Pengukuran Jarak Kendaraan Dengan Metode Haar Cascade

Measurement Of Vehicle Distance With Haar Cascade Method Using Opencv

1st Axel Davlin Yusyahnur
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
axeldav@student.telkomuniversity.ac.id

2nd Agus Virgon
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
avirgono@telkomuniversity.ac.id

3rd Umar Ali Ahmad
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
umar@telkomuniversity.ac.id

Abstrak

Di zaman sekarang kecelakaan antar kendaraan sering terjadi dimana-mana, terutama di pusat kota, itu karena kelalaian pengendara. Dalam penelitian ini akan dirancang pengukur jarak kendaraan menggunakan OpenCV dengan metode Haar Cascade.Metode Haar Cascade yaitu salah satu machine learning yang biasa digunakan sebagai aplikasi object detection (diutamakan face recognition). Sebelum melakukan metode tersebut, harus adanya proses identifikasi pengenalan objek yang dijadikan target. Proses identifikasi obyek tersebut menggunakan algoritma Haar Cascade itu sendiri, algoritma tersebut biasa dipakai untuk mengenali jarak objek maupun deteksi obyek. Kemudian Haar Cascade ini menerapkan cascade function yang dimana ada 4 proses : (1) Menentukan Haar Features, (2) Membuat gambar integral, (3) Adaboost Training, dan (4) Melakukan klasifikasi cascading classifier. Sistem perhitungan jarak obyek dengan kamera yang telah dibuat bisa mendeteksi jarak obyek dengan tingkat kesalahan yang akan ditentukan tergantung dari resolusi video dan fps (frame rate) meskipun dengan dalam rentang jarak yang terbatas.

Kata kunci : Computer Vision, Haar Cascade, Jarak Objek.

Abstract

In today's era, accidents between vehicles often occur everywhere, especially in the city center, it is due to the negligence of the driver. In this study, a vehicle distance meter will be designed using OpenCV with the Haar Cascade method. The Haar Cascade method is one of machine learning that is commonly used as an object detection application (face recognition is preferred). Before carrying out this method, there must be a process of identifying the identification of the object being targeted. The object identification process uses the Haar Cascade algorithm itself, the algorithm is commonly used to recognize object distances and object detection. Then Haar Cascade implements a cascade function in which there are 4 processes: (1) Determining Haar Features, (2) Creating integral images, (3) Adaboost Training, and (4) Performing cascading classifier classification. The object distance calculation system with the camera that has been made can detect the object distance with an error rate that will be determined depending on the video resolution and fps (frame rate) even though it is within a limited range.

Keywords: Computer Vision, Haar Cascade, Object Distance.

1. Pendahuluan

Kendaraan sebagai fasilitas pendukung kehidupan manusia yang tidak dipisahkan dari aspek- aspek kehidupan manusia contohnya mobil. Kendaraan di zaman saat ini sangat padat sekali dan kecelakaan pun tidak terhindarkan. Mengapa hal tersebut bisa terjadi, karena kelalaian pengendara yang tidak bisa memperkirakan jarak antar kendaraan di depannya. Dengan memanfaatkan teknologi maju saat ini, dirancang lah sistem pengukur jarak kendaraan menggunakan OpenCV. Yang dimana sistem ini mampu mengurangi kelalaian pengendara pemula untuk memperkirakan jarak kendaraan agar terhindar dari benturan atau gesekan antar kendaraan. Jika dilihat keadaan sekarang masih banyak masyarakat yang lalai berkendara, yang mengakibatkan banyak kecelakaan ringan sampai kecelakaan berat yang bisa merenggut korban jiwa. Sejauh ini Kepolisian RI mencatat jumlah insiden kecelakaan lalu lintas meningkat hingga 100 persen di tahun 2021, ada sebanyak 1.291 kasus atau naik 100 persen dari tahun lalu hanya 566 kejadian. Maka dari itu diusulkan sistem pengukur jarak kendaraan menggunakan kamera, yang diharapkan bagi pengendara pemula agar bisa berhati- hati dengan menyesuaikan jarak antar kendaraan agar tidak terjadi hal yang tidak diinginkan.

Sistem pengukuran jarak kendaraan menggunakan OpenCV ini merupakan sistem untuk membandingkan jarak kendaraan dengan kendaraan yang ada di depannya. Pada sistem ini juga pasti tidak jauh dari kekurangan yaitu hanya bisa menentukan jarak saja dan tidak bisa memberikan peringatan yang lain. Maka perlu adanya perkembangan lebih lanjut untuk ke depannya.

Sistem pengukuran jarak kendaraan ini juga dikembangkan dari penelitian sebelumnya yang dimana penelitian sebelumnya membahas tentang pendeteksi jarak roda kendaraan. Perbedaaanya yaitu pada implementasi dan tentu saja pada hasil. Jika dipenelitian sebelumnya yaitu mendeteksi jarak roda kendaraan, sedangkan penelitian kali ini yaitu tentang pengukuran jarak antar kendaraan. Sistem ini pasti sudah banyak diteliti. Namun belum ada yang meneliti olah video jarak antar kendaraan.

2. Ladasan Teori

2.1 Citra

Citra atau *image* merupakan representasi dua dimensi yang diciptakan dari semacam gambar atau pandangan. Pengertian lainnya ialah gambar dua dimensi yang dihasilkan dari gambar analog dua dimensi menjadi gambar diskrit dengan adanya proses *sampling*. Citra digital bisa dikatakan fungsi dua variabel f(x,y), yang dimana x dan y itu ialah koordinat spasial dan sedangkan nilai f(x,y) ialah intensitas citra pada koordinat itu. Pada dasar nya untuk menampilkan dan menciptakan warna pada citra digital merupakan kombinasi dari 3 warna dasar yaitu merah, biru, dan hijau. Hal tersebut diilustrasikan seperti pada gambar.

Citra digital ialah citra f(x,y) yang didigitalisasi baik koordinat area atau *brightness* level. Untuk menunjukkan *brightness* (kecerahan) ada pada nilai f di koordinat (x,y) atau *grayness* level dari citra pada titik tersebut. Piksel adalah satuan atau bagian terkecil dari citra digital (piksel atau *picture element*). Pada umumnya citra itu dibentuk dari kotak-kotak piksel yang teratur, jadi jarak horizontal dan vertical pada seluruh bagian citra itu sama. Setiap piksel diwakili oleh dua buah bilangan bulat integer untuk menunjukkan lokasi suatu piksel. Koordinat (0,0) digunakan pada posisi kiri atas pada bidang citra dan koordinat (m-1, n-1) digunakan pada posisi kanan bawah dalam citra berukuran m x n piksel. Pada gambar ini ialah persamaan koordinat piksel (x,y) pada citra berukuran m x n[1][2].

2.2 Citra RGB

Merah, Hijau, dan Biru adalah warna

dasar yang bisa diterima mata manusia. Setiap piksel pada citra mewakili warna yang dimana hal tersebut kombinasi dari ketiga warna terebut. Setiap titik citra membutuhkan data 3 byte. Pada tiap warna dasar RGB memiliki intensitas sendiri, nilai minimum nol (0) dan nilai maksimum 255 (8bit). Berdasarkan teori mata manusia peka sekali terhadap panjang gelombang 630nm (*red*), 530nm (*green*), dan 450nm (*blue*).

2.3 Citra Biner

Citra Biner (binary image) merupakan citra digital yang kemungkinan warna, yaitu hitam dan putih. Citra biner biasa disebut citra white dan black (W&B) atau monokrom. Nilai yang mewakili tiap piksel dari citra biner hanya membutuhkan 1 bit[3].

Persamaan binerisasi :

$$f(x,y)^{'}=\begin{cases}a_1, & f(x,y)<T \\ a_2, & f(x,y)\geq T\end{cases}$$

2.4 Object Tracking

Object tracking sangat penting dalam computer vision, dengan hadir nya komputer yang memiliki kecanggihan masing-masing dan semakin berkembang, tersedia nya kamera video dengan kualitas yang tidak diragukan. Penggunaan sistem object tracking sering dimanfaatkan untuk yang berkaitan dengan sistem keamanan, pemantau lalu lintas, sistem navigasi, dan interaksi manusia-komputer.

Object tracking merupakan proses pengindentifikasi objek atau beberapa objek pada sebuah citra secara otomatis. Untuk menandakan indentifikasi objek biasanya dengan memberikan tanda pada sekeliling objek tersebut bahwa objek tersebut ada pada proses tracking. Kasus object tracking juga digunakan dalam beberapa kasus dan tergantung inputan nya bisa berupa gambar, rekaman video dan juga rekaman video secara realtime[4].

2.5 OpenCV

Open CV (Open Source Computer Vision) adalah library dari suatu fungsi pemrograman untuk realtime visi komputer. OpenCV menggunakan lisensi BSD dan bersifat gratis baik untuk penggunaan komersial ataupun akademis. OpenCV bisa digunakan dalam bahasa C, C++, Python, Java, dan sebagainya. OpenCV juga dapat digunakan pada OS Windows, Linux, IOS, Android, dan Mac OS. OpenCV pun ada lebih dari 2500 algoritma yang sudah dioptimalkan[5][6]. Kinerja fungsi OpenCV untuk aliran optik padat dipilih sebagai baseline, dan nilai optimal untuk parameter dipilih melalui pencarian grid pada nilai kandidat. Model yang diusulkan dalam pekerjaan saat ini mengungguli metode dasar pada metrik kehilangan fotometrik[5].

2.6 Haar Cascade

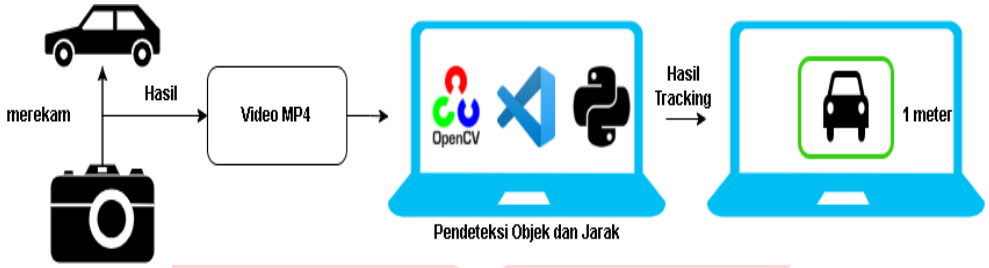
Haar Cascade yaitu salah satu machine learning yang biasa digunakan sebagai aplikasi object detection (diutamakan face recognition). Sebelum melakukan metode tersebut, harus adanya proses identifikasi pengenalan objek yang dijadikan target. Proses identifikasi objek tersebut menggunakan algoritma Haar Cascade itu sendiri, algoritma tersebut biasa dipakai untuk mengenali jarak objek maupun deteksi objek. Kemudian Haar Cascade ini menerapkan cascade function yang dimana ada 4 proses : (1) Menentukan Haar Features, (2) Membuat gambar integral, (3) Adaboost Training, dan (4) Melakukan klasifikasi cascading classifier[6][7][8].

2.7 Python

Python adalah bahasa pemrograman yang mengeksekusi setiap intruksi multi guna secara langsung (interpretatif) dengan orientasi objek menggunakan semantik dinamis yang memberikan tingkat keterbatasan sintak. Python juga menggabungkan kapabilitas dan sintaksis kode yang sangat jelas, kemudian dirancang untuk dapat dipahami dan dipelajari. Python pada penelitian ini menggunakan python dari extension Visual Studio Code[9].

3. Perancangan Sistem

3.1 Desain Sistem

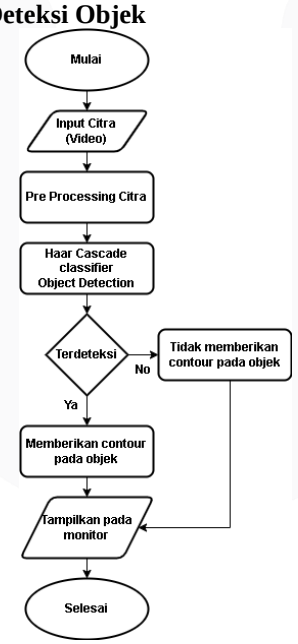


Gambar 5: Desain Sistem

Pada desain sistem ini merupakan alat berupa kamera GoPro dipasangkan pada kendaraan yang dikendarai pengguna untuk menghadap kepada sebuah kendaraan atau mobil yang ada didepannya. Lalu merekam kendaraan atau mobil yang ada didepannya dengan kondisi bergerak dan diam tersebut dengan jarak yang berbeda, jarak ditentukan dibawah 3 meter dengan waktu durasi video sekitar 1 menit. Pengujian merekam kendaraan atau mobil dilakukan pada sore hari dan malam hari. Kemudian hasil video tersebut diolah pada pemrograman Python dan OpenCV untuk bisa mendeteksi kendaraan atau mobil dan menentukan jarak kendaraan atau mobil pada video tersebut.

3.2 Flowchart

3.2.1 Flowchart Deteksi Objek



Gambar 6: Flowchart Deteksi Objek

Pada gambar 6 merupakan proses pendeteksian suatu objek bagaimana sistem mengenali objek. Dimana kamera akan menangkap objek tersebut sebagai mobil, kemudian objek tersebut dideteksi oleh sistem yang dibuat dengan program python. Objek tersebut terdeteksi ditandai dengan memberikan contour pada sekeliling objek berwarna hijau. Kemudian hasil ditampilkan pada monitor.

3.2.2 Flowchart Pendeteksian Objek Jarak DIbawah 1 Meter’

3.2.4 Flowchart Perhitungan Jarak

Gambar 9: Flowchart Perhitungan Jarak

Pada proses ini merupakan perhitungan nilai jarak pemrograman python, berikut proses nya :

1. Tentukan nilai Known_Distance dan Known_Width
   Nilai Known_Distance diketahui perkiraan jarak dari kamera menuju objek. Kemudian nilai Known_Width diketahui dari lebar nya objek.
   Nilai knownDistance = 300 cm * 0.01 (agar satuan menjadi meter)
   Nilai knownWidth = 1,8 meter (diketahui lebar mobil rata-rata sekitar 1,8 meter)
2. Kalibrasi Video
   Pemrograman memuat hasil video dengan format mp4 untuk dikalibrasi dengan perhitungan heightImg dan widthImg. Kalibrasi untuk resize output video :
   heightImg = 500
   widthImg = 500
   frame = cv2.resize(frame, (WidthImg, HeightImg)
3. Menghitung Focal Length
   Cara menghitung nilai focal length dengan cara :
   focallength = (marker[1][0] * knownDistance) / knownWidth
   *marker dimaksudnya dari nilai penanda garis tengah video
4. Perhitungan Jarak Objek Dengan Kamera
   Proses pemrograman perhitungan nilai jarak :
   distance_to_camera (knownWidth * focalLength) / perWidth
   Fungsi ini mengambil knownWidth dari penanda, yang dihitung focalLength, dan lebar yang dirasakan dari suatu objek dalam gambar (diukur dalam piksel), dan menerapkan kesamaan segitiga yang dirinci di atas untuk menghitung jarak sebenarnya ke objek. Dan untuk perhitungan nilai jarak yang terlalu dekat hanya dibatasi nilai m < 1 meter.

3.3 Analisis Kebutuhan Sistem

Tabel 1: Spesifikasi Perangkat Keras

No	Perangkat Keras	Spesifikasi
1	Laptop Acer Aspire 3	-AMD Ryzen 5 2500U -AMD Radeon Vega Mobile Graphics -12 GB DDR4 Memory -1000 GB HDD -Layar 1366 x 768 (32bit) (60hz)
2	Kamera GoPro Hero 5 Black	-Photo : 12 MP / 30 Fps Burst Time Lapse 12 MP / Rafale 30 IPS / Accelere -Video : 4K30 / 1440p80 / 1080p120 -Waterproof ETANCHE -Simple 1 button control -Wifi + Bluetooth -Advance Wind Noise Reduction -Voice Control -Video Stabilization -Touch Display -Stereo Audio + Mic Input -Location Capture -Raw + WDR Photos
3	Memory Card	-VGeN 16 GB

Tabel 2: Spesifikasi Perangkat Lunak

No	Perangkat Lunak	Versi
1	OpenCV	0.1.2
2	Windows 10	21H1
3	Python	3.10

4. Hasil dan Analisis

4.1 Tahapan Pengujian Deteksi Objek

Pada tahapan ini pengujian dilakukan setelah mendapat hasil rekaman dengan format mp4 dari kamera GoPro yang digunakan, setelah itu eksekusi pada program python. Hasil deteksi objek dilihat dengan adanya *countour* berwarna hijau disekeliling objek.

Gambar 10: Deteksi Objek Kendaraan 1

Gambar 11: Deteksi Objek Kendaraan 2

Gambar 12: Deteksi Objek Kendaraan 3

Tabel 3: Analisis Pengujian Deteksi Objek

No.	Objek	Hasil
1	Kendaraan 1	Terdeteksi
2	Kendaraan 2	Terdeteksi
3	Kendaraan 3	Terdeteksi

Keterangan :
Pada pengujian Deteksi Objek dari ketiga pengujian semua dapat terdeteksi dengan ditandai contour berwarna hijau.

4.2 Tahapan Pengujian Deteksi Jarak Objek

Pada tahapan ini pengujian dilakukan ketika objek yang dideteksi diam, saat eksekusi program jarak yang dihasilkan dapat diperoleh dan dengan akurasi yang mendekati nilai jarak sebenarnya, akan tetapi pendeteksian jarak nya menjadi tidak beraturan karena objek dideteksi lebih baik saat objek bergerak. Objek pun dapat dihitung jarak nya ketika objek terdeteksi, apabila objek tidak terdeteksi maka perhitungan jarak pun tidak akan dapat keluar nilai jaraknya. Pada pengukuran jarak kali ini dibatasi jarak kamera dengan objek yaitu 3 meter karena meteran pengukur hanya mencapai 3 meter.

Gambar 13: Meteran Pengukur Kendaraan 1

- Perhitungan manual nilai *error* jarak pengukuran :

  (Jarak Sebenarnya – Jarak Sistem = Nilai *Error*)

  1,83 m – 1,45 m = 0,38 m
- Perhitungan manual nilai persentase error :

  ((Nilai Error / Jarak Sebenarnya) * 100 = Nilai Persentase Error)

  (0,38 / 1,83 m) * 100 = 20,76% error

Jarak kedua yang didapat pada kendaraan 1 adalah 1,45 m dengan titik diam sebenarnya 1,83 m, didapat nilai selisih error 0,38 m maka hasil perhitungan manual persentase error pada sistem 20,76%.

Gambar 17: Meteran Pengukur Kendaraan 2

Gambar 18: Jarak Pertama Kendaraan 2

- Perhitungan manual nilai error jarak pengukuran :

  (Jarak Sebenarnya – Jarak Sistem = Nilai Error)

  2,99 m – 1,94 m = 1,05 m
- Perhitungan manual nilai persentase error :

  ((Nilai Error / Jarak Sebenarnya) * 100 = Nilai Persentase Error)

  (1,05 m / 2,99 m) * 100 = 35,11% error

Jarak pertama yang didapat pada kendaraan 1 adalah 1,94 m dengan titik diam sebenarnya 2,99 m, didapat nilai selisih error 1,05 m maka hasil perhitungan manual persentase error pada sistem 35,11%.

Tabel 4: Analisis Pengujian Deteksi Jarak Objek Diam

No.	Objek	Hasil		Jarak Nyata	Nilai Error	Persentase Error
		Pendeteksian Objek	Jarak Sistem			
1	Kendaraan 1	Terdeteksi	2,79 m	2,97 m	0,18 m	1,06%
2		Terdeteksi	1,45 m	1,83 m	0,38 m	1,26%
3	Kendaraan 2	Terdeteksi	1,94 m	2,99 m	1,05 m	1,54%

Keterangan :

Pengujian deteksi jarak berhasil karena jarak dapat dihitung apabila objek terdeteksi. Jika objek tidak terdeteksi maka jarak sistem tidak dapat dihitung.

5. Kesimpulan dan Saran

5.1 Kesimpulan

Kesimpulan yang dapat diperoleh dari pengujian Tugas Akhir ini ialah :

1. Sistem mampu mendeteksi objek dengan metode Haar Cascade menggunakan fitur Haar-Like Feature, Internal Image, Adaboost, dan Cascade Classifier dengan hasil pengujian deteksi objek tersebut diberi tanda contour warna hijau disekeliling objek untuk menandakan objek terdeteksi kendaraan dan pendeteksian objek kendaraan lebih baik saat objek bergerak daripada objek diam.
2. Sistem mampu menghitung jarak objek dengan kamera saat objek bergerak maupun diam, perhitungan jarak akan didapat apabila objek kendaraan tersebut terdeteksi, sebaliknya saat

objek tidak terdeteksi maka perhitungan jarak nya tidak dapat terhitung, dari 3 kali pengujian (2 kali pengujian pada kendaraan 1 dan 1 kali pengujian pada kendaraan 2) didapat nilai selisih

error pada kendaraan 1 (0,18 m dan 0,38 m) maka didapat nilai persentase error (6,06% dan 20,76%), pada kendaraan 2 didapat nilai selisih error (1,05 m).maka didapat nilai persentase error (35,11%).

3. Sistem mampu mendeteksi objek dengan jarak dibawah 1 meter dengan ditandai contour berwarna merah disekeliling objek, nilai jarak tersebut didapat saat pengujian jarak objek bergerak karena pendeteksian lebih baik saat objek bergerak, kemudian nilai jarak tersebut tidak akurat dikarenakan pengujian dilakukan saat objek bergerak yang dimana tidak bisa menghitung nilai persentase error karena jarak nyata tersebut tidak diketahui.

5.2 Saran

Saran untuk pengembangan Tugas Akhir ini yaitu :

1. Pendeteksian pada objek kendaraan mobil tersebut harus lebih akurat lagi saat objek bergerak atau diam.
2. Perhitungan nilai jarak pada kendaraan mobil tersebut harus lebih akurat lagi disaat objek bergerak atau diam.
3. Pendeteksian jarak objek terlalu dekat dibawah 1 meter harus lebih akurat lagi disaat objek bergerak atau diam.
4. Sistem hanya bisa dijalankan menggunakan pc/laptop dan menggunakan aplikasi visual studio code dengan terinstall python dan library tambahan yaitu OpenCV dan Numpy. Sehingga dibutuhkan pengembangan lebih lagi agar sistem bisa dijalankan secara realtime.
5. Sistem dibuat agar bisa juga akurat pendeteksian nya pada malam hari.

Referensi:

[1] I. N. Rifai, B. Sumanto, P. D. Elektronika, S. Vokasi, and U. G. Mada, “Computer Vision Untuk Penghitungan Jarak Obyek Terhadap,” *Semin. Nas. Teknol. Terap.*, vol. 1, pp. 464–469, 2013.

[2] D. Nugraheny, “Metode Nilai Jarak Guna Kesamaan Atau Kemiripan Ciri Suatu Citra (Kasus Deteksi Awan Cumulonimbus Menggunakan Principal Component Analysis),” *Angkasa J. Ilm. Bid. Teknol.*, vol. 7, no. 2, p. 21, 2017.

[3] I. S. Pratama, Muhammad Rizky; Rizal, Achmad; Sumaryo, “Desain Sistem Deteksi Objek Real Time Dengan Metode Haar Cascade Classifier Real Time Object Detection System Design Using Haar Cascade Classifier Method,” vol. 7, no. 1, pp. 26–34, 2020.

[4] R. T. Yunardi, A. W. Mardhiyah, M. H. Yahya, and C. Satria, “Desain dan Implementasi Visual Object Tracking Menggunakan Pan and Tilt Vision System,” *J. ELKHA*, vol. 11, no. 2, pp. 85–92, 2019.

[5] M. Shafiqul Islam *et al.*, *A Supervised Machine Learning Approach Towards Accelerating the Development of High-Performance Non-Pb Perovskite Solar Devices*. 2021.

[6] L. Wisesa, “OpenCV Face Recognition Berbasis Algoritma Haar Cascade,” 2019.

[7] S. Abidin, “Deteksi Wajah Menggunakan Metode Haar Cascade Classifier Berbasis Webcam Pada Matlab,” *J. Teknol. Elekterika*, vol. 15, no. 1, p. 21, 2018.

[8] A. H. Ginting, “IMPLEMENTASI ALGORITMA HAAR CASCADE CLASSIFIER Universitas Sumatera Utara,” 2016.

[9] M. R. Adani, “Python,” 2020.