

## DETEKSI HELM PADA PENGGUNA SEPEDA MOTOR MENGUNAKAN METODE YOLO SECARA REALTIME

Bayu Charisma Putra<sup>1</sup>, Muhammad Ilhamul Khoir<sup>2</sup>

<sup>1,2,3</sup>Teknik Informatika, Fakultas Teknik, Universitas Maarif Hasyim Latif, Sidoarjo, Indonesia

[bayu\\_charisma@dosen.umaha.ac.id](mailto:bayu_charisma@dosen.umaha.ac.id)<sup>1</sup>, [ilhammuhammad741@gmail.com](mailto:ilhammuhammad741@gmail.com)<sup>2</sup>

Naskah diterima: 26 Nopember 2025 ; Direvisi : 01 Desember 2025 ; Disetujui : 09 Desember 2025

### Abstrak (Indonesia)

Helm adalah salah satu perlengkapan wajib yang dikenakan pengendara sepeda motor, helm berguna melindungi kepala dari benturan yang mungkin saja bisa terjadi pada pengendara sepeda motor. Semakin maraknya pelanggaran lalu lintas terutama pengguna sepeda motor yang tidak menggunakan helm dan pemantauannya secara manual langsung ke jalan, semakin dirasakan kebutuhan adanya sistem otomatis yang digunakan untuk memantau agar lebih mudah dan cepat. Dikarenakan masalah tersebut, maka didalam Tugas Akhir ini tergasalah sebuah sistem untuk mengklasifikasi penggunaan helm menggunakan webcam. Sistem ini dibangun menggunakan metode YOLO (You Only Look Once) untuk pengenalan objek pengendara motor dan penggunaan helm pada pengguna sepeda motor secara realtime dengan bantuan library OpenCV. Sebelum proses deteksi, dibutuhkan pengumpulan dataset untuk anotasi/labeling dan ditraining menggunakan YOLOv4-Tiny. Dalam tahap pengujian, sistem ini nantinya akan mendeteksi objek yang ada di jalan menggunakan webcam, objek itu nantinya akan dicocokkan dengan data training yang sudah disiapkan, kalau nantinya objek yang dideteksi memiliki kecocokan dengan hasil training, maka objek akan menampilkan label serta keterangannya.

**Kata kunci:** deteksi, helm, pengendara motor, real-time, yolo

### Abstract (English Version)

*A helmet is one of the mandatory safety equipment worn by motorcycle riders, serving to protect the head from potential impacts. The increasing number of traffic violations, especially among motorcyclists who do not wear helmets, along with the limitations of manual on-road monitoring, has created a growing need for an automated monitoring system that is faster and more efficient. Based on this issue, this final project proposes a system for classifying helmet usage using a webcam. The system is developed using the YOLO (You Only Look Once) method for real-time object detection of motorcycle riders and helmet usage, supported by the OpenCV library. Prior to the detection process, a dataset must be collected for annotation/labeling and trained using YOLOv4-Tiny. In the testing phase, the system detects objects on the road through a webcam and matches them with the prepared training data. If the detected object corresponds to the training results, the system will display the appropriate label and information.*

**Keywords:** detection, helmet, motorcycle rider, real-time, yolo,

## PENDAHULUAN

Helm adalah salah satu perlengkapan wajib yang dikenakan pengendara sepeda motor, helm berguna melindungi kepala dari benturan yang mungkin saja bisa terjadi pada pengendara sepeda motor, di Indonesia telah dibuat Undang-undang tentang kewajiban memakai helm bagi pengendara sepeda motor yang tertuang dalam Undang-undang No. 22 tahun 2009 Tentang Lalu Lintas Dan Angkutan Jalan pasal 106 ayat 8 mensyaratkan bagi semua pengendara sepeda motor dan penumpangnya untuk memakai helm yang memenuhi standar nasional Indonesia.

Sejalan dengan makin maraknya pelanggaran lalu lintas terutama pengguna sepeda motor yang tidak menggunakan helm, semakin dirasakan kebutuhan adanya sistem otomatis yang digunakan untuk memantau para pelanggar dan untuk mempercepat menindak para pelanggar lalu lintas yang tidak menggunakan helm dan juga mengurangi jumlah manusia sebagai sumber daya yang dibutuhkan.

Pada penelitian sebelumnya yang dilakukan oleh (Yusuf 2020) dan (Albert, kartika & endang 2020). Pada penelitian ini menggunakan Metode CNN (Convolutional Neural Network) dan YOLO, hasil yang didapatkan menunjukkan performansi yang baik, dan pada penelitian ini datanya adalah gambar kemudian di inputkan ke dalam program dan menampilkan hasil deteksi

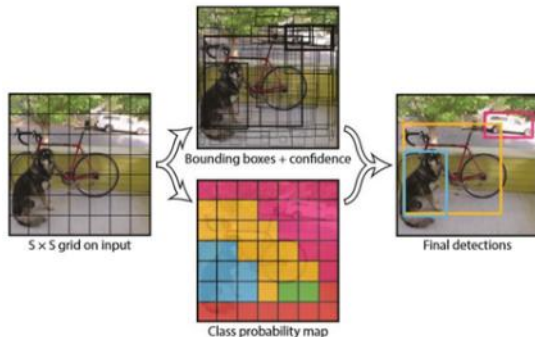
tersebut.

Dari penelitian tersebut bisa dikembangkan lagi dengan pendeteksian secara langsung atau object tracking secara realtime menggunakan webcam tanpa harus mengambil gambar dari rekaman dan menginputkan gambar terlebih dahulu, dan juga penambahan dataset untuk meningkatkan akurasi deteksi penggunaan helm. Sistem ini dibangun menggunakan metode YOLO (You Only Look Once) untuk pengenalan objek pengendara motor dan penggunaan helm pada pengguna sepeda motor secara realtime dengan bantuan library OpenCV. Sebelum proses deteksi, dibutuhkan pengumpulan dataset untuk anotasi/labeling dan ditraining, kemudian di uji coba dan jika terdapat objek yang terdeteksi maka akan menampilkan label serta keterangannya.

## METODE

YOLO (You Only Look Once) merupakan algoritma deep learning yang memanfaatkan jaringan syaraf konvolusional (CNN) dalam mendeteksi objek secara realtime. Algoritma ini akan membagi citra ke dalam grid berukuran  $s \times s$  yang kemudian pada tiap grid akan memprediksi bounding box serta peta kelas masing-masing grid. Apabila pada satu grid terprediksi objek, maka pada grid tersebut akan diprediksi bounding box yang mengelilingi objek tersebut. Nilai confidence akan dihitung pada

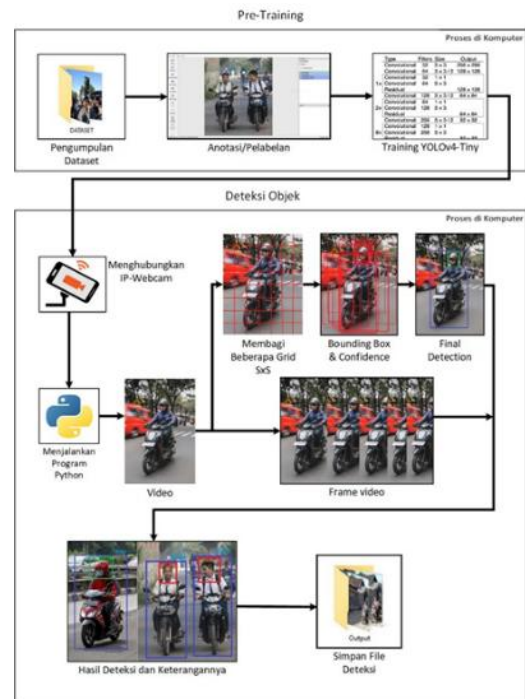
masing-masing bounding box yang kemudian akan diseleksi berdasarkan nilai yang didapat. (Redmon, Divvala, Girshick, & Farhadi, 2016)



Gambar 1. You only look once

You Only Look Once (YOLO) adalah sebuah algoritma yang dikembangkan untuk mendeteksi sebuah objek secara real-time. Sistem pendeteksian yang dilakukan adalah dengan menggunakan repurpose classifier atau localizer untuk melakukan deteksi. Sebuah model diterapkan pada sebuah citra di beberapa lokasi dan skala. Daerah dengan citra yang diberi score paling tinggi akan dianggap sebagai sebuah pendeteksian.

Berikut merupakan blok diagram keseluruhan sistem deteksi helm menggunakan metode YOLO



Gambar 2. Blok diagram sistem

Pada Gambar 2 merupakan tahapan-tahapan perancangan secara umum ini dimulai dari pengumpulan data citra, kemudian dilakukan proses anotasi/pelabelan. Setelah dilakukan proses training konfigurasi jaringan YOLOv4-Tiny sesuai dengan data citra dan dilatih sehingga model baru YOLO terbentuk. Pada tahapan deteksi objek yaitu menghubungkan terlebih dahulu kamera dengan laptop menggunakan IP-Webcam dan menjalankan program python, saat itu juga pre-trained dijalankan dan setiap frame-frame video berjalan dilakukan prediksi bounding box dan nilai confidence, jika ada objek yang terdeteksi maka akan keluar output berupa bounding box dan serta keterangannya

Data diperoleh dengan cara merekam pengendara motor di samping jalan menggunakan kamera webcam. Setelah

memperoleh Data citra tersebut dari kamera webcam kemudian diambil frame-frame dari video tersebut sehingga menjadi beberapa foto dengan format .jpg dan disimpan kedalam folder masing-masing kategori.

Selain pengambilan data dari webcam, penelitian ini juga mengambil data citra helmet, non-helmet dan rider dengan mendownload gambar di google image menggunakan Google Chrome dengan bantuan extension Download All Image. Pada penelitian ini semakin banyak jumlah data maka kemungkinan besar objek yang diinginkan dapat terdeteksi dengan optimal.

Pada proses pelabelan ini dibagi menjadi tiga kelas yaitu Menggunakan Helm (Helmet), Tanpa Menggunakan Helm (Non Helmet), Pengendara Motor (Rider). Tujuan dari pelabelan adalah untuk membuat bagian yang termasuk dalam kelas yang disiapkan. Pelabelan gambar menggunakan LabelImg, yang merupakan salah satu modul dalam Python.

Setelah pelabelan selesai, hasil dari labelling citra akan menghasilkan .txt file yang berisi koordinat pada citra dan siap untuk diproses training, dengan keterangan [class, x center, y center, width, height], contoh : [0 0.364075 0.331590 0.061054 0.112971]

Pada proses training ini menggunakan YOLOv4-Tiny dan training ini dilakukan di google colab, sebelum melakukan proses training harus mengatur Hyperparameter YOLO yang disesuaikan

dengan jumlah data dan kebutuhannya, berikut macam-macam dan rumusnya:

1. Batch size, merupakan variabel yang menentukan seberapa banyak image atau training data yang dimasukkan saat training. Semakin kecil nilai batch size yang digunakan, maka proses training semakin cepat.
2. Subdivisions, membagi nilai batch menjadi lebih kecil lagi, dan dapat disebut dengan mini-batch. Jika menggunakan nilai batch sebesar 64 dan dibagi dengan 8 subdivisions, maka menghasilkan nilai 8 yang berarti dilakukan proses training untuk 8 images tiap mini-batchnya. Proses ini berlangsung selama delapan kali hingga proses training pada satu batch tersebut selesai. Proses subdivisions bertujuan untuk mempercepat proses training sekaligus meningkatkan akurasi dengan bantuan GPU.
3. Width, merupakan resolusi lebar dataset saat ditraining, biasanya menggunakan nilai 416.
4. Height, merupakan resolusi tinggi dataset saat ditraining, biasanya menggunakan nilai 416.
5. Max Batches, merupakan banyaknya iterasi pada proses training data. Semakin tinggi nilai max batch, maka sistem akan semakin banyak mempelajari data training. Nilai max batches perlu disesuaikan dengan

jumlah kelas dari objek yang akan dideteksi dengan rumus :

$$\text{maxbatches} = \frac{\text{jumlahclass}}{\times 2000} \quad (1)$$

6. Steps dengan rumus :

$$\text{steps} = (80\% \text{maxbatch}), (90\% \text{maxbatch}) \quad (2)$$

7. Filter dengan rumus :

$$\text{filters} = (\text{jumlahkelas} + 5) \times 3 \quad (3)$$

8. Classes, merupakan banyak kelas objek yang akan ditraining.

Pada proses deteksi pertama yaitu menghubungkan webcam dengan laptop, deteksi objek ini dilakukan menggunakan kamera HP yang digunakan sebagai webcam dengan bantuan aplikasi IP-Webcam, nantinya webcam tersebut akan ditempatkan di samping jalan. Kemudian menjalankan program deteksi berbasis python yang sudah dihubungkan dengan file hasil training YOLOv4-Tiny sebelumnya dan saat setiap frame-frame video berjalan dilakukan proses deteksi YOLOv4-Tiny seperti membagi ke dalam grid SxS, prediksi bounding box & confidence, dan final detection, jika ada objek yang terdeteksi maka akan keluar output berupa bounding box dan serta keterangannya yang kemudian akan disimpan berupa gambar dan video dalam folder output.

## HASIL DAN PEMBAHASAN

### 1. Pengujian Objek

Pengujian objek dilakukan agar

mengetahui akurasi dari model yang telah dilatih. Setiap video akan memiliki status apakah terdeteksi atau tidak yang gambarkan kotak pembatasan pada objek pengendara motor dan setiap objek yang terdeteksi akan memiliki nilai confidence atau nilai kepercayaan.

Pada pengujian kali ini dibagi menjadi 2, pengujian objek menggunakan helm dan pengujian objek tidak menggunakan helm. Untuk proses pengujian ini menggunakan laptop dengan spesifikasi pada Tabel 1

**Tabel 1.** Spesifikasi laptop yang digunakan

Merk	ASUS X441U
Processor	Intel(r) Core(TM) i3-6006U CPU @ 2.00GHz
RAM	4 GB DDR4
Storage	HDD 500GB
Graphics card	Nvidia GeForce 920 MX
Operating system	Windows 10 64-bit

### 2. Pengujian Objek Menggunakan Helm

Pada pengujian ini kamera diletakkan sejajar dengan kepala dan proses deteksi helm dilakukan pada bagian depan objek dengan jarak sekitar 1 meter dan objek diam tidak berjalan/bergerak. Untuk pengujian kelas Helmet ini ada jumlah 6 objek yang diteliti dan hasil deteksi dijelaskan pada Tabel 2

**Tabel 2.** Hasil pengujian objek menggunakan helm

No	Gambar	Hasil/Keterangan
1.		Terdeteksi dengan benar, objek menggunakan helm berwarna abu-abu

2.		Terdeteksi dengan benar, objek menggunakan helm berwarna hitam
3.		Terdeteksi dengan benar, objek menggunakan helm berwarna biru
4.		Terdeteksi dengan benar, objek menggunakan helm berwarna hitam
5.		Terdeteksi dengan benar, objek menggunakan helm berwarna merah
6.		Terdeteksi dengan benar, objek menggunakan helm berwarna putih

Berdasarkan hasil pengujian yang telah dilakukan pada Tabel 2. Maka dapat dikatakan dapat mendeteksi secara baik dan benar, hal ini dibuktikan dengan mendeteksi semua objek yang menggunakan helm dan nilai tingkat keberhasilan yang menunjukkan 100%.

### 3. Pengujian Objek Tidak Menggunakan Helm

Pada pengujian ini dilakukan untuk mengetahui tingkat keberhasilan sistem dalam mendeteksi objek tanpa menggunakan helm dengan objek diam. Untuk pengujian kelas Non Helmet ini ada jumlah 8 objek yang di teliti dan hasil deteksi dijelaskan pada Tabel 3.

**Tabel 3.** Hasil pengujian objek tidak menggunakan helm

No	Gambar	Hasil/ Keterangan
----	--------	-------------------

1.		Terdeteksi benar, objek tidak menggunakan helm dengan kondisi tanpa aksesoris
2.		Terdeteksi benar, objek tidak menggunakan helm dengan kondisi tanpa aksesoris
3.		Terdeteksi benar, objek tidak menggunakan helm dengan kondisi menggunakan jilbab
4.		Terdeteksi benar, objek tidak menggunakan helm dengan kondisi menggunakan jilbab
5.		Terdeteksi salah, objek menggunakan jaket warna merah, bulat menutupi kepala yang menyerupai helm
6.		Terdeteksi benar, objek tidak menggunakan helm dengan kondisi menggunakan masker
7.		Terdeteksi benar, objek tidak menggunakan helm dengan kondisi menggunakan masker
8.		Terdeteksi salah, objek memakai topi putih menutupi kepala yang menyerupai helm

Untuk pengujian pada kelas Non Helmet ini ada jumlah 8 objek yang di teliti untuk mengetahui tingkat akurasi. Berdasarkan hasil pengujian yang telah dilakukan pada Tabel 3. Maka dapat dikatakan dapat mendeteksi cukup baik, ada 2 objek yang tidak dapat terdeteksi dengan baik dikarenakan objek menggunakan jaket warna merah, bulat menutupi kepala dan objek memakai topi putih yang menutupi bagian atas kepala dan menyerupai helm, dari hasil deteksi objek yang tidak menggunakan helm memiliki nilai tingkat



keberhasilan yang menunjukkan 75%.

#### 4. Pengujian Deteksi Objek di Jalan

Tujuan pengujian deteksi objek ini adalah untuk mengetahui tingkat keberhasilan deteksi objek penggunaan helm pada pengendara motor. Pada proses pengujian deteksi objek di jalan, parameter yang digunakan adalah posisi objek yang berjalan ke depan menghadap webcam. Total jumlah objek yang diuji adalah 42 objek yang berbeda, dan akan memiliki status apakah terdeteksi atau tidak yang gambarkan kotak pembatasan pada objek pengendara motor. Setiap objek yang terdeteksi akan memiliki nilai confidence atau nilai kepercayaan.

#### 5. Pengujian Deteksi Objek Menggunakan Helm di Jalan

Pendeteksian ini diambil dari lokasi jalan Ds. Pedagangan, Wringinanom. Dalam pengujian deteksi berdasarkan objek menggunakan helm bertujuan untuk mengetahui keakuratan model dalam mendeteksi kelas Helmet. Pada pengujian kali ini dibagi dengan 4 sesi pengujian.

**Tabel 4.** Hasil deteksi kelas helmet

No	Sesi Pengujian	Jumlah Objek	Objek dikenali
1.	Deteksi 1	5	4
2.	Deteksi 2	6	5
3.	Deteksi 3	4	3
4.	Deteksi 4	3	2
<b>jumlah</b>		<b>18</b>	<b>16</b>

Dari Tabel 4 diketahui bahwa terdapat 18 jumlah objek yang dilakukan

dalam proses pengujian. Dari 18 objek tersebut, 16 objek dapat dikenali oleh sistem deteksi penggunaan helm yang menggunakan metode YOLO. Dari data tersebut dapat dihitung akurasi pengenalan data mencapai 88,8% berdasarkan dari perhitungan berikut :

$$Akurasi = \frac{16}{18} \times 100\% = 88,8\% \quad (4)$$

Hasil akurasi pengujian pada sistem deteksi penggunaan helm yang menggunakan metode YOLO menghasilkan angka yang cukup tinggi. Presentasi ini tidak bisa mencapai nilai maksimal dikarenakan beberapa faktor. Salah satu faktor tersebut adalah oboek berjalan terlalu cepat. Sehingga sistem mengalami kesulitan dalam mendeteksi objek yang menggunakan helm.



Gambar 3. Hasil deteksi objek kelas helmet

#### 6. Pengujian Deteksi Objek Menggunakan Helm di Jalan

Pendeteksian ini diambil dari lokasi jalan Ds. Pedagangan, Wringinanom. Dalam pengujian deteksi berdasarkan objek tidak menggunakan helm bertujuan untuk

mengetahui keakuratan model dalam mendeteksi kelas Non Helmet dalam berbagai macam kondisi. Pada pengujian kali ini dibagi dengan 5 sesi pengujian.

**Tabel 5.** Hasil deteksi kelas non helmet

No	Sesi Pengujian	Jumlah Objek	Objek dikenali
1.	Deteksi 1	5	4
2.	Deteksi 2	4	4
3.	Deteksi 3	6	4
4.	Deteksi 4	3	3
5.	Deteksi 5	6	4
<b>jumlah</b>		<b>24</b>	<b>19</b>

Dari Tabel 5 diketahui bahwa terdapat 24 jumlah objek yang dilakukan dalam proses pengujian. Dari 24 objek tersebut, 19 objek dapat dikenali oleh sistem deteksi penggunaan helm yang menggunakan metode YOLO. Dari data tersebut dapat dihitung akurasi pengenalan data mencapai 79,1% berdasarkan dari perhitungan berikut :

$$Akurasi = \frac{19}{24} \times 100\% = 79,1\% \quad (5)$$

Hasil akurasi pengujian pada sistem deteksi penggunaan helm yang menggunakan metode YOLO menghasilkan angka yang cukup tinggi. Presentase ini tidak bisa mencapai nilai maksimal dikarenakan beberapa faktor yang mempengaruhi. Salah satu faktor tersebut adalah objek berjalan terlalu cepat, sehingga sistem mengalami kesulitan dalam mendeteksi objek yang tidak menggunakan helm.



Gambar 4. Hasil deteksi objek kelas non helmet

## PENUTUP

Dari hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa, Pada pengujian objek di jalan untuk kelas Helmet menggunakan YOLOv4-Tiny dapat mendeteksi cukup baik dengan tingkat akurasi 88,8% dan untuk kelas non helmet dapat mendeteksi cukup baik dengan tingkat akurasi 79,1%. Masing-masing model dapat mendeteksi berbagai variasi pelanggar seperti penggunaan masker, jilbab. Tetapi kurang akurat dalam mendeteksi penggunaan topi dan sistem dapat mendeteksi dengan baik jika objek berada kurang dari 5 meter

Untuk pengembangan lebih lanjut, terdapat beberapa beberapa saran yang dapat dilakukan seperti, Memperbanyak data train pada kelas Non Helmet guna memaksimalkan hasil deteksi terutama untuk memaksimalkan tingkat akurasi YOLOv4-tiny, membuat user interface untuk mempermudah dalam pengoperasian,



memperbanyak variasi karakteristik penggunaan aksesoris pada kepala pengendara yang tidak menggunakan helm dan menggunakan YOLOv4 agar dapat mendeteksi secara tepat, detail dan akurat.

## DAFTAR PUSTAKA

- [1] T. Akhir, Umar Y, Nrp et al., "Deteksi Penggunaan Helm Pada Pengendara," 2020.
- [2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 779–788, 2016, doi: 10.1109/CVPR.2016.91.
- [3] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 6517–6525, 2017, doi: 10.1109/CVPR.2017.690.
- [4] M. Zufar and B. Setiyono, "Convolutional Neural Networks Untuk Pengenalan Wajah Secara Real-Time," *J. Sains dan Seni ITS*, vol. 5, no. 2, p. 128862, 2016, doi: 10.12962/j23373520.v5i2.18854.
- [5] O. E. Karlina and D. Indarti, "Pengenalan Objek Makanan Cepat Saji Pada Video Dan Real Time Webcam Menggunakan Metode You Look Only Once (Yolo)," *J. Ilm. Inform. Komput.*, vol. 24, no. 3, pp. 199–208, 2019, doi: 10.35760/ik.2019.v24i3.2362.
- [6] M. L. Nazilly, B. Rahmat, and E. Y. Puspaningrum, "Implementasi Algoritma Yolo (You Only Look Once) Untuk Deteksi Api," *J. Inform. dan Sist. Inf.*, vol. 1, no. 1, pp. 81–91, 2020.
- [7] K. Gunadi, E. Setyati, and J. S. Surabaya, "Deteksi Helm pada Pengguna Sepeda Motor dengan Metode Convolutional Neural Network," *J. Infra*, vol. 8, no. 1, pp. 295–301, 2020.
- [8] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv*, 2020.
- [9] Alwafi Ridho Subarkah, "Implementasi Deep Learning Untuk Image Classification Menggunakan Algoritma Convolutional Neural Network (Cnn) Pada Citra Wayang Golek," *Nhk技研*, vol. 151, no. 2, pp. 10–17, 2018.
- [10] F. T. Hidayat and A. K. Whardana, "Deteksi Pelanggaran Sepeda Motor Menggunakan Algoritma YOLO dan Mean Average Precision," *J. Sist. Komput. dan Kecerdasan Buatan*, vol. 8, no. 1, 2024.
- [11] B. C. Putra and Y. N. Afifah, "Gaussian Mixture Model untuk Penghitungan Tingkat Kebersihan Sungai Berbasis Pengolahan Citra," *Teknika: Engineering and Sains Journal*, vol. 2, no. 1, pp. 53–58, 2018.
- [12] B. C. Putra, B. Setiyono, D. R. Sulistyaningrum, Soetrisno, and I. Mukhlash, "Moving Vehicle Classification Using Pixel Quantity Based on Gaussian Mixture Models," *Proc. 2018 3rd Int. Conf. Comput. Commun. Syst. (ICCCS)*, pp. 27–30, 2018, doi: 10.1109/CCOMS.2018.8463218.
- [13] B. C. Putra and R. A. J. Firdaus, "Storm Detection Application on Satellite Image Using the Hough Circle Method Based on Digital Image Processing," *J. Inf. Comput. Technol. Educ. (JICTE)*, vol. 4, no. 2, pp. 1–8, 2020, doi: 10.21070/jicte.v4i2.1018.
- [14] T. Puspita, E. R. Swedia, M. Cahyanti, and M. R. D. Septian, "A Real-Time Helmet Detection System Based on YOLOv8 to Support Traffic Law Enforcement," *Sebatik*, vol. 29, no. 1, pp. 1–10, 2025, doi: 10.46984/sebatik.v29i1.2585.
- [15] H. A. Reswara, B. Priyatna, A. Hananto, and T. Tukino, "Implementasi Deteksi Objek Penggunaan Helm Dengan Metode YOLOv10," *J. Minfo Polgan*, vol. 14, no. 1, pp. 1380–1387, 2025, doi: 10.33395/jmp.v14i1.15010.

- [16] A. S. Ramadhan, “Deteksi Penggunaan Helm pada Pengendara Sepeda Motor di Lalu Lintas Menggunakan Algoritma YOLOv8,” *Skripsi*, Institut Teknologi PLN, 2025.
- [17] S. M. Basha and P. Sarangamath, “Intelligent Helmet Detection for Motorcyclists Using YOLO-Based Surveillance,” *Int. J. Res. Appl. Sci. Eng. Technol.*, 2025.
- [18] F. Faturahman, P. E. Yunanto, and M. D. Sulistiyo, “Deteksi Varian Penggunaan Helm dari Kamera Surveilans Menggunakan Metode Berbasis Deep Learning,” *eProceedings of Engineering*, vol. 11, no. 1, 2024.
- [19] R. R. Kumar et al., “Real-Time Multi-class Helmet Violation Detection Using YOLOv8 with License Plate Recognition,” *Proc. Int. Conf. Adv. Comput.*, pp. 350–360, 2024.
- [20] J. Singh and A. K. Sharma, “Edge-Based AI Solution for Urban Safety: Helmet Compliance Monitoring with YOLOv9 on Raspberry Pi,” *Intell. Syst. Rev.*, vol. 4, 2025, doi: 10.1007/s43926-025-00113-9.
- [21] D. A. Pratama, “Implementasi YOLO untuk Deteksi Penggunaan Helm pada Pengendara Sepeda Motor,” *Skripsi*, Institut Teknologi PLN, 2025.