

DETEKSI WAJAH MENGGUNAKAN CASCADE CLASSIFIER DENGAN OPENCV-PYTHON

Angga Putra Prakoso¹, Abdurahman Rasyid^{2*}, Alvito Deannova³ dan Agung Edi Rahmawan⁴

^{1,2,3,4}Teknik Informatika, Fakultas Teknik, Universitas Krisnadwipayana

Email : anggaputraprakoso@gmail.com

Abstrak

Deteksi wajah merupakan aspek penting dalam bidang Computer Vision yang memiliki aplikasi luas, termasuk dalam sistem keamanan, interaksi manusia dengan komputer, dan analisis emosi. Salah satu metode yang efektif untuk deteksi wajah adalah menggunakan Cascade Classifier, khususnya metode haar cascade Classifier. Penelitian ini mengimplementasikan algoritma haar cascade classifier menggunakan OpenCV dan Python untuk mendeteksi wajah pada citra digital. Metode ini menunjukkan kecepatan komputasi yang tinggi, menjadikannya cocok untuk aplikasi real-time seperti keamanan dan pengenalan wajah di kamera ponsel. Eksperimen menunjukkan hasil deteksi wajah yang akurat, meskipun terdapat keterbatasan dalam mendeteksi wajah dengan pose atau ekspresi yang beragam. Untuk meningkatkan akurasi, diperlukan pelatihan menggunakan dataset yang lebih representatif. Secara keseluruhan, Cascade Classifier dengan OpenCV dan Python efektif untuk deteksi wajah real-time.

Abstract

Face detection is an important aspect of the Computer Vision field with wide applications, including security systems, human-computer interaction, and emotion analysis. One effective method for face detection is using a Cascade Classifier, specifically the haar cascade classifier method. This research implements the haar cascade classifier algorithm using OpenCV and Python to detect faces in digital images. This method exhibits high computational speed, making it suitable for real-time applications such as security and face recognition in mobile phone cameras. Experiments show accurate face detection results, although there are limitations in detecting faces with diverse poses or expressions. To improve accuracy, training using a more representative dataset is required. Overall, the Cascade Classifier with OpenCV and Python is effective for real-time face detection.

Kata kunci: *Deteksi wajah, Haarcascade Classifier, OpenCV dan Python.*

PENDAHULUAN

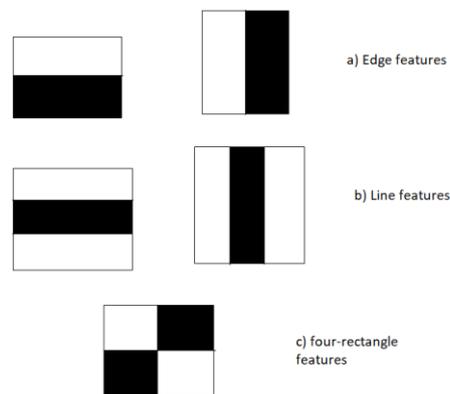
Dalam dunia yang semakin tergantung pada teknologi, *Computer Vision* telah menjadi bidang yang kritis dalam pengembangan sistem yang mampu menginterpretasi dan memahami dunia visual di sekitarnya. Salah satu aplikasi utama dari *Computer Vision* yang memiliki dampak signifikan adalah deteksi wajah.

Deteksi wajah adalah suatu proses yang mengidentifikasi dan mengekstrak fitur-fitur wajah dari gambar atau video. Kemampuan ini bukan hanya menjadi elemen penting dalam aplikasi sistem keamanan, tetapi juga dalam pengembangan interaksi manusia dengan komputer, analisis emosi, dan banyak lagi. deteksi wajah juga mengacu pada proses psikologi di mana manusia menemukan dan merawat wajah dalam adegan visual. dalam hal ini terdapat dua label kelas, wajah dan non-wajah. (1)

Pentingnya deteksi wajah semakin mendapat perhatian seiring dengan kemajuan teknologi. Algoritma dan teknik deteksi yang terus berkembang memungkinkan aplikasi yang lebih canggih dan akurat. Salah satu metode yang telah terbukti efektif dalam deteksi wajah adalah *Cascade Classifier*, Metode *Cascade Classifier* telah terbukti sebagai salah satu pendekatan yang efektif untuk

deteksi wajah. OpenCV digunakan untuk implementasi algoritma *haar cascade classifier* agar sistem dapat mendeteksi wajah pada citra digital (2), metode *haar cascade classifier* mempunyai kelebihan yaitu komputasi yang sangat cepat, karena hanya tergantung pada jumlah piksel dalam persegi bukan setiap nilai piksel dari sebuah image. (3)

Haar cascade adalah algoritma yang telah dilatih menggunakan metode Viola-Jones. Algoritma ini sangat populer dalam deteksi wajah, dan dapat diterapkan pada deteksi objek lainnya. *haar cascade classifier* juga merupakan *rectangular* (persegi) *feature*, yang memberikan indikasi secara spesifik pada sebuah gambar atau image. (4) Kecepatan dan efisiensi Cascade Classifier membuatnya menjadi pilihan yang populer untuk aplikasi deteksi objek real-time, seperti dalam kamera keamanan, pengenalan wajah di kamera ponsel, atau aplikasi augmented reality.



Gambar 1 Fitur *haar cascade classifier* yang paling dominan digunakan

Haar cascade mengekstraksi fitur dari gambar menggunakan sebuah “filter” mirip dengan konsep kernel konvolusional. Filter ini disebut fitur Haar. (2)

OpenCV (*Open Source Computer Vision Library*) adalah perpustakaan sumber terbuka yang menyediakan alat dan fungsi untuk pengembangan aplikasi *computer vision*. Tujuannya adalah memberikan kemudahan akses dan efisiensi kepada pengembang untuk membuat aplikasi computer vision yang berkinerja tinggi. OpenCV memiliki fungsi-fungsi yang mencakup berbagai tugas computer vision, termasuk deteksi objek, pengenalan wajah, pengolahan citra, kalibrasi kamera, dan machine learning. OpenCV memiliki lebih dari 2500 algoritma pengoptimalan, termasuk satu set lengkap algoritma pembelajaran mesin dan pembelajaran komputer klasik dan mutakhir. (5)

Python adalah bahasa pemrograman tingkat tinggi yang terkenal karena sintaksisnya yang mudah dipahami dan keterbacaan kode yang tinggi. Diciptakan pada tahun 1991, Python telah menjadi salah satu bahasa pemrograman paling populer di dunia. Keunikan Python meliputi kemampuan interpretatifnya, yang memungkinkan eksekusi kode tanpa proses kompilasi sebelumnya, serta sifat dinamisnya yang memungkinkan tipe data ditentukan secara otomatis saat runtime. Bahasa ini mendukung berbagai paradigma pemrograman, termasuk prosedural, berorientasi objek, dan fungsional. Fleksibilitas ini memungkinkan pengembang untuk memilih pendekatan pemrograman yang paling sesuai dengan kebutuhan proyek mereka. Dengan kemampuan ini, Python digunakan secara luas dalam berbagai bidang, termasuk pengembangan web, ilmu data, kecerdasan buatan, dan pengembangan perangkat lunak secara umum. Keberhasilan Python tidak hanya terletak pada kekuatan teknisnya, tetapi juga pada komunitasnya yang besar, dukungan yang kuat, dan ekosistem modul yang kaya. Penelitian ini bertujuan untuk mengaplikasikan *Cascade Classifier* dengan menggunakan OpenCV dan Python untuk mendeteksi wajah pada citra digital, dengan harapan dapat meningkatkan kecepatan dan akurasi deteksi wajah dalam aplikasi *real-time*.

METODE PELAKSANAAN

2.1 Studi Literatur

Tahap pertama yang dilakukan adalah mengumpulkan dan mempelajari literatur-literatur yang berkaitan dengan deteksi wajah dan *Haar Cascade Classifier*. Pentingnya tahap studi literatur adalah untuk memastikan bahwa penelitian yang dilakukan tidak hanya membangun pada pengetahuan yang

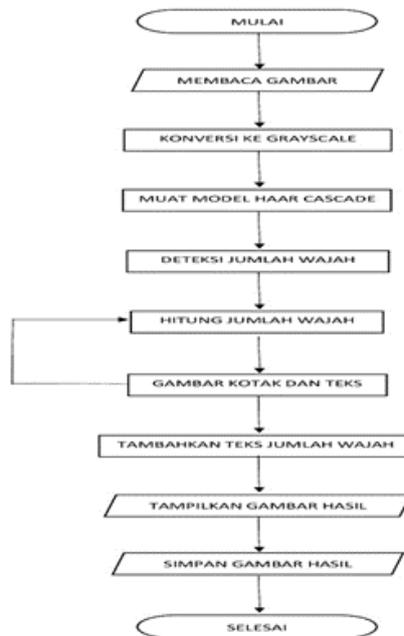
sudah ada, tetapi juga menggambarkan pemahaman yang kuat terhadap konsep-konsep yang relevan dalam bidang deteksi wajah dan penggunaan *Haar Cascade Classifier*.

2.2 Pengumpulan Data Citra

Ditahap ini, data berupa gambar citra dikumpulkan sebagai citra inputan, gambar citra diambil dari foto masing-masing anggota kelompok yang kemudian disimpan. Jumlah citra gambar yang diambil untuk diuji berjumlah 9 citra, beberapa citra diambil dari foto citra anggota kelompok dan sisinya diambil dari internet, selain data citra file yang berisi model *haarcascade* bernama *Haarcascade_frontalface_default.xml* di simpan, file tersebut berisi informasi yang diperlukan untuk mengidentifikasi pola wajah menggunakan fitur Haar.

2.3 Pengembangan Sistem

Dalam tahap ini OpenCV dimanfaatkan untuk menerapkan algoritma *haar cascade classifier* sehingga sistem dapat mengidentifikasi wajah pada gambar digital. Input sistem berupa gambar berwarna yang memiliki variasi jumlah objek wajah. Berikut adalah flowchart dari sistem deteksi wajah menggunakan *cascade classifier* dengan *opencv-python*.



Gambar 2 Flowchart dari sistem deteksi wajah menggunakan cascade classifier dengan *opencv-python*

Pertama impor modul OpenCV dengan menggunakan alias `cv2`, kemudian input gambar citra yang mengandung wajah untuk dideteksi. Awalnya, gambar tersebut adalah gambar tiga lapis (yaitu RGB), sehingga diubah menjadi gambar satu lapis (yaitu skala abu-abu), masukan modul *Haarcascade*. Hal ini dilakukan dengan menggunakan metode `cv2.CascadeClassifier::detectMultiScale`, yang mengembalikan persegi batas untuk wajah yang terdeteksi (yaitu, `x`, `y`, `w`, `h`). Dibutuhkan dua parameter yaitu, `scaleFactor` dan `minNeighbors`. `ScaleFactor` menentukan faktor peningkatan ukuran jendela yang awalnya dimulai pada ukuran “`minSize`”, dan setelah menguji semua jendela dengan ukuran tersebut, jendela diperbesar dengan “`scaleFactor`”, dan ukuran jendela naik menjadi “`maxSize`”. Jika “`scaleFactor`” besar, (misalnya, 2.0), langkah yang diperlukan akan lebih sedikit, sehingga deteksi akan lebih cepat, namun kita mungkin melewati objek yang ukurannya berada di antara dua skala yang diuji. (faktor skala `default` adalah 1,3). Semakin tinggi nilai “`minNeighbors`”, semakin sedikit jumlah positif palsu, dan semakin

sedikit kesalahan dalam hal deteksi wajah yang salah. Namun, ada kemungkinan beberapa jejak wajah yang tidak jelas juga hilang, Persegi panjang digambar di sekitar wajah yang terdeteksi dengan metode persegi panjang pada modul cv2 dengan melakukan iterasi pada semua wajah yang terdeteksi.

```
import cv2

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

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

haar_cascade =
cv2.CascadeClassifier('Haarcascade_frontalface_default.xml')

faces_rect = haar_cascade.detectMultiScale(gray_img, 1.1, 9)

num_faces = len(faces_rect)

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

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

cv2.putText(img, f'jumlah yang terdeteksi: {num_faces}
wajah',
(10, 30), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255),
2)

cv2.imshow('Detected faces', img)

cv2.waitKey(0)

output_file = 'detected_faces.jpg'

cv2.imwrite(output_file, img)
```

HASIL DAN PEMBAHASAN

Gambar citra yang diinputkan kedalam sistem dapat menghasilkan hasil yang baik. Sistem dapat mendeteksi wajah dan jumlah wajah dengan cukup akurat. Hasil dari beberapa output dapat dilihat pada gambar dibawah ini.



Gambar 3 Deteksi wajah dan jumlah wajah



Gambar 4 Deteksi wajah dan jumlah wajah



Gambar 5 Deteksi wajah dan jumlah wajah

Dapat terlihat bahwa sistem dapat mendeteksi wajah – wajah yang ada pada citra dengan akurat, waktu eksekusi juga relatif cepat sehingga cocok untuk aplikasi real-time, Wajah yang telah terdeteksi diperlihatkan pada bagian kotak persegi diposisi wajah. Metode Cascade Classifier berbasis Haar-like features telah terbukti efektif dalam deteksi objek, khususnya wajah. Kecepatan dan efisiensi Cascade Classifier membuatnya menjadi pilihan populer untuk aplikasi real-time seperti keamanan, pengenalan wajah di kamera ponsel, dan *augmented reality*.



Gambar 6 Deteksi wajah dan jumlah wajah



Gambar 7 Deteksi wajah dan jumlah wajah

Sistem juga dapat mendeteksi wajah pada citra yang berisi banyak wajah banyak.



Gambar 8 Deteksi wajah dan jumlah wajah



Gambar 9 Deteksi wajah dan jumlah wajah

Namun pada gambar 8 dan gambar 9 output yang dihasilkan menampilkan kesalahan, dimana ada beberapa wajah pada citra yang tidak dapat dideteksi. Berdasarkan hasil uji coba yang telah dilakukan menunjukkan bahwa metode Cascade Classifier masih belum dapat mendeteksi wajah pada citra pada gambar yang terlalu jauh. Pada Tabel 1 merupakan detail hasil pengujian dari beberapa citra inputan dengan variasi jumlah objek wajah dan posisi wajah.

Tabel 4.1 Data Pengujian Deteksi Wajah

Citra ke -	Jumlah Wajah Pada Gambar	Jumlah Wajah Terdeteksi	Rata-Rata Deteksi Wajah(%)
1	2	2	100
2	2	2	100
3	1	1	100
4	1	1	100
5	5	5	100
6	5	5	100
7	5	4	80
8	5	4	80
9	3	3	100
10	3	2	66.67

11	2	2	100
12	3	1	33.3
13	6	4	66.67
14	5	3	60
15	7	3	43
16	7	5	71.42
17	7	7	100
18	7	2	28.57
19	6	6	100
20	6	6	100
21	6	5	80.33
22	6	3	50
23	4	4	100
24	4	4	100
25	2	1	50

Hasil pengujian menunjukkan bahwa deteksi wajah mencapai tingkat keberhasilan 100% ketika wajah berada dalam posisi frontal dan dekat. Akurasi metode ini dinilai berdasarkan posisi wajah dalam citra input. Oleh karena itu, citra input yang ingin dideteksi harus menunjukkan fitur wajah yang berada dalam posisi frontal dan dekat. Analisis citra dengan posisi wajah yang tidak frontal dan jauh akan menghasilkan tingkat klasifikasi yang sangat rendah, bahkan dapat menyebabkan ketidakmampuan mendeteksi wajah sama sekali.

KESIMPULAN

Dalam penelitian ini, implementasi Cascade Classifier berbasis *haar cascade classifier* dengan menggunakan OpenCV dan Python berhasil melakukan deteksi wajah pada citra digital. Sistem menunjukkan kehandalan dalam mendeteksi wajah dengan akurasi yang cukup tinggi jika wajah berada pada posisi frontal dan dekat dan waktu eksekusi yang relatif cepat, memenuhi persyaratan untuk aplikasi real-time. Namun, penelitian ini juga mengidentifikasi keterbatasan metode Cascade Classifier, terutama dalam mendeteksi wajah dengan pose atau terlalu jauh. Untuk meningkatkan akurasi, penelitian selanjutnya dapat mempertimbangkan pelatihan menggunakan dataset yang lebih representatif, khususnya mencakup variasi pose dan ekspresi wajah

DAFTAR PUSTAKA

- Boundless. (2016, May 26). "*Politics*.". Retrieved Juny 2016, 01, from Boundless Sociology: <https://www.boundless.com/sociology/textbooks/boundless-sociology-textbook/stratification-inequality-and-social-class-in-the-u-s-9/the-impacts-of-social-class-77/politics-460-4972/>
- Miles, M. B., & Huberman, A. Michael. (1992). *Qualitative Data Analysis, ter: Tjetjep Rohendi Rohendi dengan Judul: Analisis Data Kualitatif*. Jakarta: UI Press.
- Samovar, L., Porter, R., R.Mc Daniel, E., & Roy, C. (2013). *Communication Between Cultures.Eighth Edition*. Wadsworth: Cengage Learning.
- Würtz, E. (2005). Intercultural Communication on Web sites: A Cross-Cultural Analysis of Web sites from High-Context Cultures and Low-Context Cultures. *Journal of Computer-Mediated Communication*, 11: 274–299.