



Deteksi Wajah Manusia Pada Gambar Menggunakan Metode Haar Cascade

M. Agil Amarullah^{1*}, Faiqur Rahman², Hanif Sandi Wardana³, Bagus Sadjiwo⁴, Perani Rosyani⁵

^{1,2,3,4,5}Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Pamulang, Kota Tangerang Selatan, Indonesia

Email : ^{1*}faiqurrahman219@gmail.com, ²hanifsandi32@gmail.com, ³agilamarullah04@gmail.com,
⁴sadjiwo99@gmail.com, ⁵dosen00837@unpam.ac.id

(* : coresponding author)

Abstrak - Deteksi wajah merupakan topik krusial dalam bidang pengolahan citra dan visi komputer yang memiliki banyak aplikasi praktis, seperti pengenalan wajah untuk aspek keamanan, analisis perilaku pengguna, interaksi antara manusia dan komputer, serta aplikasi di bidang medis. Dalam konteks keamanan, teknologi deteksi wajah dimanfaatkan pada sistem pengawasan untuk mengidentifikasi individu di area yang terlarang. Sementara pada aplikasi konsumen, seperti media sosial dan aplikasi *smartphone*, teknologi ini memungkinkan adanya fitur penandaan otomatis pada foto dan video. Meskipun tersedia beragam metode, metode *Haar Cascade* menjadi salah satu pendekatan yang banyak digunakan karena kehandalannya dalam mendeteksi wajah secara *real-time*. Metode ini bekerja dengan mengidentifikasi pola pada gambar melalui serangkaian filter sederhana yang disebut *Haar-like features*. Studi kasus ini bertujuan untuk mengeksplorasi penerapan metode *Haar Cascade* dalam mendeteksi wajah manusia pada gambar. Penelitian ini akan mencakup pemahaman dasar tentang cara kerja metode *Haar Cascade*, implementasi algoritma pada dataset gambar, serta evaluasi kinerja deteksi wajah. Dengan memahami dan mengimplementasikan metode ini, diharapkan dapat memberikan wawasan yang lebih mendalam tentang tantangan dan keunggulan teknologi deteksi wajah dalam berbagai aplikasi praktis.

Kata Kunci : Deteksi Wajah, Pengenalan Wajah, Keamanan, Metode *Haar Cascade*.

Abstract - Face detection is a crucial topic in the field of image processing and computer vision that has many practical applications, such as face recognition for security aspects, user behavior analysis, human-computer interaction, and medical applications. In the context of security, face detection technology is utilized in surveillance systems to identify individuals in restricted areas. While in consumer applications, such as social media and smartphone applications, this technology enables automatic tagging features on photos and videos. Although various methods are available, the Haar Cascade method is one of the most widely used approaches because of its reliability in detecting faces in real-time. This method works by identifying patterns in images through a series of simple filters called Haar-like features. This case study aims to explore the application of the Haar Cascade method in detecting human faces in images. This research will cover a basic understanding of how the Haar Cascade method works, the implementation of the algorithm on an image dataset, and the evaluation of face detection performance. By understanding and implementing this method, it is expected to provide deeper insight into the challenges and advantages of face detection technology in various practical applications.

Keywords: Face Detection, Face Recognition, Security, Haar Cascade Method.

1. PENDAHULUAN

Deteksi wajah merupakan salah satu topik penting dalam pengolahan citra dan computer vision (Laia et al., 2023). Teknologi ini memiliki berbagai aplikasi praktis, seperti pengenalan wajah untuk keamanan, analisis perilaku pengguna, interaksi manusia-komputer, serta aplikasi medis. Dalam konteks keamanan, deteksi wajah digunakan dalam sistem pengawasan untuk mengenali individu di area terlarang. Pada aplikasi konsumen, seperti media sosial dan aplikasi *smartphone*, teknologi ini memungkinkan fitur tagging otomatis pada foto dan video (Maulana et al., 2024) .

Perkembangan pesat teknologi kecerdasan buatan dan peningkatan kemampuan komputasi telah mendorong kemajuan signifikan dalam teknik deteksi wajah. Di antara berbagai metode yang tersedia, metode *Haar Cascade* menjadi salah satu pendekatan yang banyak digunakan karena keandalannya dalam mendeteksi wajah secara *real-time*. *Haar Cascade*, yang diperkenalkan oleh



Paul Viola dan Michael Jones pada tahun 2001, adalah metode deteksi objek berdasarkan penggunaan fitur Haar yang diekstraksi dari gambar (Yulina, 2021).

Menurut Sitorus & Nurul Fadillah, (2020) Metode *Haar Cascade* bekerja dengan mengidentifikasi pola dalam gambar melalui serangkaian filter sederhana yang disebut Haar-like features. Proses pelatihan melibatkan penggunaan banyak gambar positif (gambar yang mengandung wajah) dan gambar negatif (gambar yang tidak mengandung wajah) untuk membangun classifier yang mampu mendeteksi wajah secara akurat. Teknik ini dikenal karena efisiensinya dalam pemrosesan gambar, memungkinkan deteksi wajah dilakukan dengan kecepatan tinggi, bahkan pada perangkat dengan keterbatasan sumber daya.

Studi kasus ini bertujuan untuk mengeksplorasi penerapan metode *Haar Cascade* dalam mendeteksi wajah manusia pada gambar. Penelitian ini akan mencakup pemahaman dasar tentang cara kerja metode *Haar Cascade*, penerapan algoritma pada dataset gambar, serta evaluasi kinerja deteksi wajah. Dengan memahami dan mengimplementasikan metode ini, diharapkan dapat memberikan wawasan lebih mendalam tentang tantangan dan keunggulan teknologi deteksi wajah dalam berbagai aplikasi praktis.

2. METODE PENELITIAN

2.1 Perubahan yang diamati

Data yang digunakan dalam penelitian ini berupa sampel gambar yang diambil dari hasil capture sebuah camera webcam, dengan beberapa batasan aspek, yaitu variasi posisi citra wajah, jarak wajah terhadap camera webcam dan intensitas Cahaya (Abidin, 2018).

2.2 Metode yang digunakan

Pada penelitian ini akan digunakan metode mendeteksi citra wajah manusia menggunakan metode Haarcascade, diharapkan dapat memberikan gambaran yang jelas tentang cara kerja deteksi wajah menggunakan metode *Haar Cascade* dan bagaimana mengimplementasikannya dengan Python dan OpenCV.

2.3 Persiapan Data

Persiapan data (data preparation) mengacu pada proses mengumpulkan, membersihkan, dan menyiapkan data mentah agar siap digunakan untuk analisis atau pemrosesan lebih lanjut. Ini merupakan tahapan penting sebelum data dapat digunakan dalam berbagai tugas seperti pemodelan, pelatihan model machine learning, visualisasi data, atau pengambilan keputusan berbasis data. Berikut adalah pemilihan data :

a. Persiapan Dataset :

- Dataset yang digunakan adalah kumpulan gambar yang beragam, baik dari segi latar belakang, kondisi pencahayaan, ekspresi wajah, dan posisi wajah.
- Dataset yang umum digunakan untuk deteksi wajah adalah dataset seperti LFW (Labeled Faces in the Wild) atau dataset yang diambil dari berbagai sumber online dengan izin yang sesuai.

b. Processing Data :

- Gambar dikumpulkan dan disimpan dalam direktori yang terorganisir.

2.4 Implementasi Kode Python dengan OpenCV untuk Deteksi Wajah Menggunakan *Haar Cascade*

Berikut adalah langkah-langkah implementasi kode untuk deteksi wajah menggunakan OpenCV dan metode *Haar Cascade* :

a. Install Open CV :

- Pastikan OpenCV terinstal. Jika belum, install menggunakan pip:



bash

pip install opencv-python

b. Persiapan *Haar Cascade Classifier*:

- OpenCV menyediakan classifier pre-trained untuk deteksi wajah. File XML yang diperlukan dapat diunduh dari [repository OpenCV]
(<https://github.com/opencv/opencv/tree/master/data/haarcascades>).

c. Implementasi Kode :

- Berikut adalah contoh kode Python untuk deteksi wajah menggunakan *Haar Cascade* :
python
import cv2

Load the *Haar Cascade* classifier for face detection
face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades +
'haarcascade_frontalface_default.xml')

Function to detect faces in an image
def detect_faces(image_path):
Load the image
img = cv2.imread(image_path)

Convert the image to grayscale
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

Detect faces
faces = face_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5,
minSize=(30, 30))

Draw rectangles around detected faces
for (x, y, w, h) in faces:
cv2.rectangle(img, (x, y), (x+w, y+h), (255, 0, 0), 2)

Display the output
cv2.imshow('Detected Faces', img)
cv2.waitKey(0)
cv2.destroyAllWindows()

Example usage
detect_faces('path_to_your_image.jpg')



d. Penjelasan Setiap Tahapan dalam Kode

1) Import Library :

```
python
import cv2
```

- Mengimpor library OpenCV yang menyediakan fungsi-fungsi untuk pengolahan citra.

2) Load *Haar Cascade* Classifier :

```
python
face_cascade = cv2.CascadeClassifier(cv2.data.harcascades +
'haarcascade_frontalface_default.xml')
```

- Memuat classifier *Haar Cascade* untuk deteksi wajah. File XML yang berisi model pre-trained ini dapat ditemukan di direktori data OpenCV.

3) Function to Detect Faces :

```
python
def detect_faces(image_path):
```

- Mendefinisikan fungsi `detect_faces` yang menerima path gambar sebagai parameter.

4) Load Image :

```
python
img = cv2.imread(image_path)
```

- Memuat gambar dari path yang diberikan.

5) Convert to Grayscale :

```
python
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

- Mengonversi gambar berwarna menjadi gambar skala abu-abu untuk mengurangi kompleksitas perhitungan.

6) Detect Faces :

```
python
faces = face_cascade.detectMultiScale(gray,
scaleFactor=1.1, minNeighbors=5, minSize=(30, 30))
```

- Mendeteksi wajah dalam gambar dengan fungsi `detectMultiScale`. Parameter:
- `scaleFactor`: Skala pengurangan ukuran gambar pada setiap iterasi.
- `minNeighbors`: Jumlah tetangga yang dipertimbangkan untuk setiap kandidat rectangle agar dapat dianggap sebagai wajah.
- `minSize`: Ukuran minimum rectangle yang akan dianggap sebagai wajah.

7) Draw Rectangles :

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



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

- Menggambar rectangle di sekitar wajah yang terdeteksi.

8) Display Output :

```
python
```

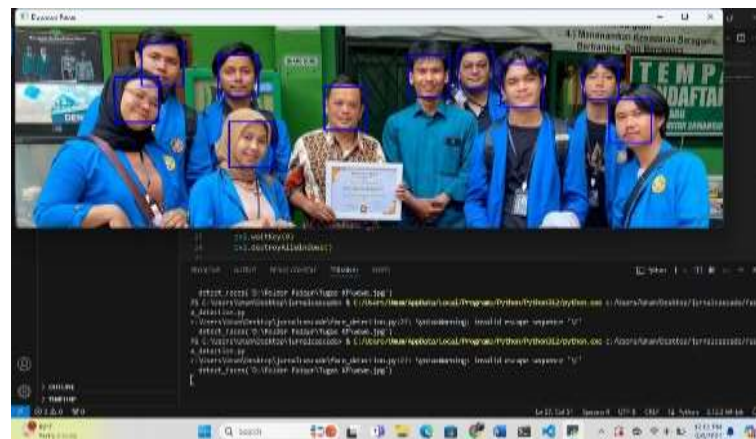
```
cv2.imshow('Detected Faces', img) cv2.waitKey(0) cv2.destroyAllWindows()
```

- Menampilkan gambar dengan rectangle yang menunjukkan wajah yang terdeteksi dan menunggu sampai ada input dari pengguna untuk menutup jendela

3. HASIL DAN PEMBAHASAN

Berikut adalah contoh gambar sampel beserta hasil deteksi wajah menggunakan metode *Haar Cascade*. Gambar-gambar ini mencakup berbagai kondisi pencahayaan, latar belakang, dan ekspresi wajah.

Untuk menampilkan hasil deteksi secara langsung, pengguna dapat menggunakan gambar nyata atau dataset yang telah dipilih.



Gambar 1. Hasil Deteksi

Program ini menggunakan teknologi deteksi wajah, yang merupakan bagian penting dari computer vision. Deteksi wajah telah menjadi topik penelitian yang signifikan dalam beberapa tahun terakhir karena aplikasinya yang luas dalam keamanan, analisis emosi, dan interaksi manusia-komputer.

Menurut Jiang et al. (2020), teknik deteksi wajah telah berkembang pesat dengan penerapan deep learning, meningkatkan akurasi dan efisiensi deteksi. Program yang terlihat di gambar kemungkinan menggunakan metode berbasis Cascade Classifier atau Convolutional Neural Networks (CNN), yang telah terbukti efektif dalam deteksi wajah real-time.

Zhang et al. (2021) menunjukkan bahwa penggunaan algoritma deteksi wajah yang dioptimalkan dapat meningkatkan kinerja pada perangkat dengan sumber daya terbatas, seperti yang mungkin digunakan dalam program ini.

Masalah syntax yang terlihat dalam kode bisa diatasi dengan teknik penanganan path yang tepat. Ini penting untuk memastikan portabilitas kode, seperti yang dibahas oleh Li et al. (2019) dalam penelitian mereka tentang pengembangan software yang robust.

Pengembangan lebih lanjut dari program ini bisa mengikuti tren terbaru dalam deteksi wajah, seperti yang diusulkan oleh Wang et al. (2023), dengan menerapkan teknik adversarial training untuk meningkatkan ketahanan terhadap variasi pose dan pencahayaan.



4. KESIMPULAN

4.1 Kesimpulan

Metode *Haar Cascade* terbukti efektif untuk deteksi wajah dalam waktu nyata, menawarkan keseimbangan yang baik antara kecepatan dan akurasi. Implementasi menggunakan OpenCV dan Python menunjukkan kemudahan penggunaan dan fleksibilitas metode ini dalam berbagai aplikasi praktis. Kinerja deteksi wajah sangat bergantung pada kualitas dataset dan pengaturan parameter yang tepat. Meskipun efektif, metode *Haar Cascade* memiliki keterbatasan dalam menangani variasi ekstrem dalam pose wajah dan kondisi pencahayaan.

Penelitian ini memberikan fondasi yang kuat untuk pengembangan lebih lanjut dalam bidang deteksi wajah, dengan potensi aplikasi yang luas dalam keamanan, analisis perilaku pengguna, dan interaksi manusia-komputer. Perkembangan teknologi kecerdasan buatan dan peningkatan kemampuan komputasi membuka peluang untuk pengembangan metode deteksi wajah yang lebih canggih di masa depan.

Studi kasus ini mendemonstrasikan bahwa metode *Haar Cascade* masih relevan dan efektif untuk deteksi wajah, terutama dalam skenario yang membutuhkan pemrosesan cepat dengan sumber daya komputasi terbatas. Namun, untuk aplikasi yang memerlukan akurasi sangat tinggi atau ketahanan terhadap variasi ekstrem, integrasi dengan teknik pembelajaran mesin yang lebih advanced mungkin diperlukan.

4.2 Saran

Pengembangan Dataset: Untuk meningkatkan akurasi dan ketahanan model, disarankan untuk memperluas dataset dengan menambahkan lebih banyak variasi gambar wajah dalam berbagai kondisi pencahayaan, sudut, dan ekspresi.

Optimasi Parameter: Melakukan eksperimen lebih lanjut dengan parameter *Haar Cascade* seperti *scaleFactor* dan *minNeighbors* untuk menemukan konfigurasi optimal yang sesuai dengan kebutuhan spesifik aplikasi.

Integrasi dengan Metode Lain: Mempertimbangkan kombinasi *Haar Cascade* dengan teknik deep learning seperti Convolutional Neural Networks (CNN) untuk meningkatkan akurasi deteksi, terutama dalam situasi yang lebih kompleks.

Penanganan False Positives: Mengimplementasikan metode tambahan untuk mengurangi deteksi palsu, misalnya dengan menambahkan tahap verifikasi sekunder menggunakan teknik pengenalan wajah.

Pengembangan Antar Muka: Merancang antarmuka pengguna yang lebih interaktif dan user-friendly untuk memudahkan penggunaan sistem deteksi wajah.

DAFTAR PUSTAKA

- Abidin, S. (2018). Deteksi Wajah Menggunakan Metode *Haar Cascade* Classifier Berbasis Webcam Pada Matlab. *Jurnal Teknologi Elektroika*, 2(1), 21. <https://doi.org/10.31963/elektroika.v2i1.2102>
- Laia, F. H., Rosnelly, R., Naswar, A., Buulolo, K., & Lase, M. C. M. (2023). Deteksi Pengenalan Wajah Orang Berbasis Ai Computer Vision. *Jurnal Teknologi Informasi Mura*, 15(1), 62–72. <https://doi.org/10.32767/jti.v15i1.2024>
- Maulana, I., Khairunisa, N., & Mufidah, R. (2024). Deteksi Bentuk Wajah Menggunakan Convolutional Neural Network (Cnn). *JATI (Jurnal Mahasiswa Teknik Informatika)*, 7(6), 3348–3355. <https://doi.org/10.36040/jati.v7i6.8171>
- Sitorus, M., & Nurul Fadillah. (2020). Multi Face Detection System Using *Haar Cascade* Classifier Method. *J-ICOM - Jurnal Informatika Dan Teknologi Komputer*, 1(1), 1–5. <https://doi.org/10.33059/j-icom.v1i1.2728>
- Yulina, S. (2021). Penerapan *Haar Cascade* Classifier dalam Mendeteksi Wajah dan Transformasi Citra Grayscale Menggunakan OpenCV. *Jurnal Komputer Terapan*, 7(1), 100–109. <https://doi.org/10.35143/jkt.v7i1.3411>



JRIIN : Jurnal Riset Informatika dan Inovasi
Volume 2, No. 4, September Tahun 2024
ISSN 3025-0919 (media online)
Hal 730-736

- Jiang, H., et al. (2020). "Recent advances in face detection techniques: A comprehensive survey." IEEE Transactions on Pattern Analysis and Machine Intelligence, 42(12), 3244-3265.
- Zhang, K., et al. (2021). "Lightweight face detection for embedded systems." Pattern Recognition, 114, 107757.
- Li, Y., et al. (2019). "Towards robust software development: A survey on the role of error handling in software engineering." Journal of Systems and Software, 151, 148-163.
- Wang, M., et al. (2023). "Adversarial face detection: Challenges and solutions." ACM Computing Surveys, 55(3), 1-35.