

Deteksi Kecelakaan Lalu Lintas Otomatis Pada Rekaman CCTV Indonesia Menggunakan Deep Learning

Adam Rezky¹, Abdullah Bagir², David Pamerean³, Faizal Makhrus⁴

¹Departemen Ilmu Komputer dan Elektronika, Universitas Gadjah Mada, Sleman, 55281, email: adamrezky2020@mail.ugm.ac.id

²Departemen Ilmu Komputer dan Elektronika, Universitas Gadjah Mada, Sleman, 55281, email: albagir99@mail.ugm.ac.id

³Departemen Ilmu Komputer dan Elektronika, Universitas Gadjah Mada, Sleman, 55281, email: davidpamerean@mail.ugm.ac.id

⁴Departemen Ilmu Komputer dan Elektronika, Universitas Gadjah Mada, Sleman, 55281, email: faizal_makhrus@ugm.ac.id

Corresponding Author: Adam Rezky

INTISARI — Kecelakaan lalu lintas sering kali mengakibatkan korban jiwa dan kerugian harta benda. Penggunaan CCTV bisa digunakan untuk memonitor kecelakaan. Akan tetapi, pengawasan lalu lintas menggunakan CCTV dalam skala besar seperti di Indonesia memerlukan sumber daya manusia yang besar dan sulit untuk implementasi skala besar. Solusi yang diusulkan adalah penggunaan teknologi *deep learning* untuk mendeteksi kecelakaan lalu lintas secara otomatis pada rekaman CCTV lalu lintas Indonesia. Akan tetapi, arsitektur *deep learning* memiliki beban komputasi tinggi dikarenakan kompleksitas dari arsitekturnya dan memberikan tantangan dalam deteksi kecelakaan secara *real-time*. Penelitian ini bertujuan untuk mengembangkan model deteksi kecelakaan dengan beban komputasi yang ringan. Dalam penelitian ini, arsitektur YOLOv8 digunakan untuk deteksi objek pada video CCTV Indonesia yang diambil dari berbagai sumber seperti Twitter, Youtube, dan siaran CCTV publik Indonesia. Untuk mendeteksi adanya kecelakaan, anomali yang terdapat pada video akan dideteksi dengan menggunakan metode *Simple Moving Average* (SMA) dan frame yang memiliki anomali tertinggi akan diklasifikasi oleh model CNN *MobileNetv2* yang dilatih menggunakan dataset kecelakaan yang diperoleh dari Kaggle. Model *MobileNetv2* memprediksi frame yang memiliki anomali dengan akurasi 0.89 dari dua kelas yaitu kecelakaan atau tidak kecelakaan. Model yang kami kembangkan memiliki beban komputasi rata-rata 21 ms/frame, yang menunjukkan bahwa model yang kami kembangkan memiliki beban komputasi relatif rendah apabila dibandingkan dengan metode-metode yang dikembangkan oleh peneliti sebelumnya.

KATA KUNCI — Deteksi Kecelakaan, YOLO, *Deep Learning*, Beban Komputasi.

I. PENDAHULUAN

Kecelakaan lalu lintas di Indonesia adalah permasalahan serius yang sering kali mengakibatkan korban jiwa dan kerugian harta benda. Penanganan pertama yang cepat dan efisien menjadi sangat penting dalam situasi ini, mengingat setiap menit sangat berarti bagi keselamatan korban [1]. Metode konvensional penanganan kecelakaan, yaitu menghubungi nomor darurat, bisa mengalami hambatan seperti penundaan dan sulit diterapkan bila lokasi insiden berada di tempat yang terisolasi. Sebagai alternatif, penggunaan CCTV bisa digunakan untuk memonitor kecelakaan. Akan tetapi, pengawasan lalu lintas menggunakan CCTV dalam skala besar seperti di Indonesia memerlukan sumber daya manusia yang besar dan menjadi sulit untuk dipertahankan dalam proses implementasi.

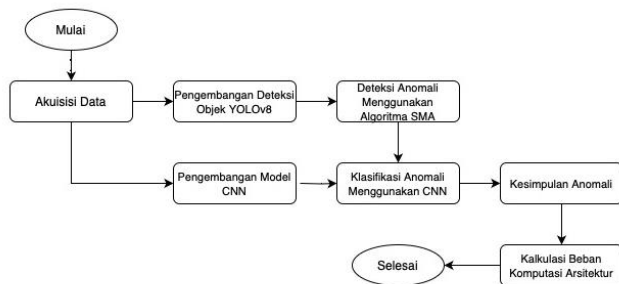
Solusi yang diusulkan adalah penggunaan teknologi Kecerdasan Artifisial (*Artificial Intelligence*) seperti pengembangan arsitektur *deep learning*, yang mampu untuk mengotomasi deteksi kecelakaan dalam citra CCTV [2]. Dengan teknologi ini, sistem respon cepat otomatis yang mampu mendeteksi kecelakaan lalu lintas dan memberikan peringatan awal kepada institusi penanganan pertama dapat dikembangkan. Penelitian-penelitian mengenai teknologi ini sudah dilaksanakan oleh beberapa peneliti. Referensi [3] mengembangkan model *Convolutional Neural Network* (CNN) untuk mendeteksi kecelakaan pada kamera CCTV dan mendapatkan jumlah deteksi tinggi, tetapi model tersebut mengalami kesulitan dalam memproses informasi temporal pada sekuensi video secara efektif. Referensi [4] mengembangkan arsitektur R-CNN dengan akurasi 90% dengan limitasi waktu proses yang tinggi. Penelitian-penelitian tersebut memiliki akurasi yang tinggi dalam mendeteksi objek. Akan tetapi, model-model penelitian tersebut memiliki kompleksitas yang

tinggi, sehingga menghasilkan beban komputasi yang berat. Hal tersebut akan mempersulit teknologi ini untuk diterapkan pada infrastruktur deteksi otomatis kamera CCTV lalu lintas.

Penelitian ini akan menekankan kompleksitas model ringan, sehingga akan lebih mudah untuk implementasi infrastruktur deteksi kecelakaan lalu lintas nyata. Pilihan arsitektur *deep learning* pada penelitian ini adalah *You Only Look Once* v8 (YOLOv8) dan *MobileNetv2* yang keduanya merupakan model CNN. Model YOLOv8 merupakan versi terbaru dari model CNN YOLO yang dikembangkan oleh [5] dan akan digunakan untuk proses deteksi objek dan mengidentifikasi anomali pada video. Referensi [6] menyatakan bahwa model YOLOv8 memiliki akurasi tinggi dan kecepatan inferensi tinggi, sehingga model melakukan pembelajaran pada data dengan waktu singkat. Model *MobileNetv2* dikembangkan oleh [7]. Konsiderasi penggunaan *MobileNetv2* pada penelitian ini adalah adanya fitur *depthwise separable convolution*, yang mengaplikasikan filter spasial terhadap kanal input secara terpisah dan mengambil informasi spasial lokal sehingga mengurangi jumlah parameter pembelajaran [8], sehingga meringankan beban komputasi model. *MobileNetv2* akan mengklasifikasikan anomali yang terdeteksi oleh model YOLOv8 ke dalam dua kelas yaitu *Accident/No Accident*. Beban komputasi dari arsitektur akan diukur dan dibandingkan dengan metode-metode yang dikembangkan oleh peneliti lain. Arsitektur deteksi kecelakaan dengan beban komputasi ringan akan membuka jalan untuk infrastruktur skala besar deteksi kecelakaan kamera CCTV.

II. METODE

Pada Gambar 1, prosedur dari penelitian ini diperlihatkan dalam bentuk *flowchart*. Penelitian akan menggunakan arsitektur *You Only Look Once v8* (YOLOv8) dan memanfaatkan rekaman dari CCTV lalu lintas Indonesia yang diperoleh dari berbagai sumber. Dari data rekaman yang diperoleh, arsitektur deteksi objek akan dikembangkan untuk melacak pergerakan objek-objek pada gambar. Dari rekam jejak objek-objek, anomali dari pergerakan objek akan dideteksi menggunakan algoritma *Simple Moving Average* (SMA).



Gambar 1. Prosedur Penelitian

Untuk menghindari false alarm, anomali akan diklasifikasikan sebagai Kecelakaan atau Non Kecelakaan menggunakan model *MobileNetv2* yang akan dikembangkan menggunakan dataset gambar yang diperoleh dari Kaggle. Penekanan utama dari penelitian ini adalah untuk mengembangkan arsitektur deteksi kecelakaan dengan beban komputasi ringan, sehingga eksperimen yang mempengaruhi beban komputasi akan dilaksanakan menggunakan metrik seperti waktu komputasi model.

A. AKUISISI DATA

Tabel I berisi informasi mendetail mengenai dataset yang digunakan pada penelitian. Pengembangan deteksi objek YOLOv8 menggunakan data rekaman CCTV lalu lintas Indonesia. Rekaman CCTV yang terdapat insiden lalu lintas sebagai data validasi model akan diperoleh dari Twitter. Sistem CCTV lalu lintas di Indonesia umumnya menyimpan gambar dalam format yang umum digunakan seperti JPEG dan memiliki resolusi 1920×1080 piksel. Gambar-gambar yang ditangkap memiliki kedalaman warna sebesar 24 bit per piksel (bpp) dan rasio aspek standar 16:9.

TABEL I
INFORMASI MENGENAI DATASET YANG DIGUNAKAN

Sumber	Jumlah	Ukuran	Format
Kaggle	989	284 MB	JPEG
Twitter	6	22 MB	MP4
CCTV Indonesia	13	48 MB	MP4

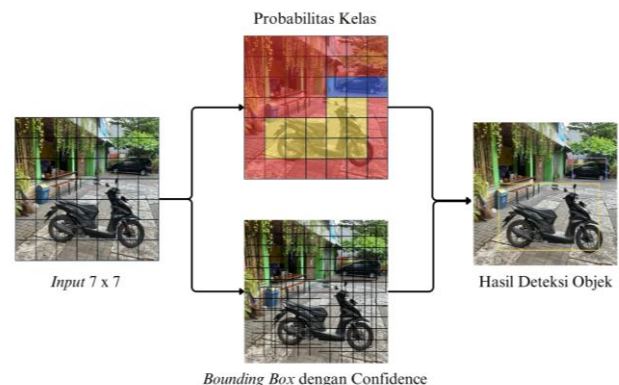
B. INDIKATOR KECELAKAAN PADA REKAMAN CCTV

Kecelakaan dapat dibagi menjadi kecelakaan tunggal dan kecelakaan ganda. Perbedaan yang terdapat pada kedua jenis kecelakaan tersebut ada pada jumlah aktor yang terlibat. Pada

kecelakaan tunggal, terdapat objek kendaraan yang memperlihatkan pola abnormal. Abnormalitas tersebut dapat berupa berhenti mendadak atau berhenti di tempat yang sama dalam rentang waktu lama. Pada kecelakaan ganda, aktor yang terlibat berjumlah lebih dari satu. Dalam kasus ini, selain adanya anomali dalam pergerakan objek, dapat ditemui juga adanya kontak fisik antara dua atau lebih objek kendaraan. Akibat lain dari kontak fisik tersebut adalah perubahan kecepatan secara drastis. Pada kedua tipe kecelakaan terjadi perubahan drastis pada karakteristik gerakan objek. Perubahan drastis inilah yang menjadi indikator untuk mendeteksi adanya kecelakaan dari rekaman CCTV.

C. DETEKSI OBJEK MENGGUNAKAN YOLOV8

Identifikasi objek pada setiap frame video akan dilakukan dengan menggunakan model YOLOv8 [9]. Model ini dapat memperoleh informasi tentang lokasi dan ukuran objek pada setiap frame. YOLOv8 terdiri dari arsitektur CNN yang melakukan proses deteksi objek dengan membagi gambar menjadi grid dengan sel-sel yang pada umumnya berdimensi 7×7 atau 13×13. Setiap sel akan melakukan prediksi jumlah nilai yang berada di dalam *bounding box*. Setiap *bounding box* direpresentasikan dengan lima elemen, yang pertama adalah koordinat titik tengah dari *bounding box* (x, y), diikuti oleh elemen kedua dan ketiga yang masing-masing merepresentasikan lebar dan tinggi, keempat adalah skor kepercayaan yang mengindikasikan keberadaan objek di dalam *bounding box*, dan yang kelima adalah probabilitas kelas yang merepresentasikan kemungkinan keberadaan setiap kelas di dalam *bounding box*.



Gambar 2. Prosedur YOLOv8 dalam mendeteksi objek.

Gambar 2 adalah proses deteksi objek pada model YOLOv8. Pada saat proses pembelajaran, YOLO mengalami optimasi menggunakan fungsi loss yang menggabungkan *localization loss* dan *classification loss* [10]. Dalam deteksi objek, terdapat dua jenis *loss function* yang digunakan untuk melatih model. *Localization loss* mengukur perbedaan antara prediksi lokasi objek dengan lokasi sebenarnya dalam *bounding box*, sementara *classification loss* mengukur perbedaan antara prediksi kelas objek dengan nilai probabilitas kelas yang sebenarnya. Penggabungan kedua *loss function* dan pengurangan total *loss* akan membuat model dapat belajar untuk mendeteksi objek dengan akurasi tinggi. Pada proses prediksi video resolusi rendah, filter Kalman akan melakukan kalkulasi matematis untuk memprediksi aksi selanjutnya. Kalkulasi prediksi filter Kalman tertulis pada (1).

$$\hat{X}_{k|k-1} = F_k X_{k-1|k-1} + B_k u_k \quad (1)$$

Dimana pada suatu sistem, $X_{k|k-1}$ adalah prediksi pada waktu k , F_k adalah bentuk model pada transisi selanjutnya, dan B_k adalah input kontrol pada sistem U_k yang merupakan variabel kontrol [11]. Hasil dari kalkulasi prediksi akan menjadi input pada tahap pembaharuan yang mengubah parameter pada sistem. Proses ini berlangsung secara sekuensial pada jarak waktu yang ditentukan.

D. OBJECT TRACKING

Setelah objek terdeteksi pada frame pertama, pelacakan objek dilakukan untuk mengikuti pergerakan objek pada frame-frame berikutnya. Dalam proses pelacakan objek, perubahan posisi objek dihitung dengan membandingkan posisi *bounding box* objek pada frame saat ini dengan frame sebelumnya. Hal ini dilakukan untuk memantau pergerakan objek dari waktu ke waktu dengan cara mengidentifikasi objek menggunakan YOLOv8 dan melacaknya pada setiap frame menggunakan ByteTrack.



Gambar 3. Proses pelacakan objek menggunakan ByteTrack.

Pada Gambar 3, algoritma ByteTrack menggunakan semua kotak deteksi yang ada, baik dengan skor tinggi maupun rendah. Kotak-kotak deteksi ini dipisahkan menjadi dua kelompok, yaitu kotak dengan skor tinggi dan kotak dengan skor rendah. Kotak-kotak ini dibedakan dengan warna *bounding box*, dimana warna biru menandakan skor tinggi ($0.5 <$), warna merah dan kuning menandakan skor rendah (< 0.5).

E. DETEKSI ANOMALI VIDEO

Pada proses deteksi anomali, perubahan posisi objek dihitung dan dibandingkan dengan nilai *threshold* tertentu. Pendeteksian anomali dilakukan menggunakan algoritma *Simple Moving Average* (SMA). Algoritma ini akan mengkalkulasi rata-rata dari beberapa titik data selama jangkauan waktu yang telah ditentukan [12]. SMA memiliki ekspresi matematis pada (2) sebagai berikut:

$$SMA(t) = \frac{(d_{(t-n+1)} + d_{(t-n+2)} + \dots + d_t)}{n} \quad (2)$$

Pada konteks deteksi kecelakaan, nilai SMA adalah laju rata-rata objek-objek pada gambar. Sebagai contoh, kendaraan yang melaju akan dikalkulasi pergerakannya dari frame ke frame dan diambil rata-rata lajunya. Dengan ini, kendaraan yang berubah arah belum tentu menjadi anomali karena masih memiliki laju yang konstan. *Threshold* akan ditentukan dari laju rata-rata objek pada frame. Dimana SMA akan dikalkulasi pada setiap frame t , dan $t \geq n$ untuk titik data n . Untuk deteksi anomali, perpindahan dt dengan SMA pada frame t akan

diperbandingkan. Apabila perbedaan absolut melebihi *threshold* yang ditentukan oleh algoritma, anomali akan terdeteksi. Pergerakan objek yang melebihi *threshold* bisa berjumlah lebih dari satu. Anomali-anomali yang terdeteksi akan disimpan oleh model pada sebuah list dan anomali dengan nilai maksimum akan dipilih untuk menjadi input klasifikasi menggunakan model *MobileNetv2*.

F. KLASIFIKASI MENGGUNAKAN MODEL MOBILENETV2

Untuk mengklasifikasi anomali yang terdeteksi, model *MobileNetv2* akan dikembangkan menggunakan data yang diperoleh dari Kaggle. Arsitektur dari *MobileNetv2* meliputi *depthwise separable convolution*, yang mengaplikasikan filter spasial terhadap kanal input secara terpisah dan mengambil informasi spasial lokal sehingga mengurangi jumlah parameter pembelajaran yang dibutuhkan [8]. Alhasil, beban komputasi model *MobileNetv2* lebih rendah daripada model CNN tradisional lainnya [13]. Arsitektur *MobileNetv2* juga mengimplementasikan residual terbalik dengan *bottlenecks* linear. Lapisan *bottleneck* adalah lapisan *convolutional* 1×1 yang menahan jumlah fitur, sedangkan *bottlenecks* linear adalah penggunaan fungsi aktivasi linear pada lapisan *bottleneck*, sehingga mengurangi kompleksitas model tanpa mengurangi akurasi.

G. PERHITUNGAN BEBAN KOMPUTASI

Setelah seluruh proses dilakukan, beban komputasi arsitektur yang dikembangkan akan dikalkulasi. Beban komputasi diukur oleh kekuatan proses, penggunaan memori, atau *bandwidth* jaringan [14]. Pada umumnya, task yang memiliki kompleksitas tinggi akan memiliki beban komputasi yang berat dan sebaliknya, task yang memiliki kompleksitas rendah akan lebih cepat untuk dieksekusi karena memiliki beban komputasi ringan dan lebih mudah dalam pemeliharaan (*Maintenance*). Beban komputasi dari arsitektur pada penelitian ini diperhitungkan untuk konsiderasi implementasi deteksi kecelakaan secara real-time. Rata-rata waktu proses per objek pada frame akan dihitung untuk mendapatkan informasi beban komputasi dari arsitektur yang dikembangkan.

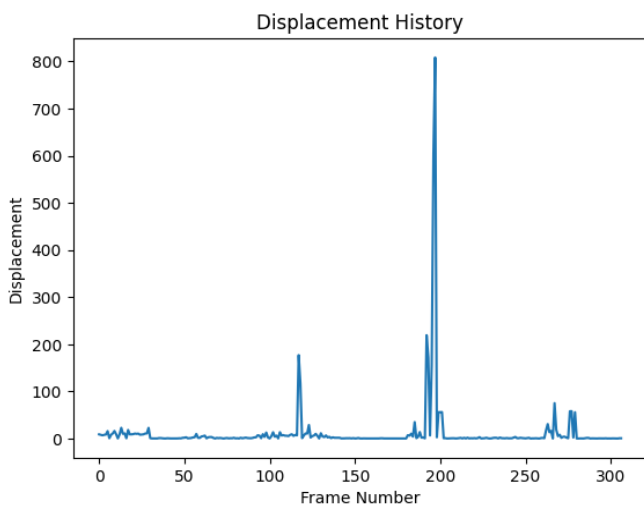
III. HASIL DAN PEMBAHASAN

Model *pre-trained* YOLOv8 yang telah dilatih menggunakan gambar-gambar kendaraan mampu mengklasifikasi setiap objek pada dataset CCTV Indonesia seperti kendaraan mobil dan sepeda motor dengan *confidence rate* mencapai 0.90. Gambar 4 adalah demonstrasi deteksi objek oleh arsitektur YOLO V8.

Anomali yang terdapat pada dataset video insiden lalu lintas terdeteksi oleh penggunaan metode SMA. Gambar 5 memperlihatkan histogram yang mencontohkan hasil implementasi metode SMA pada sebuah video yang memiliki insiden lalu lintas. Histogram tersebut adalah grafik angka pergerakan y terhadap jumlah frame x . Anomali ditentukan oleh keberadaan nilai pergerakan abnormal yang terdapat pada jarak jumlah frame tertentu. Sebaliknya, pada video yang tidak memiliki anomali, histogram yang dihasilkan akan berupa grafik yang memiliki nilai y yang relatif mendatar dikarenakan pada setiap frame, perpindahan yang terjadi memiliki nilai delta dibawah *threshold*.



Gambar 4. Hasil deteksi objek menggunakan YOLO V8



Gambar 5. Histogram metode SMA pada video yang terdapat insiden

Gambar 6 memperlihatkan hasil dari deteksi anomali pada rekaman CCTV yang terdapat kecelakaan lalu lintas. Pada saat insiden, terjadi perpindahan signifikan yang melebihi *threshold* (pergerakan objek pada beberapa frame sangat tinggi), sehingga anomali terdeteksi oleh algoritma SMA.



Gambar 6. Deteksi anomali pada rekaman CCTV kecelakaan lalu lintas

Dari frame-frame yang terdapat anomali akan diambil frame dengan nilai anomali maksimum. Frame tersebut akan menjadi input pada klasifikasi menggunakan model *MobileNetv2* yang telah dilatih. Model *MobileNetv2* akan memprediksi frame dengan anomali maksimum dan mengklasifikasi frame sebagai kecelakaan/tidak kecelakaan. Gambar 7 memperlihatkan contoh

frame-frame yang memiliki anomali dan hasil prediksi dari model *MobileNetv2*.



Gambar 7. Hasil klasifikasi kecelakaan menggunakan *MobileNetv2*

Performa dari proses klasifikasi *MobileNetv2* menghasilkan angka pada sejumlah metrik yang ditentukan. Untuk nilai *Precision* dari model adalah 0.97 pada kelas kecelakaan dan 0.84 pada kelas tidak kecelakaan. Nilai *Recall* dari model adalah 0.79 untuk kelas kecelakaan dan 0.90 pada kelas tidak kecelakaan. *F-Score* dari model memberikan nilai 0.87 untuk kelas kecelakaan dan 0.90 pada tidak kecelakaan. Nilai *Accuracy* dari model adalah 0.89. Tabel II memperlihatkan performa dari model secara mendetail..

TABEL II
HASIL DARI KLASIFIKASI *MOBILENETV2* TERHADAP ANOMALI

Kelas	Precision	Recall	F-Score	Accuracy
Kecelakaan	0,97	0,79	0,87	0,89
Tidak Kecelakaan	0,84	0,98	0,90	

Iluminasi dan kualitas rekaman mempengaruhi hasil deteksi anomali dikarenakan kondisi hujan atau resolusi video rendah memperburuk proses deteksi anomali. Angka *accuracy* juga terpengaruh oleh proses *training* model yang mempelajari data dengan kondisi pencahayaan dan kualitas berbeda dari frame-frame pada proses deteksi anomali. Faktor resolusi area target dari frame anomali juga mempengaruhi proses prediksi karena area target pada frame anomali memiliki area target klasifikasi kecil dibandingkan dengan target pada data pada tahap training model.

Pada Tabel III, rata-rata waktu per frame merepresentasikan waktu yang diperlukan model untuk memproses setiap frame. Pada video 1 terdapat 135 frame, total penggunaan waktu dari model untuk memproses video adalah 3 detik sehingga setiap frame dibutuhkan waktu 0.022 detik atau 22 ms per frame. Video 2 dengan frame berjumlah 116 frame membutuhkan waktu proses 2.4 detik sehingga waktu yang dibutuhkan untuk memproses setiap frame adalah 20.87 ms per frame. Video 3 yang memiliki 510 frame memiliki waktu proses 10 detik sehingga waktu proses per frame adalah 20.18.

TABEL III
KALKULASI BEBAN KOMPUTASI PROSES DETEKSI KECELAKAAN

Video	Beban Komputasi Deteksi Objek	Deteksi Objek + Deteksi Anomali	Rata-rata beban komputasi model
Video 1 (135 frame)	13,42 ms/frame	22,23 ms/frame	21 ms/frame
Video 2 (116 frame)	14,82 ms/frame	20,87 ms/frame	
Video 3 (510 frame)	13,87 ms/frame	20,18 ms/frame	

TABEL IV
PERBANDINGAN BEBAN KOMPUTASI DENGAN METODE LAIN

Metode	Beban Komputasi	Akurasi
[15]	24 ms/frame	0,93
[4]	46,1 ms/frame	0,90
Metode yang diusulkan	21 ms/frame	0,89

Dari waktu proses per frame setiap video, diambil rata-rata waktu proses 21 ms/frame. Apabila dilakukan komparasi dengan model-model yang dikembangkan peneliti lain, model dari penelitian ini memiliki waktu proses lebih cepat. Model R-CNN dari [15] memiliki beban komputasi 24 ms/frame dan nilai akurasi 0.93, sedangkan metode R-CNN yang dikembangkan oleh [4] memiliki beban komputasi 46.1 ms/frame dengan akurasi 0.90. Tabel IV memperlihatkan perbandingan beban komputasi dan akurasi antara metode yang diusulkan dan metode lain. Metode yang diusulkan berhasil memiliki beban komputasi lebih rendah dari metode peneliti lain dengan akurasi yang kompetitif. Hal ini disebabkan oleh kompleksitas model penelitian dari [15] dan [4] yang lebih tinggi, sehingga akurasi yang dihasilkan lebih tinggi dengan beban komputasi yang lebih besar.

IV. KESIMPULAN

Penelitian ini membangun model deteksi objek dengan menggunakan arsitektur YOLOv8 dan *MobileNetv2* dengan tujuan mendeteksi kecelakaan lalu lintas dengan beban komputasi ringan. Konklusi yang didapatkan dari penelitian ini adalah sebagai berikut:

- Model yang diusulkan mampu mendeteksi frame yang memiliki anomali dengan metode *Simple Moving Average* (SMA).
- Frame yang terdeteksi memiliki anomali diklasifikasikan sebagai kecelakaan atau tidak kecelakaan dengan model *MobileNetv2*. Model *MobileNetv2* berhasil memprediksi anomali dengan akurasi bernilai 0.89.
- Model ini memiliki waktu proses rata-rata 21 ms/frame sehingga bisa disimpulkan memiliki beban komputasi ringan apabila dibandingkan oleh metode-metode yang dikembangkan oleh peneliti lain.

Dari limitasi penelitian ini, peneliti lain yang mempunyai intensi untuk mengembangkan metode ini disarankan untuk meningkatkan akurasi klasifikasi dengan menggunakan model

CNN *pre-trained* lain yang memiliki waktu proses lebih cepat dan beban komputasi yang ringan. Beban komputasi model juga bisa dikurangi dengan cara melakukan data pre-processing seperti segmentasi data dan pemrosesan frame video. Dengan menurunkan laju frame, proses deteksi objek akan mengalami pertambahan kecepatan dan menurunkan beban komputasi. Hasil penelitian ini dapat menjadi pertimbangan untuk mengembangkan model-model lain yang mempunyai performa lebih baik, sehingga arsitektur deteksi kecelakaan melalui siaran CCTV secara *real-time* dapat dikembangkan.

REFERENSI

- [1] M. D. Astuti, Sutrisno, and Rahmawati, "Hubungan Respon Time Perawat dengan Penanganan Nyeri pada Pasien Fraktur Akibat Kecelakaan Lalu Lintas di IGD RSUD Dr. Raden Soedjati Soemodirdjo Purwodadi," *Journal of TSCNers*, vol. 8, no. 1, 2023. e-ISSN: 2503-2453.
- [2] T. Tamagusko, M. G. Correia, M. A. Huynh, and A. Ferreira, "Deep Learning applied to Road Accident Detection with Transfer Learning and Synthetic Images," *Transportation Research Procedia*, vol. 64, pp. 90-97, 2022. <https://doi.org/10.1016/j.trpro.2022.09.012>.
- [3] Y. Zhang et al., "ByteTrack: Multi-Object Tracking by Associating Every Detection Box," 2022.
- [4] D. Tian, C. Zhang, X. Duan, and X. Wang, "An Automatic Car Accident Detection Method Based on Cooperative Vehicle Infrastructure Systems," *IEEE Access*, vol. PP, no. 1, pp. 1-1, 2019. <https://doi.org/10.1109/ACCESS.2019.2939532>.
- [5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 779-788.
- [6] D. Reis, J. Kupec, J. Hong, and A. Daoudi, "Real-Time Flying Object Detection with YOLOv8," *arXiv preprint arXiv:2305.09972*, 2023.
- [7] A. Howard et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *arXiv preprint arXiv:1704.04861*, 2017.
- [8] M. Sandler et al., "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4510-4520. <https://doi.org/10.1109/CVPR.2018.00474>.
- [9] J. Wu, "Complexity and accuracy analysis of common artificial neural networks on pedestrian detection," *MATEC Web of Conferences*, vol. 232, 01003, 2018. doi:10.1051/mateconf/201823201003.
- [10] A. Bochkovskiy, C. Wang, and H. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *ArXiv*, abs/2004.10934, 2020.
- [11] D. Kalita and P. Lyakhov, "Moving Object Detection Based on a Combination of Kalman Filter and Median Filtering," *Big Data and Cognitive Computing*, vol. 6, no. 4, p. 142, 2022. doi:10.3390/bdcc6040142.
- [12] J. D. Hamilton, *Time Series Analysis*, Vol. 2. Princeton, NJ: Princeton University Press, 1994.
- [13] Y. Gulzar, "Fruit Image Classification Model Based on MobileNetV2 with Deep Transfer Learning Technique," *Sustainability*, vol. 15, p. 1906, 2023. <https://doi.org/10.3390/su15031906>.
- [14] S. Arora and B. Barak, *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.
- [15] S. Robles-Serrano, G. Sanchez-Torres, and J. Branch-Bedoya, "Automatic Detection of Traffic Accidents from Video Using Deep Learning Techniques," *Computers*, vol. 10, no. 11, p. 148, 2021. <https://doi.org/10.3390/computers10110148>.