

Studi Kasus Penggunaan YOLO dan OpenCV untuk MENDETEKSI JENIS KENDARAAN di JALAN

**Santoso Adi Nugroho^{1*}, Muhammad Kahfi², Mochammad Fidzri Akhbar Alamsyah³,
Alice Natanael⁴, Perani Rosyani⁵**

¹⁻⁵Fakultas Ilmu Komputer, Teknik Informatika, Universitas Pamulang, Jl. Raya Puspipetek No. 46,
Kel. Buaran, Kec. Serpong, Kota Tangerang Selatan. Banten 15310, Indonesia

Email: ^{1*}adiny26@gmail.com, ²mhmdkahfi21@gmail.com, ³akbaralamsyah74@gmail.com,
⁴alicenatanael05@gmail.com, ⁵dosen00837@unpam.ac.id

(* : coressponding author)

Abstrak-Deteksi kendaraan merupakan salah satu aspek penting dalam sistem pemantauan lalu lintas dan manajemen transportasi modern. Dalam penelitian ini, kami mengembangkan sebuah aplikasi untuk mendeteksi jenis kendaraan pada jalan raya dengan OpenCV dan model YOLOv4 (You Only Look Once). Aplikasi ini mampu mengidentifikasi dan mengklasifikasikan berbagai **macam** kendaraan seperti mobil, truk, bus, dan sepeda motor dari video atau kamera langsung. Proses deteksi melibatkan beberapa tahap, mulai dari pemuatan model YOLOv4 yang telah terlatih, membaca input dari video atau kamera, hingga mengolah dan menampilkan hasil deteksi dengan bounding box dan label klasifikasi. Hasil penelitian menunjukkan bahwa sistem ini dapat mendeteksi dan mengklasifikasikan kendaraan dengan tingkat akurasi yang memadai, memungkinkan implementasi dalam berbagai aplikasi seperti pengawasan lalu lintas, pengumpulan data transportasi, dan pengembangan sistem keamanan. Keunggulan dari penggunaan YOLOv4 adalah kecepatan dan efisiensinya dalam mendeteksi objek secara real-time, sehingga cocok untuk diaplikasikan pada lingkungan dengan kecepatan tinggi seperti jalan raya.

Kata Kunci: Deteksi Kendaraan, YOLO, OpenCV, Real-Time, Pengawasan Lalu Lintas

Abstract-Vehicle detection is a crucial aspect of modern traffic monitoring and transportation management systems. In this research, we developed an application to detect vehicle types on the road using OpenCV and the YOLOv4 (You Only Look Once) model. This application can identify and classify various vehicle types such as cars, trucks, buses, and motorcycles from videos or live camera feeds. The detection process involves several stages, starting from loading the pre-trained YOLOv4 model, reading input from videos or cameras, to processing and displaying detection results with bounding boxes and classification labels. The study's findings demonstrate that the system has sufficient accuracy for vehicle identification and categorization, enabling it to be applied in a multitude of ways, as well as tracking traffic, transportation data collection, and also the creation of security systems. The advantage of using YOLOv4 lies in its speed and efficiency in real-time object detection, making it suitable for high-speed environments like highways.

Keyword: Vehicle Detection, YOLOv4, OpenCV, Real-Time, Traffic Monitoring

1. PENDAHULUAN

Deteksi kendaraan merupakan komponen penting dalam sistem pemantauan lalu lintas dan manajemen transportasi modern. Dengan meningkatnya jumlah kendaraan di jalan, kebutuhan akan teknologi yang dapat mengidentifikasi dan mengklasifikasikan jenis kendaraan secara otomatis menjadi semakin mendesak. Deteksi kendaraan tidak hanya membantu dalam pemantauan lalu lintas secara real-time tetapi juga memberikan data yang berharga untuk analisis lalu lintas, perencanaan kota, dan pengembangan sistem keamanan.

Dalam beberapa tahun terakhir, perkembangan teknologi pembelajaran mesin dan visi komputer telah memungkinkan pengembangan sistem deteksi objek yang lebih canggih dan efisien. Salah satu model yang paling menonjol dalam deteksi objek adalah YOLO (You Only Look Once). YOLOv4, versi terbaru dari model ini, menawarkan kecepatan dan akurasi yang lebih baik, membuatnya ideal untuk aplikasi deteksi kendaraan di lingkungan berkecepatan tinggi seperti jalan raya.

Observasi ini bertekad untuk mengelaborasi aplikasi yang berperan dalam mendeteksi dan mengklasifikasikan macam kendaraan pada jalan raya dengan OpenCV dan model YOLOv4. Aplikasi ini diharapkan dapat mengidentifikasi berbagai macam kendaraan seperti mobil, truk, bus, dan sepeda motor dari video atau umpan kamera langsung.

2. METODOLOGI PENELITIAN

2.1 Tahapan Observasi

a. Pengambilan Informasi:

1. Dataset pada observasi ini adalah model YOLOv4 yang telah dilatih menggunakan dataset COCO (Common Objects in Context). Dataset ini mencakup berbagai jenis objek termasuk kendaraan.

b. Pemrosesan Data:

1. Model YOLOv4 yang telah dilatih diintegrasikan ke dalam aplikasi menggunakan OpenCV.
2. File konfigurasi (yolov4.cfg), bobot model (yolov4.weights), dan daftar kelas (coco.names) diimpor ke dalam program untuk digunakan dalam deteksi objek.

c. Pengembangan Aplikasi:

1. Aplikasi dikembangkan menggunakan bahasa pemrograman Python dengan pustaka OpenCV untuk menangkap video dan melakukan deteksi objek.
2. Aplikasi membaca input dari video atau kamera langsung dan melakukan deteksi kendaraan secara real-time.
3. Hasil deteksi ditampilkan dengan menggambar kotak pembatas (bounding box) dan label klasifikasi pada setiap kendaraan yang terdeteksi.

d. Deteksi dan Klasifikasi:

1. Proses deteksi melibatkan beberapa tahapan: ekstraksi fitur menggunakan blob dari gambar input, input blob ke dalam model YOLOv4, dan mendapatkan output dari model.
2. Hasil deteksi diproses lebih lanjut untuk menghitung koordinat kotak pembatas dan kelas dari setiap objek yang terdeteksi.
3. Non-Maximum Suppression (NMS) digunakan untuk menghilangkan deteksi ganda dan hanya mempertahankan deteksi dengan kepercayaan tertinggi.

e. Evaluasi dan Pengujian:

1. Aplikasi diuji menggunakan berbagai video dan umpan kamera langsung untuk mengevaluasi kinerja deteksi dan klasifikasi kendaraan.
2. Kinerja diukur berdasarkan akurasi deteksi, kecepatan pemrosesan, dan kemampuan aplikasi dalam kondisi pencahayaan dan lingkungan yang berbeda

Dengan metodologi ini, penelitian ini diharapkan dapat menghasilkan aplikasi yang efektif dan efisien untuk mendeteksi dan mengklasifikasikan jenis kendaraan di jalan, mendukung upaya peningkatan keselamatan dan efisiensi transportasi.

3. ANALISA DAN PEMBAHASAN

3.1 Definisi Sistem

Teknik mengetahui macam kendaraan dengan tools OpenCV dan YOLOv4 dirancang untuk mengidentifikasi dan mengklasifikasikan berbagai jenis kendaraan dalam rekaman video atau umpan kamera langsung. Sistem ini bertujuan untuk memberikan deteksi kendaraan secara real-time dengan akurasi tinggi, sehingga dapat digunakan untuk pemantauan lalu lintas, analisis transportasi, dan peningkatan keselamatan jalan.

3.2 Arsitektur Sistem

Arsitektur sistem terdiri dari beberapa komponen utama:

a. Input Data:

Video atau umpan kamera langsung yang menyediakan gambar kendaraan di jalan.

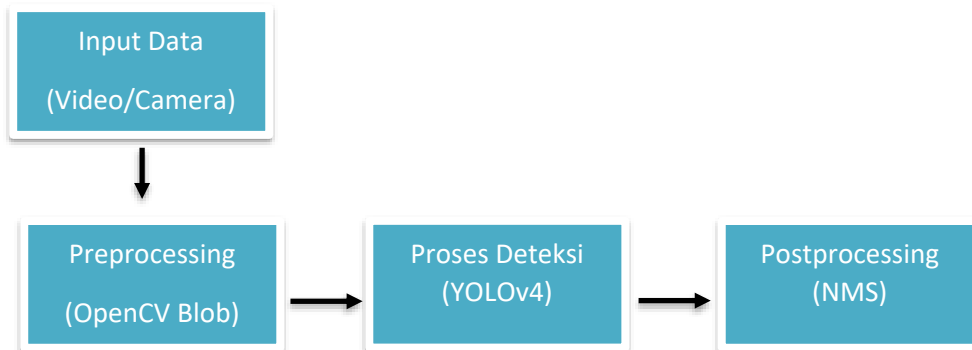
b. Model Deteksi:

Model YOLOv4 yang telah dilatih untuk mendeteksi objek termasuk kendaraan.

c. Preprocessing:

Ekstraksi fitur dari gambar input menggunakan blob dari OpenCV.

- d. Proses Deteksi:**
 Proses inferensi menggunakan model YOLOv4 untuk mendeteksi kendaraan dan menghasilkan kotak pembatas serta label klasifikasi.
- e. Post-processing:**
 Penerapan Non-Maximum Suppression (NMS) untuk mengeliminasi deteksi ganda dan mempertahankan deteksi dengan kepercayaan tertinggi.
- f. Output:**
 Tampilan gambar dengan kotak pembatas dan label klasifikasi untuk setiap kendaraan yang terdeteksi.



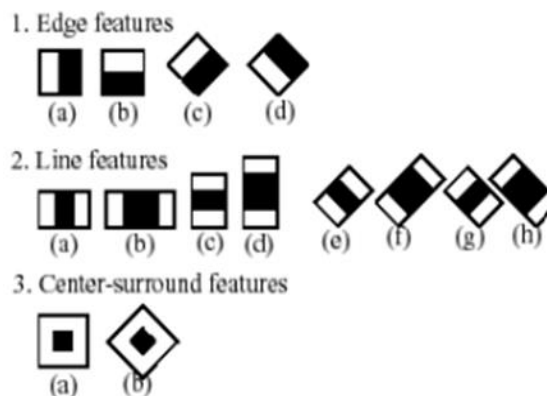
Gambar 1. Proses Alur Arsitektur Sistem

3.3 OpenCV

Merupakan perpustakaan perangkat lunak untuk computer vision dan machine learning yang tersedia secara gratis (lisensi BSD). Guna memfasilitasi penggunaan realtime visi komputer secara akademis maupun komersil. Library ini dapat digunakan pada beberapa bahasa pemrograman yang umum dijumpai seperti C, C++, Python, Java. Tak hanya itu library ini dapat diaplikasikan pada operation System seperti Windows, Linux, Android, iOS dan Mac OS. Library ini mempunyai algoritma yang telah diperbaharui lebih dari 2500 algorithms. Observasi ini menggunakan OpenCV 2.4.13.

3.4 Haar-like

Paul Viola dan Michael Jones menjadi pelopor hadirnya Haar-like Feature yang memiliki fungsi mengidentifikasi rupa seseorang. Fitur ini dikaji ulang oleh Rainer Lienhart dan Jochen Maydt. Cara kerja fitur ini adalah melatih penggolongan data dengan sampel citra objek tertentu. Hal ini mengaplikasikan prinsip algoritma Adaboost. Sampel citra kendaraan dari segala sisi dengan ukuran yang sama disetiap sisinya digunakan pada observasi ini. Sampel ini disebut sebagai sampel positif sedangkan sampel negatif adalah citra dari objek yang lain tetapi mempunyai ukuran seperti sampel positif. Gabungan sampel positif dan negatif ini akan menjadi fitur objek/cascade. Pada algoritma ini akan menunjukkan nilai "0" apabila tidak ditemukan objek sesuai dengan klasifikasi, sedangkan jika sesuai maka ditunjukkan nilai "1".



Implementasi fitur pada setiap domain gambar menunjukkan kuantitas pixel yang berada pada persegi dan persegi panjang berwarna gelap hasil intersection dengan daerah berwarna terang. Koordinat kiri atas (x,y), lebar (w), tinggi (h) dicitrakan menjadi persegi. RecSum(ri) merepresentasikan total pixel pada setiap sampel.

$$\begin{aligned} feature_1 &= \sum_{i=1}^N W_i \times RecSum(r_i) \\ &= \sum_{i=1}^N W_i \times RecSum(x, y, w, h) \end{aligned}$$

3.5 Mean Squared Error

Mean Squared Error (MSE) Mean squared error digunakan untuk mengetahui perbedaan antara estimator dengan hasil estimasi [6]. MSE digunakan sebagai tolok ukur suatu estimator. Hasil yang diperoleh selalu berupa angka positif. Semakin mendekati nol maka semakin baik kinerja estimator tersebut. (2) Dalam penelitian ini, MSE digunakan untuk mengecek apakah sebuah citra merupakan sebuah objek kendaraan atau bukan. Ini dilakukan dengan membandingkan kesimetrisan secara horizontal dan vertikal dari sebuah citra. Sebuah citra kendaraan bersifat simetris secara horizontal maupun vertikal. Kesimetrisan horizontal digunakan untuk mengetahui tinggi objek kendaraan sedangkan kesimetrisan vertikal digunakan untuk mengetahui apakah objek yang terdeteksi sisi kiri dan kanannya simetris atau tidak. Ini dapat mempermudah dalam menandai objek kendaraan yang terdeteksi. Jika perbedaan kesimetrisan terlalu besar, citra yang terdeteksi bukan merupakan objek kendaraan.

$$MSE = \frac{\sum_{i=0}^n (X_i - F_i)^2}{n}$$

Sistem deteksi kendaraan menggunakan OpenCV telah diimplementasikan dan diuji menggunakan dataset gambar dan video dari berbagai sumber. Program CarDetection/Deteksi Kendaraan, memiliki langkah penting yaitu mengekspor video dengan cara mencantumkan nama file dan domain. Selanjutnya program akan memberikan file XML yang akan menjadi acuan dalam menentukan objek sebagai kendaraan atau bukan. Lalu, program akan membaca video secara frame-by frame. Pembacaan ini dilakukan pada wilayah deteksi yang telah ditetapkan yaitu di antara dua garis biru.

Hal ini dilakukan untuk mengurangi risiko kesalahan deteksi. Dengan demikian, pengenalan objek kendaraan akan menjadi lebih fokus. Sehingga pengenalan jenis kendaraan yang dideteksi akan lebih akurat. Kemudian dilanjutkan dengan mendeteksi objek kendaraan menggunakan classifier. Classifier akan mencocokkan objek kendaraan file XML yang telah dimuat sebelumnya. Jika memenuhi syarat, selanjutnya akan dicek kesimetrisan objek kendaraan yang terdeteksi. Objek kendaraan yang terdeteksi akan dicek apakah simetris secara vertikal dan horizontal atau tidak. Jika simetris secara vertikal dan horizontal, objek tersebut dianggap sebagai kendaraan. Aplikasi akan memberikan tanda segi empat pada objek yang dianggap sebagai kendaraan.



Gambar 1. Proses Alur Arsitektur Sistem

2. METODE PENELITIAN

Penelitian ini mengembangkan aplikasi deteksi kendaraan di jalan menggunakan OpenCV dan model YOLOv4. Aplikasi ini dapat mengidentifikasi dan mengklasifikasikan berbagai jenis kendaraan seperti mobil, truk, bus, dan sepeda motor dari video atau umpan kamera langsung.

a. Metodologi Penelitian:

1. Pengumpulan data: Dataset model YOLOv4 yang telah dilatih menggunakan dataset COCO.
2. Pemrosesan data: Integrasi model YOLOv4 ke dalam aplikasi menggunakan OpenCV.
3. Pengembangan aplikasi: Aplikasi dikembangkan dengan Python dan OpenCV untuk menangkap video dan melakukan deteksi objek.
4. Deteksi dan klasifikasi: Ekstraksi fitur, input ke model YOLOv4, dan mendapatkan output dari model.
5. Evaluasi dan pengujian: Pengujian aplikasi dengan berbagai video dan umpan kamera langsung.

b. Hasil:

1. Sistem dapat mendeteksi dan mengklasifikasikan kendaraan dengan akurasi yang memadai.
2. Sistem dapat bekerja di berbagai kondisi pencahayaan dan lingkungan.
3. Keuntungan YOLOv4: kecepatan dan efisiensi dalam deteksi objek real-time.

c. Aplikasi:

1. Pemantauan lalu lintas real-time.
2. Pengumpulan data transportasi.
3. Pengembangan sistem keamanan.

REFERENCES

- Peningkatan Akurasi Penghitungan Jumlah Kendaraan dengan Membangkitkan Urutan Identitas Deteksi Berbasis Yolov4 Deep Neural Networks,
Adji Valentino, Ahmad Satrio Nugroho Rangga, Fajar Wijoyo, Indah Janti Lestari, Tri Puspita Andari, & Puspita Rosyani. (2023). *Jurnal Teknik*, 14(2), 189-202.
- Joko Buliali, & Eka Putri Setyaningtyas. (2023). Deteksi jenis kendaraan di jalan menggunakan opencv.
- Adji Valentino, Ahmad Satrio Nugroho Rangga, Fajar Wijoyo, Indah Janti Lestari, Tri Puspita Andari, & Puspita Rosyani. (2023). *Jurnal Teknik*, 14(2), 189-202.
- Bagas Wicaksono, & Agus Budiman. (2022). Implementasi YOLOv4 Tiny dan Region of Interest untuk meningkatkan performa deteksi objek kendaraan.
- Yuniar Aji, & Muhammad Nur Cahyo. (2022). Pendeteksi gerak berbasis kamera menggunakan opencv pada ruangan. *Jurnal Sains dan Teknologi Universitas Muhammadiyah Surakarta*, 7(1), 1-10.
- Rini Puspita Sari, & Teguh Budiarto. (2022). Pendeteksian objek menggunakan opencv dan metode yolov4-tiny untuk membantu tunanetra, *Jurnal Sains dan Teknologi Komputer*, 9(2), 384-391.