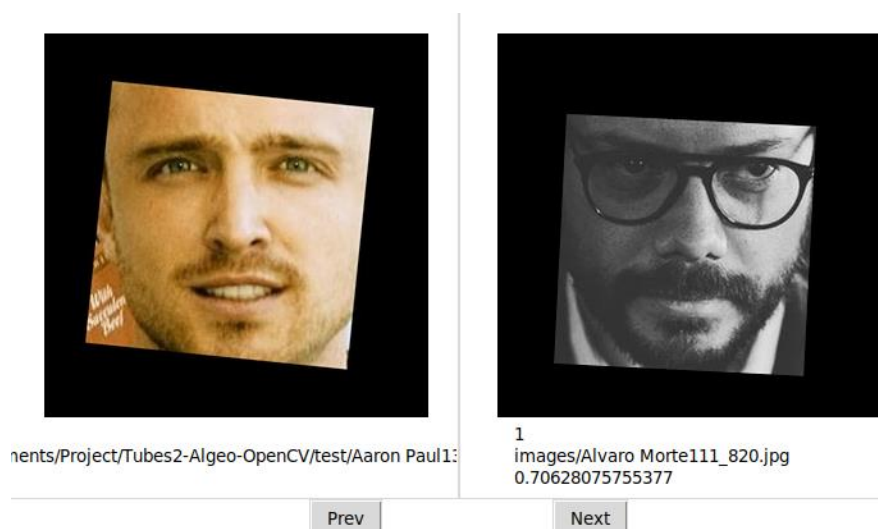
A. Start Page

Berikut ini adalah beberapa tampilan awal dari GUI kami yang digunakan pengguna untuk memilih gambar input, metode pencocokan gambar, serta nilai ambang yang digunakan untuk menampilkan banyaknya wajah yang mirip dengan inputnya.



B. Cosine Similarity

Berikut ini adalah gambar-gambar hasil pencocokan dengan metode Cosine similarity, dengan nilai ambang 5.





nents/Project/Tubes2-Algeo-OpenCV/test/Aaron Paul13

Prev



2
images/Alvaro Morte79_810.jpg
0.6848060382929381

Next



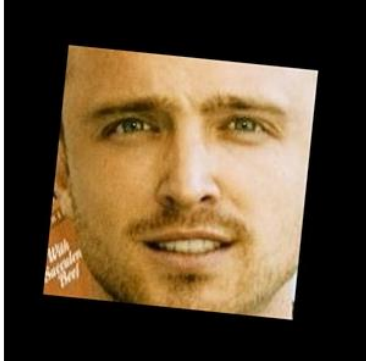



nents/Project/Tubes2-Algeo-OpenCV/test/Aaron Paul13

Prev



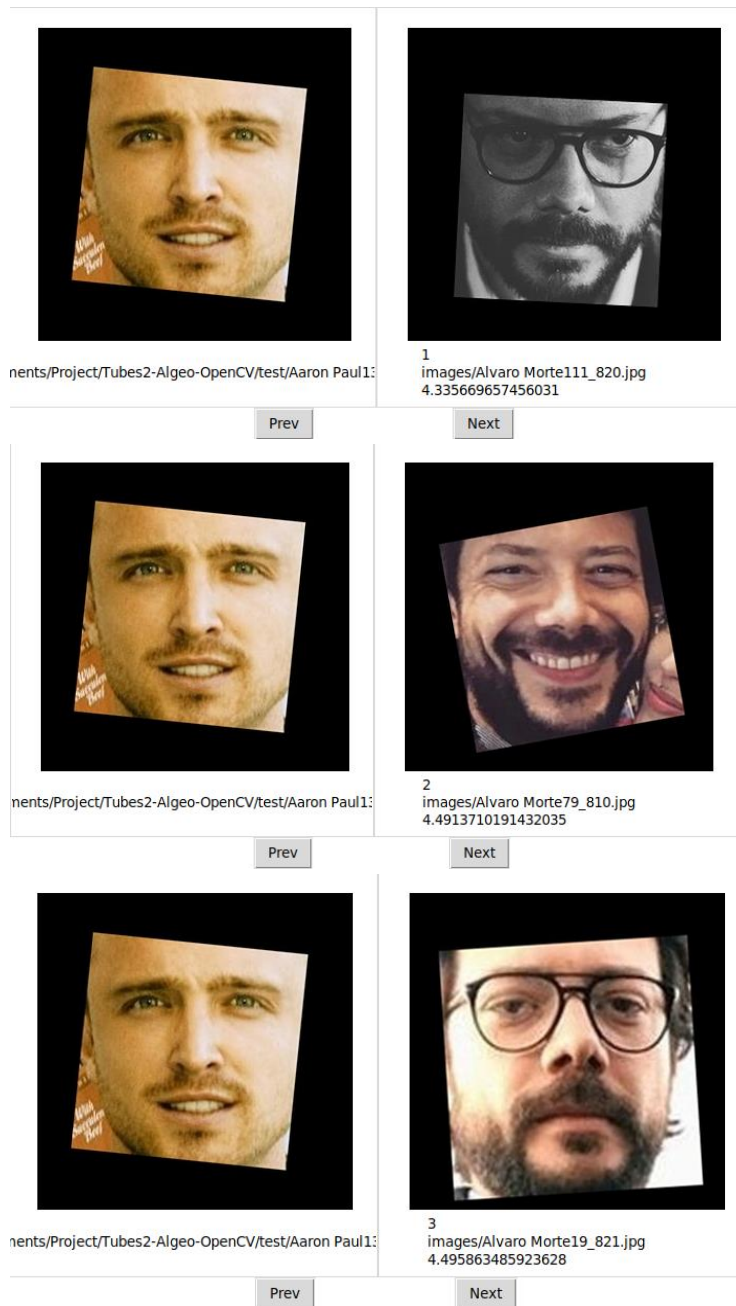
3
images/Alvaro Morte19_821.jpg
0.684175184717092

Next

 <p>nents/Project/Tubes2-Algeo-OpenCV/test/Aaron Paul1:</p>	 <p>4 images/alicia debnam carey face35.jpg 0.6807753258882597</p>
Prev	Next
 <p>nents/Project/Tubes2-Algeo-OpenCV/test/Aaron Paul1:</p>	 <p>5 images/Alvaro Morte2_863.jpg 0.6796428771658131</p>
Prev	Next

C. Euclidean Distance

Berikut ini adalah gambar-gambar hasil pencocokan dengan metode Euclidean distance, dengan nilai ambang 5.





nents/Project/Tubes2-Algeo-OpenCV/test/Aaron Paul13



4
images/alicia debnam carey face35.jpg
4.519997696899443

Prev

Next



nents/Project/Tubes2-Algeo-OpenCV/test/Aaron Paul13



5
images/Alvaro Morte2_863.jpg
4.528007905980778

Prev

Next

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Pendeteksian wajah yang dilakukan dengan pengekstrakan fitur menggunakan KAZE *Features* yang kemudian dibandingkan dengan *cosine similarity* ataupun *euclidean distance* menghasilkan keluaran yang tidak konsisten.

Analisis kami yang pertama, KAZE *Features* yang memang dibuat bukan untuk melakukan ekstraksi fitur wajah sangat berpengaruh pada hasilnya karena posisi titik fitur yang diambil pada setiap gambar wajah menjadi tidak konsisten sehingga vektor deskriptornya menjadi tidak merepresentasikan wajah pada gambar tersebut. Karena vektor deskriptornya tidak merepresentasikan wajah dengan baik, pada perbandingan antara dua vektor deskriptor menggunakan *cosine similarity* ataupun *euclidean distance* menjadi tidak mirip.

B. Saran

Metode pengekstrakan fitur wajah dapat diganti menjadi metode/pustaka yang melakukan ekstraksi fitur khusus wajah, semisal *facial landmark detector* pada dlib yang menghasilkan vektor pada posisi yang telah terdefinisi pada suatu wajah sehingga perbandingan menggunakan *cosine similarity* ataupun *euclidean distance* menjadi lebih relevan.

C. Refleksi

Dari program yang dibuat, kami jadi lebih mengerti pemrograman menggunakan bahasa python, serta dapat mengenal pustaka OpenCV yang dapat digunakan untuk penglihatan komputer dan pembelajaran komputer. Selain itu kami juga lebih memahami kegunaan operasi-operasi vektor pada dunia informatika.

REFERENSI

- [1] Pablo Fernández Alcantarilla, Adrien Bartoli, and Andrew J. Davison. 2012. KAZE features. In Proceedings of the 12th European conference on Computer Vision - Volume Part VI (ECCV'12), Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid (Eds.), Vol. Part VI. Springer-Verlag, Berlin, Heidelberg, 214-227.
- [2] Anton, Howard dan Rorres, Chris. *Elementary Linear Algebra*.