

Sistem Penghitung Jumlah Kendaraan dan Pendeteksi Kecepatan pada Ruas Jalan Menggunakan Metode Haar Cascade Classifier

Amar Andra Saifullah, Ega Hegarini dan Bheta Agus Wardijono

SInformatika, Fakultas Teknologi Industri, Universitas Gunadarma
Jl. Margonda Raya No. 100, Depok 16424, Jawa Barat
Sistem Informasi, Fakultas Ilmu Komputer, Universitas Gunadarma
Jl. Margonda Raya No. 100, Depok 16424, Jawa Barat
Sistem Komputer, STMIK Jakarta STI&K
Jl. BRI Radio Dalam No.17, Gandaria Utara, Kby Baru, Jakarta Selatan, Jakarta 12140
E-mail: amarandra14@gmail.com, ega@staff.gunadarma.ac.id, bheta@staff.jak-stik.ac.id*

Abstrak

Kecepatan berkendara pada lalu lintas di jalan raya yang mengakibatkan meningkatnya kewaspadaan akan pergerakan pada ruas jalan dapat menyebabkan sering terjadinya kecelakaan pada jalan tersebut. Kecelakaan banyak terjadi di kota-kota besar yang ada di Indonesia, penyebab terjadinya kecelakaan beragam seperti hilangnya fokus saat berkendara, mengantuk, dan lain sebagainya. Bukan hanya itu, kecelakaan juga terjadi dapat disebabkan oleh jumlah masyarakat atau penduduk dalam suatu yang begitu padat atau kelalaian penduduk setempat seperti menyeberang jalan tanpa menghiraukan kendaraan sekitar. Pada penelitian ini berhasil dibuat program untuk mendeteksi kendaraan dan mengetahui kecepatan kendaraan yang melintas di jalan raya. Metode deteksi yang digunakan adalah haar cascade classifier. Hasil ujicoba terdapat 16 mobil yang berhasil terdeteksi dari total keseluruhan 40 mobil yang melintas di jalan. Mobil-mobil ini memiliki kecepatan yang bervariasi. Pada penelitian ini tingkat keberhasilan dari metode yang diterapkan dalam mendeteksi kendaraan yang lewat adalah 40.0 %.

Kata kunci : Machine Learning, Pendeteksi, Kecepatan, Kendaraan, Haar Cascade Classifier

Pendahuluan

Teori arus lalu lintas adalah suatu kajian tentang gerakan pengemudi dan kendaraan antara dua titik dan interaksi antara keduanya membuat satu sama lain. Namun, mempelajari arus lalu lintas sulit karena perilaku pengemudi adalah sesuatu yang tidak dapat diprediksi dengan pasti.

Lancarnya suatu lalu lintas dipengaruhi oleh beberapa faktor, antara lain luas jalan, jumlah kendaraan dan jenis kendaraan yang lewat. Seiring berjalannya waktu, lalu lintas semakin dipenuhi dengan berbagai jenis kendaraan. Ditambah dengan luas jalan yang sudah tidak dapat menampung jumlah kendaraan yang berlalu lintas dan strategi lalu lintas yang kurang baik membuat arus lalu lintas suatu jalan semakin padat dan kemacetan semakin berbondong-bondong. Hal ini sangat berpengaruh besar di dalam kehidupan sehari-hari manusia, misalkan waktu yang terbuang cukup

banyak. Suatu hal yang dapat dilakukan untuk mengurangi kemacetan adalah mengatur strategi lalu lintas yang baik (Dufan J. P. Manajang, 2020).

Saat ini penghitungan jumlah kendaraan masih dilakukan secara manual dengan mencatat setiap jenis kendaraan yang lewat. Cara lain adalah dengan memasang suatu sensor pada ruas jalan tertentu dimana sensor akan mendeteksi setiap kendaraan yang lewat berdasarkan kuat tekan, infra merah, ultras sonic atau jenis sensor yang lain. Sejalan dengan perkembangan metode kecerdasan buatan, visi komputer dan murahnya perangkat kamera, metode deteksi dan penghitungan jumlah kendaraan juga semakin berkembang. Sensor berbasis penglihatan (vision) dapat memberikan informasi yang lebih banyak daripada sensor lain, sehingga deteksi trafik berbasis kamera memiliki peran penting pada deteksi kendaraan dan pemantauan lalu lintas. Namun, video yang ditangkap oleh kamera membutuhkan pemrosesan yang canggih untuk mendeteksi

DOI : <http://dx.doi.org/10.32409/jikstik.21.1.2897>,
*)Penulis Korespondensi

kendaraan dikarenakan berbagai macam bentuk, warna dan perspektif (Faqih Rofii, 2021).

Kecepatan berkendara pada lalu lintas di jalan raya yang mengakibatkan meningkatnya kewaspadaan akan pergerakan pada ruas jalan dapat menyebabkan sering terjadinya kecelakaan pada jalan tersebut. Kecelakaan banyak terjadi di kota-kota besar yang ada di Indonesia, penyebab terjadinya kecelakaan beragam seperti hilangnya fokus saat berkendara, mengantuk, dan lain sebagainya. Bukan hanya itu, kecelakaan juga terjadi dapat disebabkan oleh jumlah masyarakat atau penduduk dalam suatu yang begitu padat atau kelalaian penduduk setempat seperti menyeberang jalan tanpa menghiraukan kendaraan sekitar.

Sejumlah metode telah digunakan untuk mendeteksi objek. Penggunaan Metode Haar Cascade Classifier dan LBPH Untuk Pengenalan Wajah Secara Realtime telah ditulis oleh Febrin Ludia Ramadani, 2022. Pada penelitian ini penulis menggabungkan dua metode haar cascade classifier dan LBPH untuk membuat sistem yang dapat mengenali wajah seseorang. Proses pengambilan dataset dilakukan secara otomatis, saat sistem sudah mendeteksi adanya wajah manusia dan diambil sebanyak 40 foto untuk setiap satu wajah user. Keberhasilan sistem ini sebesar 88,42%.

Peneliti lain, Abidin, 2018, meneliti Deteksi Wajah Menggunakan Metode Haar Cascade Classifier Berbasis Webcam Pada Matlab. Penelitian ini bertujuan untuk mengimplementasikan algoritma Haar Cascade Classifier kedalam sebuah aplikasi deteksi wajah dengan menggunakan aplikasi MATLAB R2017 a. Uji coba dilakukan berbedabeda dengan masing-masing mendapatkan perlakuan variasi yang sama yaitu : kemiringan sudut posisi citra wajah, jarak wajah terhadap camera webcam dan intensitas cahaya.

M. Fazal Mustaqim, 2021 melakukan penelitian dengan judul Sistem Deteksi Kecepatan Kendaraan Menggunakan Metode Haar Cascade untuk Keamanan Berkendara. Pada penelitian ini metode yang digunakan adalah deteksi objek kendaraan menggunakan haar cascade dan metode perhitungan kecepatan yaitu jarak aktual dibagi dengan waktu. Hasil penelitian menunjukkan bahwa sistem pendeteksi kecepatan kendaraan untuk tingkat keamanan berkendara memiliki tingkat keberhasilan deteksi sebesar 70.6%, jadi dapat disimpulkan bahwa program ini memiliki tingkat akurasi yang tergolong cukup tinggi dan permasalahan yang terjadi dapat terselesaikan.

Metode haar cascade classifier diterapkan dalam penelitian ini, yang merupakan hasil modifikasi dari sistem deteksi wajah yang disusun oleh Viola dan Jones, dengan prinsip deteksi objek berdasarkan nilai sederhana dari fitur tetapi bukan nilai piksel dari citra objek. Metode ini sangat efektif dalam mendeteksi objek, yang merupakan gabungan beberapa konsep yaitu haar features, integral image, adaboost learning, dan cascade classifier.

Pencarian ciri dilakukan dengan menggunakan data-data latih yang dipelajari oleh model program. Metode ini dapat melakukan pengidentifikasian obyek citra dengan kecepatan olah sebesar 0,067 detik. Hasil ini menunjukkan bahwa metode Haar cascade lebih cepat dari metode-metode pengidentifikasian obyek lainnya, seperti metode berbasis neural network, yaitu Baluja-Kanade dan metode Schneiderman-Kanade. Maka karena itu, tingginya kecepatan pengolahan citra yang dilakukan metode Haar cascade dapat menjadi metode yang baik untuk mengidentifikasi kendaraan dan melacak pergerakan kendaraan.

Pada penelitian ini tujuan penelitiannya adalah menghasilkan sistem penghitung jumlah kendaraan dan pendeteksi kecepatan kendaraan pada ruas jalan menggunakan Metode Haar Cascade Classifier.

Machine Learning

Machine Learning adalah studi tentang algoritma untuk mempelajari bagaimana melakukan tugas-tugas tertentu yang dilakukan secara otomatis oleh orang-orang. Di sini, belajar mengacu pada kemampuan untuk melakukan berbagai kegiatan yang sudah ada atau untuk mengekstrapolasi kesimpulan baru dengan benar dari berbagai pola yang diamati sebelumnya. Salah satu subbidang kecerdasan buatan, Machine Learning, dapat berdampak pada sejumlah bidang lain, termasuk statistik, matematika, dan banyak bidang teoretis ilmu komputer. Intinya, tujuan machine learning adalah untuk mengembangkan algoritma yang dapat mengeksekusi sistem pembelajaran otonom dengan sangat sedikit masukan dari orang-orang pada umumnya (Fathurohman, 2021).

Deep Learning

Deep Learning merupakan metode learning yang memanfaatkan artificial neural network yang berlapis-lapis (multi layer) (Pulung Adi Nugroho, 2020). Artificial Neural Network ini dibuat mirip otak manusia, dimana neuron-neuron terkoneksi satu sama lain sehingga membentuk sebuah jaringan neuron yang sangat rumit. Deep Learning atau deep structured learning atau hirarchical learning atau deep neural merupakan metode learning yang memanfaatkan multiple non-linier transformation, deep learning dapat dipandang sebagai gabungan machine learning dengan AI (artificial neural network).

Beberapa algoritma termasuk dalam kategori Deep Learning antara lain:

1. Convolutional Networks
2. Restricted Boltzmann Machine (RBM)
3. Deep Belief Networks (DBN)
4. Stacked Autocoders

Haar Cascade Metode

Haar Cascade adalah salah satu jenis machine learning yang sering digunakan sebagai object detection, pada video maupun gambar. Metode ini menerapkan sample haar like features. Classifier ini akan menggunakan gambar yang ukurannya tetap (pada umumnya berukuran 24x24). Haar dalam mendeteksi objek mempunyai cara kerja yaitu menggunakan teknik sliding window yang memiliki ukuran 24x24 pada semua gambar dan mencari ada atau tidaknya bagian pada gambar yang berbentuk seperti objek (Alif Yudhistira Putra Bayu, 2021).

Haar Cascade menggunakan cascade function untuk mengambil gambar melalui 4 tahapan: (1) menentukan Haar features (2) mengambil gambar integral, (3) Adaboost training dan (4) melakukan klasifikasi dengan cascading classifier.

Library

Library pada Python merupakan gabungan dari package dan module untuk memudahkan dalam membuat suatu sistem atau aplikasi. Library merupakan sebutan kode program tambahan untuk keperluan tertentu, dengan penggunaan library ini dapat menghasilkan kode program secara efisien tanpa perlu menuliskan seluruh skrip. Library bersifat reusable yang dapat digunakan berkali-kali (Ni Kadek Ratna Saria, 2021).

Open CV

OpenCV (Open Source Computer Vision Library) adalah sebuah library open source yang dikembangkan oleh intel yang fokus untuk menyederhanakan programming terkait citra digital (Darujati, 2021). Di dalam OpenCV sudah mempunyai banyak fitur, antara lain: pengenalan wajah, pelacakan wajah, deteksi wajah, kalman filtering, dan berbagai jenis metode AI (Artificial Intelligence) dan menyediakan berbagai algoritma sederhana terkait computer vision untuk low level, OpenCV merupakan open source computer vision library untuk bahasa pemrograman C/C++, dan telah dikembangkan ke Phyton, Java, Matlab.

OpenCV mempunyai banyak fitur yang dapat dimanfaatkan, berikut ini adalah fitur utama dari OpenCV antara lain:

1. Image and video I/O dengan antar muka ini kita dapat membaca data gambar dari file, atau dari umpan video langsung dan juga dapat menciptakan file gambar maupun video.
2. Computer vision secara umum dan pengolahan citra digital (untuk low dan mid level API) dengan antar muka ini kita dapat melakukan eksperimen uji coba dengan berbagai standar algoritma computer vision. Termasuk juga

deteksi garis, tepi, pucuk, proyeksi elips, image pyramid untuk pemrosesan gambar multi skala, pencocokan template, dan berbagai transform (Fourier, cosine diskrit, distance transform) dan lain-lain.

3. Modul computer vision highlevel
4. Metode untuk AI dan machine learning aplikasi computer vision sering kali memerlukan machine learning atau metode AI lainnya, beberapa metode tersebut tersedia dalam paket OpenCV machine learning.

Numpy

Numpy merupakan salah satu pustaka pendukung untuk bahasa pemrograman python. Numpy memberikan dukungan untuk python dalam hal matriks atau himpunan dengan multidimensi yang cukup besar. Numpy merupakan library yang lebih berfokus kepada scientific computing (Avrijsto Amandri Achyar, 2022).

PyTorch

PyTorch merupakan sebuah library yang dapat digunakan dalam machine learning, deep learning, dan bahkan natural language processing yang dikembangkan oleh Facebook AI (sekarang Meta AI) pada tahun 2016. PyTorch pada dasarnya adalah compiled library /shared object yakni sebuah file yang isinya adalah fungsi-fungsi yang dapat di eksekusi pada masing-masing sistem operasi atau secara native.

Format file berbeda-beda pada tiap sistem operasi, yaitu .dll pada windows dan .so pada linux. PyTorch dikembangkan menggunakan c++, namun PyTorch memiliki frontend API berbasis python yang sangat mempermudah penggunaan library tersebut. Selain itu, PyTorch juga memiliki frontend API c++ bernama LibTorch yang dimana digunakan pada proyek ini, namun pada bahasa pemrograman Rust menggunakan wrapper bernama tch-rs (<https://github.com/LaurentMazare/tch-rs>).

Model yang dapat di-load pada LibTorch adalah Torchscript yang pada dasarnya adalah kumpulan fungsi yang dapat dieksekusi tanpa harus ada ketergantungan pada python atau python environment contohnya pada aplikasi c++ atau rust (seperti proyek ini) (Arsil Qodryantha, 2023).

Jupyter Notebook

Jupyter adalah aplikasi web gratis untuk yang digunakan untuk membuat dan membagikan dokumen yang memiliki kode, hasil hitungan, visualisasi, dan teks. Jupyter adalah singkatan dari tiga bahasa pemrograman Julia (Ju), Python (Py), dan R. Tiga bahasa pemrograman ini adalah sesuatu yang penting bagi seorang data scientist. Jupyter berfungsi untuk membantu kamu dalam membuat

narasi komputasi yang menjelaskan makna dari data di dalamnya dan memberikan insight mengenai data tersebut (Medina, 2022).

Metode Penelitian

Metode penelitian yang dilakukan pada penelitian ini meliputi 4 tahapan, yaitu: analisis kebutuhan, pengolahan data, perancangan tampilan antar muka program dan implementasi kode program. Gambar 1 memperlihatkan tahapan penelitian.



Gambar 1: Tahapan penelitian

Analisis Kebutuhan

Perangkat Penelitian ini membutuhkan software dan hardware yang compatible dengan pembuatan sistem. Perangkat keras yang digunakan adalah satu unit PC/Laptop dengan spesifikasi minimum AMD Ryzen 5 3500U, 8GB RAM, SSD, Radeon Vega Mobile Gfx 2.10 GHz dan perangkat lunak yang dibutuhkan adalah Windows 10 sebagai sistem operasi, Jupyter Notebook sebagai text editor program, Opencv sebagai library utama program, Draw.io sebagai perangkat pembuat flowchart.

Pengolahan Data

Pada Tahap ini menjelaskan data apa saja yang di butuhkan untuk melakukan pendeteksian kecepatan pada kendaraan. Peneliti ini menggunakan metode Haar Cascade Classifier yang membutuhkan hasil training berupa cascade.

Banyaknya pada data penghitung dan pendeteksi kecepatan yang digunakan merupakan hasil dari training dengan menggunakan metode Haar Like Feature, kemudian hasil tersebut akan menghasilkan file yang berekstensi *.xml. Data xml yang digunakan diantaranya adalah data untuk mendeteksi objek kendaraan yaitu file “cars.xml”, yang digunakan untuk mendeteksi objek kendaraan sebagai parameternya.

Perancangan Tampilan Antarmuka Program

Haar Cascade Classifier merupakan rectangular (persegi) feature, yang memberikan indikasi secara spesifik pada sebuah gambar atau image. Pada tampilan ini akan terjadi pendeteksian, pengklasifikasian, dan menghitung kecepatan objek kendaraan. Pada proses pendeteksian objek, setiap objek pada kendaraan yang terdeteksi akan di berikan bounding box beserta jumlah kecepatan kendaraan yang terdeteksi. Kemudian terdapat garis intersect yang tersedia, dimana jika

objek kendaraan tersebut menyentuh atau melewati garis intersect tersebut, maka program akan menampilkan jumlah kendaraan yang terdeteksi dan akan muncul jumlah kecepatan pada kendaraan tersebut.

Implementasi Kode Program

Perintah berikut ini adalah kode python yang menggunakan library Open CV2 untuk bekerja dengan gambar dan video, serta modul time untuk mengatur waktu jeda (Sleep).

```
import cv2
from time import sleep
```

Pada perintah `cascade_src = 'cars.xml'` bekerja untuk menginisialisasi variabel model cascade detector yang telah dilatih sebelumnya untuk melatih objek mobil dalam gambar, dan nama variabel `video_src = 'video2.mp4'` adalah nama video yang akan digunakan untuk mendeteksi mobil.

`cap=cv2.VideoCapture(video_src)` pada perintah ini menggunakan baris open cv untuk membuka file video yang ditentukan oleh `video_src`, dan akan dibaca frame demi frame pada video tersebut.

```
cascade_src = 'cars.xml'
video_src = 'video2.mp4'

cap = cv2.VideoCapture(video_src)
car_cascade =
cv2.CascadeClassifier(cascade_src)
```

Pada perintah `delay = 600` untuk mengatur waktu tunda dalam milidetik antara setiap iterasi frame video. `detec = []` digunakan untuk menyimpan informasi tentang mobil yang terdeteksi seperti koordinat dan waktu terdeteksi. `pos_line = 550` ini adalah mobil yang akan melewati garis akan di hitung. `offset = 10` membantu menghindari deteksi ganda jika mobil bergerak dengan cepat. `car = 0` digunakan untuk menghitung jumlah mobil yang melewati garis referensi. `prev_time = cv2.getTickCount() / cv2.getTickFrequency()` digunakan untuk menghitung waktu antara dua deteksi mobil berturut-turut. `prev_center = None` digunakan untuk menghitung arah pergerakan mobil.

```
delay = 600
detec = []
pos_line = 550
offset = 10
car = 0
prev_time = cv2.getTickCount() /
cv2.getTickFrequency()
prev_center = None
```

Kemudian untuk mengubah gambar ke skala abu-abu adalah langkah yang umum dalam analisis citra karena lebih sederhana dan biasanya cukup untuk tujuan deteksi objek. variabel yang akan berisi daftar dari persegi panjang (bounding box) yang mengindikasikan lokasi perkiraan mobil dalam gambar. Setiap bounding box akan memiliki empat nilai, yaitu koordinat x dan y dari pojok kiri atas, serta lebar dan tinggi bounding box

```
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
cars = car_cascade.detectMultiScale(gray, 1.1,
1)
cv2.line(img, (25, pos_line), (1200, pos_line),
(255, 127, 0), 3)
```

Program akan menambahkan teks pada gambar yang menunjukkan informasi tentang kendaraan yang lewat (berdasarkan variabel car), dan kemudian menampilkan gambar dengan teks tersebut dalam jendela tampilan. Teks akan ditempatkan pada koordinat (450, 70) dengan ukuran font 2 dan warna merah.

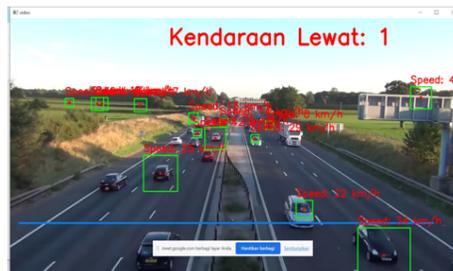
```
cv2.putText(img, "Kendaraan Lewat: " +
str(car), (450, 70), cv2.FONT_HERSHEY_SIMPLEX,
2, (0, 0, 255), 5)
cv2.imshow('video', img)
```

Hasil dan Pembahasan

Penelitian ini telah menghasilkan program yang dapat digunakan untuk menghitung jumlah kendaraan dan menampilkan informasi berapa kecepatan kendaraan yang lewat di ruas jalan raya. Hasil dari program diujikan pada kondisi nyata, untuk melihat apakah program dapat berjalan sesuai dengan harapan.

Pengujian Sistem Deteksi Kendaraan

Untuk mendapatkan hasil yang signifikan, dalam tahap ini, video digunakan sebagai data masukan. Video tersebut menggambarkan lalu lintas di jalan yang ramai dengan kendaraan bergerak di atasnya. Pengamatan dilakukan untuk mendapatkan wawasan tentang pelanggaran lalu lintas yang sering terjadi, salah satunya adalah dengan mengumpulkan informasi tentang kecepatan kendaraan. Ini menjadi penting karena salah satu faktor penyebab kecelakaan adalah pengemudi yang melampaui batas kecepatan maksimal yang berpotensi membahayakan pengemudi lain.



Gambar 2: Hasil Pengujian Kendaraan

Tabel 1: Hasil Ujicoba Program

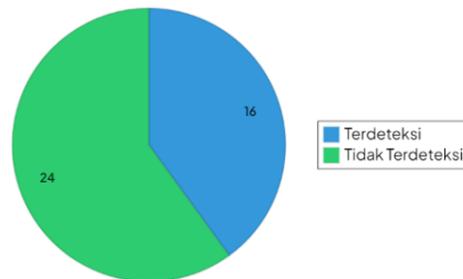
Urutan Mobil	ID Mobil	Kecepatan (km/jam)	Terdeteksi
1	1	26	Ya
2	2	25	Ya
3	3	27	Ya
4	-	-	-
5	-	-	-
6	4	42	Ya
7	-	-	-
8	-	-	-
9	-	-	-
10	-	-	-
11	-	-	-
12	5	36	Ya
13	-	-	-
14	-	-	-
15	-	-	-
16	6	26	Ya
17	-	-	-
18	-	-	-
19	7	40	Ya
20	-	-	-
21	-	-	-
22	-	-	-
23	-	-	-
24	-	-	-
25	-	-	-
26	-	-	-
27	-	-	-
28	8	42	Ya
29	9	15	Ya
30	10	19	Ya
31	11	24	Ya
32	12	18	Ya
33	13	20	Ya
34	14	25	Ya
35	-	-	-
36	-	-	-
37	15	22	Ya
38	-	-	-
39	-	-	-
40	16	19	OK

Pada tabel tersebut, terdapat 16 mobil yang berhasil terdeteksi dari total keseluruhan 40 mobil yang melintas di jalan. Mobil-mobil ini memiliki kecepatan yang bervariasi. Untuk mengukur keberhasilan program ini, rumus yang digunakan adalah sebagai berikut:

$$TingkatKeberhasilandeteksikendaraan =$$

$$\frac{JumlahMobilTerdeteksi}{TotalMobil} \times 100 \quad (1)$$

Dengan menggunakan rumus ini, kita dapat menentukan tingkat keberhasilan dari metode yang diterapkan dalam pengembangan program ini 40.0 %. Data tersebut dapat direpresentasikan dalam bentuk grafik sebagai berikut.



Gambar 3: Grafik Hasil Deteksi Kecepatan Kendaraan

Pada gambar 2 terlihat bahwa kendaraan dapat dideteksi dengan menampilkan box warna hijau dan menampilkan kecepatan setiap kendaraan. Hasil Ujicoba terlihat pada tabel 1 berikut ini.

Penutup

Dari penelitian yang telah dilakukan, dapat disimpulkan:

1. Sistem Penghitung Jumlah Kendaraan dan pendeteksi kecepatan menggunakan metode haar cascade classifier dengan open cv telah berhasil dibuat.
2. Deteksi Kendaraan Menggunakan metode Haar Cascade Classifier belum sempurna, karena ada beberapa kendaraan yang tidak terdeteksi. Grafik yang di hasilkan oleh sistem ini di rancang menggunakan dari hasil kendaraan yang terdeteksi dan tidak terdeteksi. terdapat 16 mobil yang berhasil terdeteksi dari total keseluruhan 40 mobil yang melintas di jalan. Mobil-mobil ini memiliki kecepatan yang bervariasi. Untuk mengukur keberhasilan program ini.
3. Hasil kinerja dari sistem ini belum dikatakan sempurna karena pada saat uji coba terdapat beberapa kendaraan yang tidak terdeteksi ataupun yang terhitung ganda sehingga tidak terhitung oleh program sistem.

Terdapat beberapa saran yang bisa diambil sebagai pertimbangan dalam upaya mengembangkan sistem ini agar lebih baik:

1. Mengintegrasikan teknik pelacakan objek seperti Kalman Filtering untuk mengikuti pergerakan kendaraan secara berkelanjutan dan memastikan deteksi yang lebih konsisten.
2. Meningkatkan tahap preprocessing data untuk mengatasi masalah seperti objek yang keluar dari bidang pandang.
3. Mencoba menggunakan metode selain Haar cascade classifier untuk mendeteksi kendaraan agar dapat mengatasi masalah kotak pembatas ganda

Daftar Pustaka

- [1] Dufan J. P. Manajang, S. R. S. A. J., 2020. Implementasi Framework Tensorflow Object Detection Dalam Mengklasifikasi Jenis Kendaraan Bermotor. *Jurnal Teknik Informatika*, 15(3), pp. 171-178.
- [2] Faqih Rofii, G. P. M. I. F. A. S., 2021. Peningkatan Akurasi Penghitungan Jumlah Kendaraan dengan Membangkitkan Urutan Identitas Deteksi Berbasis Yolov4 Deep Neural Networks. *TEKNIK*, pp. 169-177.
- [3] Febrin Ludia Ramadini, E. H., 2022. Penggunaan Metode Haar Cascade Classifier dan LBPH Untuk Pengenalan Wajah secara realtime. *Jurnal Nasional Informatika Dan Teknologi Jaringan*, 6(2)
- [4] Abidin S., 2018. Deteksi Wajah Menggunakan Metode Haar Cascade Classifier Berbasis Webcam Pada Matlab. *Jurnal Teknologi Elektika*, 15(1), pp. 21-27.
- [5] M. Fazal Mustaqim, A. N. A. F. S., 2021. Sistem Deteksi Kecepatan Kendaraan Menggunakan Metode Haar Cascade untuk keamanan berkendara. *Edu Elektika Journal*, 10(2), pp. 30-34.
- [6] Panji Ahmad Nurhusnia, H. N. R. F., 2022. Sistem Pendeteksian Kecepatan Kendaraan Menggunakan Haar Cascade Berbasis Raspberry PI. *Prosiding Seminar Nasional Fisika*, Volume 10, pp. 131-136.
- [7] Rony, S., 2021. dicoding. [Online] Available at: <https://www.dicoding.com/blog/mengenal-deep-learning/> [Accessed 9 October 2023].
- [8] Rohman, Y. A., 2019. Medium. [Online] Available at: <https://medium.com/@yasirabd/pengenalan-numpy-pandas-matplotlib-b90bafd36c0> [Accessed 9 December 2013].
- [9] Feradhita, 2020. LOGIQUE. [Online] Available at: <https://www.logique.co.id/blog/2020/07/09/-pengertian-python/> [Accessed 9 July 2023]
- [10] Fathurohman, A., 2021. Machine Learning Untuk Pendidikan: Mengapa Dan Bagaimana. *JURNAL JITEK*, 1(3), pp. 57-62.
- [11] Pulung Adi Nugroho, I. F. R. A., 2020. Implementasi Deep Learning Menggunakan Convolutional Neural Network (Cnn) Pada Ekspresi Manusia. *JURNAL ALGOR*, 2(1), pp. 12-21.
- [12] Alif Yudhistira Putra Bayu, S. S., 2021. Sistem Pemantauan Penggunaan Protokol kesehatan Covid-19 Menggunakan Metode haar Cascade Dan Neural Network. *Jurnal Qua Teknik*, 11(2), pp. 32-46.
- [13] Darujati, T. S. d. C., 2021. Pengolahan Citra Untuk Pengenalan Wajah (Face Recognition) Menggunakan OpenCV. *Jurnal Syntax Admiration*, 2(3), pp. 534-545.
- [14] Avrijsto Amandri Achyar, A. M. O. M. R. P. A. S. A. D. G., 2022. Identifikasi Ras Wajah dengan Menggunakan Metode Deep Learning Model Keras. *Jurnal Teknik Mesin, Industri, Elektro Dan Informatika (JTMEI)*, 1(1), pp. 29-37.
- [15] Arsil Qodryantha, R. A. R. A. S., 2023. Pembangunan Sistem Pengawasan Cerdas Dengan Visualisasi 3D. 9(1), pp. 465-460.
- [16] Medina, M. I., 2022. glints. [Online]. Available at: <https://glints.com/id/lowongan/jupyter-adalah/> [Accessed 23 November 2022].
- [17] Ni Kadek Ratna Saria, I. M. A. D. S. P. W. B., 2021. Perbandingan Translation Library Pada Python (Studi Kasus: Analisis Sentimen Penyakit Menular Di Indonesia). *Jurnal Ilmiah Teknologi dan Komputer*, 2(3).