

Model Pelacakan Objek Berbasis YOLO-OCSort Untuk Estimasi Kepadatan Lalu Lintas

Deppi Linda*¹, Nursiyanto², Isnandar Agus³

^{1,2}Sistem Informasi, Ilmu Komputer, Institut Informatika dan Bisnis Darmajaya, Bandar Lampung, Indonesia

³Magister Manajemen Teknologi, Institut Informatika dan Bisnis Darmajaya, Bandar Lampung, Indonesia
Email: deppilinda@darmajaya.ac.id¹, ikinursiyanto@darmajaya.ac.id², isnandaragus@darmajaya.ac.id³

Abstract — A High traffic density poses a major challenge in urban transportation management. Accurate density estimation is crucial for supporting intelligent transportation systems. However, conventional object tracking methods are often ineffective in handling nonlinear motion and temporal occlusion. This research aims to develop a real-time traffic density estimation system based on the integration of YOLOv8 and Online Clustering SORT (OCSort). YOLOv8 is employed as a high-accuracy object detector, while OCSort applies an observation-centric association approach to enhance tracking performance. Evaluation was conducted using the MOT17FRCNN benchmark dataset for tracking accuracy measurement and a subset of UADETRAC for computational efficiency testing. The system was tested on Intel Core i7 Gen-13, 32 GB RAM, and NVIDIA GeForce RTX 3060 12 GB VRAM hardware. Results demonstrate MOTA values of 0.4910, MOTP of 0.9072, IDF1 of 0.6702, with 7 identity switches and 65 fragmentation events. Comparison with state-of-the-art methods shows the best processing time of 118.01 seconds on 2001 frames, outperforming ByteTrack (146.98 s), BoTSORT (161.91 s), and DeepSORT (152.33 s). The system achieves 17.0 FPS with an average processing time of 33.8 ms per frame and detects 20.1 objects per frame with an average confidence score of 0.608. The integration of YOLOv8 and OCSort proves capable of providing a balance between tracking accuracy and computational efficiency for real-time traffic monitoring applications

Keyword — computer vision; multiobject tracking; realtime processing; YOLO-OCSort

Abstrak — Kepadatan lalu lintas yang tinggi menjadi tantangan utama dalam pengelolaan transportasi perkotaan. Estimasi kepadatan secara akurat sangat penting dalam mendukung sistem transportasi cerdas. Namun, metode pelacakan objek konvensional sering kali tidak efektif dalam menghadapi gerakan nonlinier dan oklusi temporal. Penelitian ini bertujuan mengembangkan sistem estimasi kepadatan lalu lintas realtime berbasis integrasi YOLOv8 dan Online Clustering SORT (OCSort). YOLOv8 digunakan sebagai detektor objek berakurasi tinggi, sedangkan OCSort menerapkan pendekatan *observation-centric association* untuk meningkatkan performa pelacakan. Evaluasi dilakukan menggunakan MOT17FRCNN benchmark dataset untuk pengukuran akurasi pelacakan dan subset UADETRAC untuk pengujian efisiensi komputasi. Sistem diuji pada perangkat Intel Core i7 Gen-13, 32 GB RAM, dan GPU NVIDIA GeForce RTX 3060 12 GB VRAM. Hasil menunjukkan nilai MOTA sebesar 0.4910, MOTP 0.9072, IDF1 0.6702, dengan *identity switches* sebanyak 7 dan *fragmentations* sebanyak 65 kejadian. Perbandingan terhadap metode *state-of-the-art* menunjukkan waktu proses terbaik 118.01 detik pada 2001 frame, mengungguli ByteTrack (146.98 s), BoTSORT (161.91 s), dan DeepSORT (152.33 s). Sistem mencapai FPS 17.0 dengan rata-rata waktu proses 33.8 ms per frame dan mendeteksi 20.1 objek per frame dengan *confidence score* rata-rata 0.608. Integrasi YOLOv8 dan OCSort terbukti mampu memberikan keseimbangan antara akurasi pelacakan dan efisiensi komputasi untuk aplikasi monitoring lalu lintas secara *realtime*.

Kata kunci— pelacakan multi objek; pengolahan citra; pemrosesan waktu nyata; YOLO-OCSort

I. PENDAHULUAN

Pertumbuhan urbanisasi yang pesat di berbagai negara telah meningkatkan volume lalu lintas secara signifikan, menciptakan tantangan kompleks dalam manajemen transportasi perkotaan. Kemacetan lalu lintas tidak hanya berdampak pada efisiensi ekonomi, tetapi juga berkontribusi terhadap peningkatan emisi karbon dan penurunan kualitas udara di area metropolitan [1]. Sistem pemantauan lalu lintas yang efektif menjadi kebutuhan krusial untuk memahami pola pergerakan kendaraan, mengoptimalkan pengaturan sinyal lalu lintas, dan mengembangkan strategi manajemen transportasi yang berkelanjutan [2].

Estimasi kepadatan lalu lintas secara realtime memerlukan teknologi yang mampu melakukan deteksi dan pelacakan objek dengan akurasi tinggi. Pendekatan konvensional menggunakan sensor induktif atau

pneumatic tubes memiliki keterbatasan dalam hal fleksibilitas instalasi dan cakupan area pemantauan [3], [4]. Sebaliknya, pendekatan berbasis *computer vision* menawarkan solusi yang lebih adaptable dan *costeffective* untuk aplikasi skala besar [5].

Dalam domain *computer vision*, model *You Only Look Once* (YOLO) telah menunjukkan performa yang impresif untuk tugas *object detection* [6]. YOLOv8, sebagai salah satu iterasi dari arsitektur YOLO, telah mencapai keseimbangan optimal antara akurasi deteksi dan kecepatan *inference*, menjadikannya pilihan populer untuk aplikasi *realtime* [7]. Namun demikian, YOLO pada dasarnya dirancang sebagai *single frame object detector* yang tidak memiliki kemampuan inheren untuk melacak objek antar *frame* secara temporal [8]. Keterbatasan fundamental ini menciptakan gap signifikan dalam aplikasi yang memerlukan kontinuitas informasi objek sepanjang waktu, seperti estimasi kepadatan lalu lintas dan analisis pola pergerakan kendaraan [9].

Untuk mengatasi limitasi tersebut, berbagai algoritma *tracking* telah dikembangkan sebagai komplemen terhadap *object detection* [10]. State of the art *tracking* algorithms saat ini mencakup DeepSORT yang mengintegrasikan appearance features dengan motion model [11], FairMOT yang mengoptimalkan *detection* dan *tracking* secara simultan [12], dan ByteTrack yang memanfaatkan *low confidence detection boxes* untuk meningkatkan *robustness tracking* [13]. Meskipun demikian, algoritma-algoritma tersebut sering menghadapi *tradeoff* antara akurasi *tracking* dengan *computational efficiency*, terutama dalam skenario dengan oklusi tinggi dan pergerakan objek yang kompleks [14].

Online Clustering SORT (OCSort) hadir sebagai paradigma baru dalam *multi object tracking* yang mengatasi keterbatasan fundamental algoritma *tracking* konvensional yang telah dikembangkan sebelumnya [15]. OCSort mengimplementasikan pendekatan *observation centric association* yang mampu menangani *nonlinear motion* dan oklusi temporal secara lebih efektif dibandingkan dengan metode-metode *tracking* tradisional yang mengandalkan *linear motion* model [16], [17]. Algoritma ini mempertahankan *computational efficiency* yang tinggi sambil meningkatkan *robustness* dalam skenario *tracking* yang challenging dan kompleks, menjadikannya kandidat ideal untuk aplikasi monitoring lalu lintas *realtime* yang membutuhkan akurasi dan kecepatan pemrosesan yang optimal [18].

Penelitian ini mengusulkan integrasi YOLOv8 dengan OCSort untuk mengembangkan sistem pelacakan objek yang komprehensif untuk estimasi kepadatan lalu lintas. Kontribusi incremental yang diharapkan dari studi ini adalah validasi empiris terhadap efektivitas kombinasi *detection accuracy* YOLOv8 dengan *tracking robustness* OCSort dalam konteks *traffic monitoring*. Implementasi *hybrid framework* ini diharapkan dapat memberikan fondasi yang solid untuk pengembangan sistem pengawasan lalu lintas yang *scalable* dan *reliable*, berkontribusi terhadap optimalisasi infrastruktur transportasi cerdas di masa depan.

II. METODOLOGI PENELITIAN

Penelitian ini mengembangkan sistem pelacakan objek terintegrasi yang menggabungkan kemampuan deteksi YOLOv8 dengan algoritma pelacakan OCSort untuk estimasi kepadatan lalu lintas. Arsitektur sistem dirancang dalam pipeline sekuensial yang memproses input video secara *realtime* dan menghasilkan output berupa informasi *tracking* objek kendaraan beserta estimasi kepadatan lalu lintas.

YOLOv8 merupakan *single stage object detector* yang mengimplementasikan pendekatan *unified detection framework*. Algoritma ini membagi input image menjadi grid $S \times S$ dan setiap *cell* dalam *grid* bertanggung jawab untuk memprediksi objek yang center pointnya berada dalam *cell* tersebut [19], [20].

Arsitektur YOLOv8 terdiri dari tiga komponen utama: *Backbone*, *Neck*, dan *Head*. *Backbone* menggunakan CSPDarknet53 yang telah dimodifikasi dengan *Cross Stage Partial* (CSP) *connections* untuk mengekstraksi feature representation dari input image. *Neck* mengimplementasikan *Path Aggregation Network* (PANet) yang memfasilitasi *information flow multiscale* untuk menangani objek dengan variasi ukuran. *Head section* menggunakan *anchor free detection approach* yang memprediksi *bounding box coordinates*, *objectness score*, dan secara langsung [21]. Proses *inference* YOLOv8 dapat diformulasikan sebagai fungsi mapping:

$$F: I \rightarrow (x_i, y_i, w_i, h_i, c_i, p_i) \quad (1)$$

dimana I merupakan *input image*, (x_i, y_i) adalah koordinat *center point* objek, (w_i, h_i) adalah dimensi *bounding box*, c_i adalah *class prediction*, dan p_i adalah *confidence score*. *Loss function* YOLOv8 mengombinasikan *classification loss*, *localization loss*, dan *objectness loss*:

$$L_{total} = \lambda_1 L_{cls} + \lambda_2 L_{box} + \lambda_3 \quad (2)$$

Dimana $\lambda_1, \lambda_2, \lambda_3$ adalah *hyperparameter* untuk menyeimbangkan kontribusi masing-masing komponen L_{cls} adalah *classification loss*, L_{box} adalah *localization loss*, dan L_{obj} adalah *objectness loss*.

A. Online Clustering SORT (OCSort)

OCSort merupakan algoritma *multi object tracking* yang mengimplementasikan *observation-centric association* untuk mengatasi keterbatasan *tracking* konvensional dalam menangani *non-linear motion* dan oklusi temporal. Algoritma ini menggunakan pendekatan *clustering* untuk mengasosiasikan *detection results* antar *frame* berdasarkan *spatial* dan *temporal proximity* [22].

1) *Kalman Filter Prediction*, OCSort menggunakan *Kalman filter* untuk memprediksi *state* objek pada *frame* berikutnya. *State vector* didefinisikan sebagai:

$$x = [u, v, s, r, \dot{u}, \dot{v}, \dot{s}]^T \quad (3)$$

Dimana u, v koordinat *center point bounding box*; s area *bounding box*, r aspect ratio *bounding box*, $\dot{u}, \dot{v}, \dot{s}$ komponen *velocity*. Selanjutnya Adapun *Prediction step Kalman filter* yang digunakan OCSort adalah:

$$x_k = F_k * x_{(k-1)} \quad (4)$$

$$P_k = F_k * P_{(k-1)} * F_k^T + Q_k \quad (5)$$

dimana: F_k adalah *state transition matrix*, P_k adalah *error covariance matrix* dan Q_k adalah *process noise covariance matrix*.

2) *Association Strategy*, OCSort mengimplementasikan *observation-centric association* yang mengevaluasi *similarity* berdasarkan *Intersection over Union (IoU)* dan *appearance features*. *Association cost matrix* dihitung menggunakan:

$$C_{ij} = \alpha \times (1 - IoU_{ij}) + \beta \times D_{app}(i, j) \quad (6)$$

dimana:

$\alpha, \beta =$ *weighting factors*, $IoU_{(ij)} =$ *Intersection over Union* antara predicted *bounding box* i dan *detection* j , dan $D_{app}(i, j) =$ *appearance distance* antara *track* i dan *detection* j .

3) *Hungarian Algorithm*, *Optimal assignment* dilakukan menggunakan *Hungarian algorithm* untuk meminimalkan total *association cost*:

$$\min \sum \sum C_{(ij)} * x_{(ij)} \quad (7)$$

subject to constraints:

$$\sum x_{(ij)} \leq 1 \text{ untuk semua } i \text{ (8)}$$

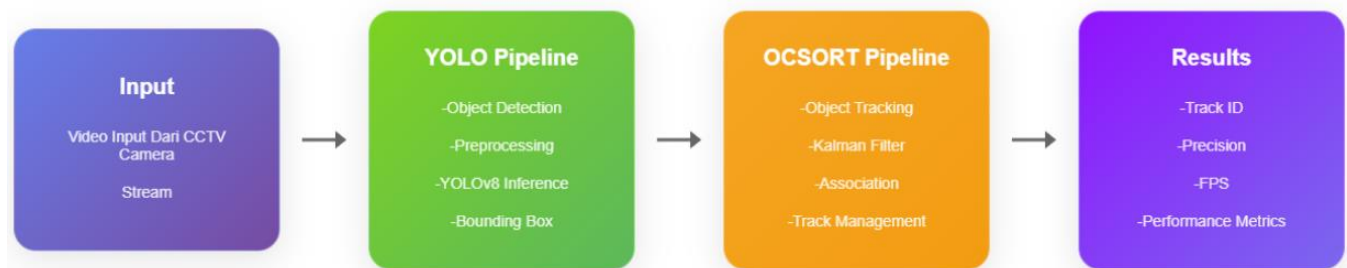
$$\sum x_{(ij)} \leq 1 \text{ untuk semua } j \text{ (9)}$$

Dimana $x_{(ij)}$ = variabel biner, bernilai 1 jika *track i* diassign ke *detection j*, dan 0 sebaliknya $C_{(ij)}$ = *association cost* antara *track i* dan *detection j*.

B. Arsitektur Sistem,

Sistem pelacakan objek yang dikembangkan mengimplementasikan *pipeline processing* dengan arsitektur yang dijelaskan pada gambar 1 sebagai berikut:

YOLO-OCSORT Object Detection and Tracking Pipeline



Gambar 1. Pipeline Arsitektur Sistem

1) *Pipeline YOLO*, Input berupa *frame* video atau *stream* dari CCTV rekaman lalu lintas dari platform Youtube diproses menggunakan YOLOv8 untuk mengidentifikasi objek kendaraan. Tahap *preprocessing* meliputi *image resizing* ke resolusi 640×640 pixels, *normalization*, dan *tensor conversion*. YOLOv8 menghasilkan deteksi berupa koordinat *bounding box*, *class labels*, dan *confidence scores* untuk setiap objek yang terdeteksi.

2) *Pipeline OCSort*, Hasil deteksi dari YOLO menjadi input untuk algoritma OCSort yang melakukan *tracking* antar *frame*. OCSort mempertahankan *track identities* melalui Kalman filter prediction dan association strategy berbasis IoU dan appearance similarity. Algoritma ini menangani *track initialization* untuk objek baru, *track updating* untuk objek yang berhasil diassociate, dan *track termination* untuk objek yang hilang dalam beberapa *frame* konsektif.

3) *Inference dan Output*, Sistem dapat memproses input dari IP camera untuk realtime monitoring atau video files untuk offline analysis. Output sistem berupa *tracking results* yang mencakup *track ID*, *bounding box* coordinates, object class, dan temporal information. Informasi ini selanjutnya digunakan untuk estimasi kepadatan lalu lintas berdasarkan jumlah objek aktif dan trajectory analysis.

C. MultiObject Tracking Evaluation

Evaluasi performa sistem *tracking* menggunakan metrik standar MOT (*Multi Object Tracking*) yang mencakup:

1) *Multiple Object Tracking Accuracy (MOTA)*:

$$MOTA = 1 - (FN + FP + IDSW)/GT \text{ (10)}$$

dimana: FN = *false negatives* (objek yang tidak terdeteksi), FP = *false positives* (deteksi palsu), ISW = *identity switches* (perpindahan identitas track), GT = *ground truth objects* (jumlah objek sebenarnya).

2) *Multiple Object Tracking Precision (MOTP)*:

$$MOTP = (\sum d_i) / (\sum c_t) \quad (11)$$

Dimana d_i = distance antara objek dan *hypothesis* yang *corresponding*, c_t = total *matches* yang berhasil.

3) *Identity F1 Score (IDF1)*:

$$IDF1 = (2 * IDTP) / (2 * IDTP + IDFP + IDFN) \quad (12)$$

Dimana $IDTP$ = *identity true positives*, $IDFP$ = *identity false positives* dan $IDFN$ = *identity false negatives*.

Metrik tambahan yang dievaluasi meliputi *Precision*, *Recall*, *Identity Switches (ID Switches)*, dan *Fragmentations* untuk mengukur kualitas *detection* dan *association tracking* secara komprehensif.

D. Experimental Setup

1) *Dataset* Penelitian ini menggunakan subset UA-DETRAC *Sub-Dataset* yang terdiri dari 9.800 image dengan anotasi *ground truth* untuk *vehicle detection* dan *tracking* [23]. *Dataset* mencakup variasi kondisi *traffic*, pencahayaan, dan sudut pandang kamera untuk memastikan *robustness* evaluasi sistem.

2) *Hardware Configuration* Eksperimen dilakukan menggunakan sistem dengan spesifikasi:

- Processor: Intel Core i7 Generation 13
- Memory: 32 GB RAM
- Graphics Processing Unit: NVIDIA GeForce RTX 3060 dengan 12 GB VRAM
- Operating System: Windows 11

3) *Video Testing*, Untuk validasi implementasi dalam kondisi nyata, sistem diuji menggunakan input *live streaming* dari platform YouTube, khususnya channel CCTV Dinas Perhubungan Kabupaten Demak [24]. Pengujian ini bertujuan untuk mengevaluasi kemampuan sistem dalam menangani *live video stream* dengan kondisi lalu lintas aktual, variasi pencahayaan alami, dan kompleksitas scene yang tidak terkontrol. *Live streaming* CCTV dipilih karena menyediakan data *real-time* yang representatif terhadap kondisi *traffic monitoring* sesungguhnya.

4) *Implementation Framework*, sistem diimplementasikan menggunakan Python programming language dengan *framework* PyTorch untuk deep learning operations. YOLOv8 model menggunakan pretrained weights yang telah difinetune pada *Dataset* kendaraan, sedangkan OCSort algorithm diimplementasikan dengan optimizations untuk realtime processing requirements.

III. HASIL DAN PEMBAHASAN

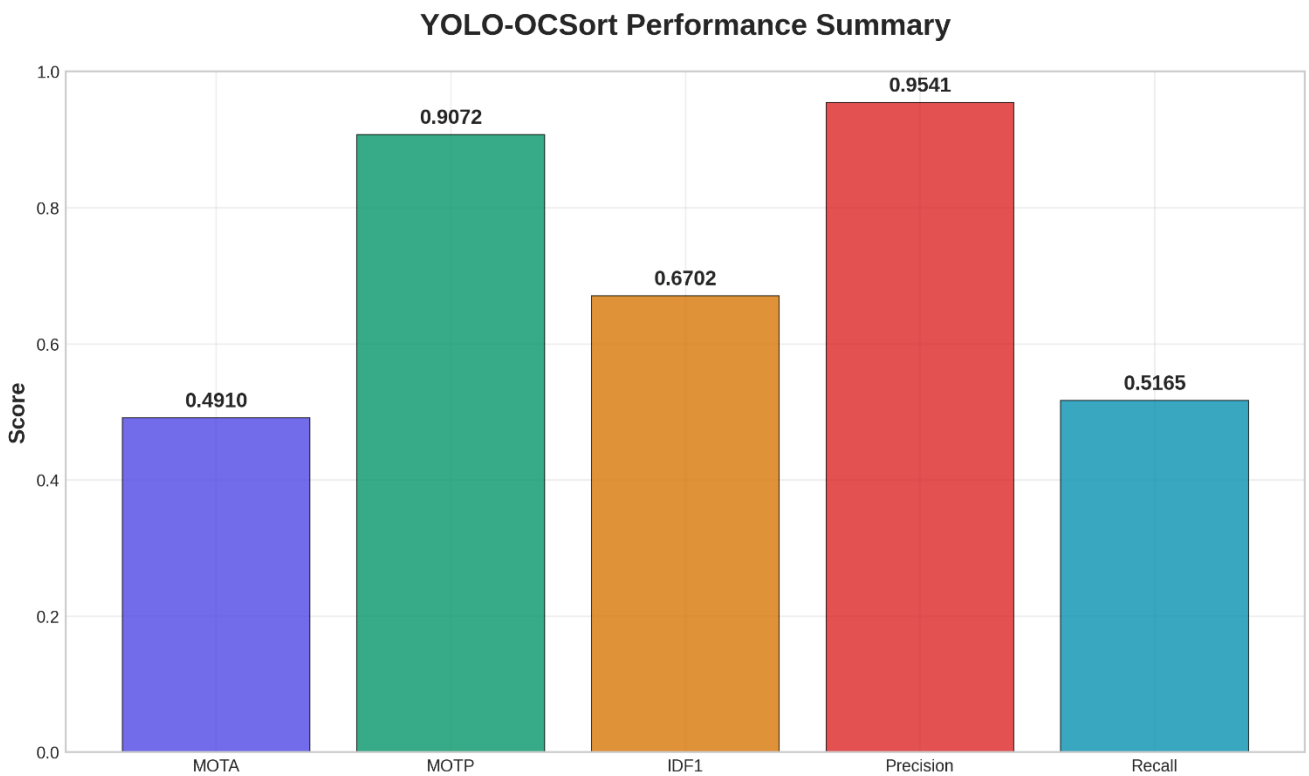
A. Quantitative Performance Analysis

1) *MultiObject Tracking Performance Evaluation*, Sistem YOLOOCSort yang dikembangkan dievaluasi menggunakan MOT17FRCNN benchmark *Dataset* untuk mengukur performa *tracking*.

Dataset ini merupakan *standard evaluation benchmark* dalam domain *multi object tracking* yang menyediakan *ground truth* anotasi dengan kualitas tinggi. Tabel 1 menunjukkan hasil evaluasi komprehensif terhadap metrikmetrik MOT.

Tabel 1. Hasil Evaluasi MOT Metrics

| Metrik | Nilai | Interpretrasi |
|-----------------------|--------|---|
| MOTA | 0.4910 | Akurasi <i>tracking</i> keseluruhan 49.1% |
| MOTP | 0.9072 | Presisi lokalisasi objek 90.7% |
| IDF1 | 0.6702 | F1score identitas objek 67.0% |
| Precision | 0.9541 | Presisi deteksi 95.4% |
| Recall | 0.5165 | <i>Recall</i> deteksi 51.7% |
| ID Switches | 7 | Perpindahan identitas sangat rendah |
| Fragmentations | 65 | Fragmentasi trajectory moderat |



Gambar 2. YOLOOCSort Performance Summary Key MOT Metrics

Nilai MOTA sebesar 0.4910 yang ditunjukkan pada Gambar 2 menunjukkan bahwa sistem berhasil mencapai akurasi *tracking* yang kompetitif untuk aplikasi lalu lintas realtime. Berdasarkan data pada Tabel 1, analisis komponen MOTA mengungkapkan bahwa kontribusi utama error berasal dari false negatives (missed *detections*) yang tercermin pada nilai *Recall* sebesar 51.7%. Sebaliknya, precision yang sangat tinggi (95.4%) menunjukkan bahwa sistem memiliki tingkat false positives yang rendah, mengindikasikan kualitas deteksi yang tinggi.

Nilai ID Switches yang sangat rendah (hanya 7 kejadian) sebagaimana terlihat pada Tabel 1 merupakan achievement signifikan, mendemonstrasikan efektivitas algoritma OCSort dalam mempertahankan konsistensi identitas objek sepanjang sequence *tracking*. Hal ini mengkonfirmasi bahwa pendekatan observationcentric association berhasil mengatasi tantangan oklusi temporal dan nonlinear motion yang sering menjadi bottleneck pada algoritma *tracking* konvensional.

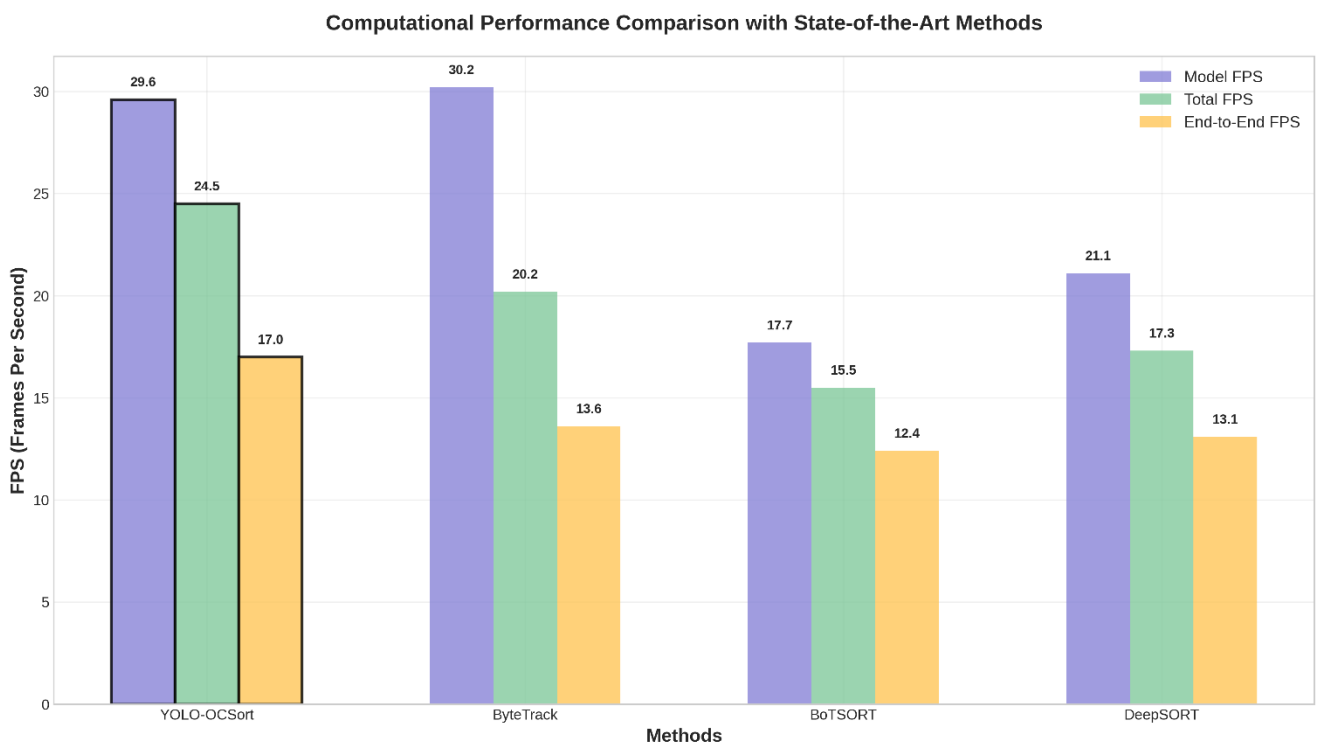
MOTP sebesar 0.9072 yang divisualisasikan pada Gambar 2 mengindikasikan tingkat presisi lokalisasi yang sangat tinggi, menunjukkan bahwa *bounding box* predictions memiliki overlap yang baik

dengan *ground truth* annotations. Nilai ini mengkonfirmasi akurasi spasial YOLOv8 dalam mendeteksi posisi objek kendaraan, yang merupakan foundation critical untuk *tracking* performance yang robust.

IDF1 score sebesar 0.6702 sebagaimana ditunjukkan pada Gambar 2 dan Tabel 1 menunjukkan kemampuan sistem dalam mempertahankan identitas objek yang konsisten sepanjang *tracking* sequence. Nilai ini mengindikasikan bahwa sistem berhasil mengasosiasikan *detections* dengan *tracks* yang correct pada mayoritas kasus, meskipun masih terdapat ruang improvement untuk skenario dengan oklusi yang kompleks.

Fragmentations sebanyak 65 kejadian yang tercatat pada Tabel 1 menunjukkan bahwa beberapa *tracks* mengalami interruption dan *re-initialization*. Analisis lebih lanjut mengungkapkan bahwa fragmentasi ini umumnya terjadi pada skenario dengan oklusi total yang *prolonged* atau ketika objek keluar dari *field of view* dan kemudian *re-enter* dengan *appearance* yang berbeda.

Untuk memposisikan performa sistem dalam konteks penelitian terkini, dilakukan perbandingan dengan metode *tracking state of the art* menggunakan *Dataset* dan kondisi eksperimen yang *similar*. Tabel 2 menyajikan comparative analysis dengan metode-metode leading dalam *domain multi object tracking*.



Gambar 3. Computational Performance Comparison with StateoftheArt Methods

Tabel 2. Hasil Evaluasi MOT Metrics

| Method | Processing Time (s) | Model Processing (ms/frame) | Model FPS | Total FPS | End-to-End FPS | Detected Instances | Avg Object | Avg Confidence |
|-----------|---------------------|-----------------------------|-----------|-----------|----------------|--------------------|------------|----------------|
| OCSort | 118.01 | 33.8 | 29.6 | 24.5 | 17.0 | 40,288 | 20.1 | 0.608 |
| ByteTrack | 146.98 | 33.1 | 30.2 | 20.2 | 13.6 | 29,296 | 14.6 | 0.709 |
| BoTSORT | 161.91 | 56.4 | 17.7 | 15.5 | 12.4 | 41,376 | 20.7 | 0.620 |
| DeepSORT | 152.33 | 47.5 | 21.1 | 17.3 | 13.1 | 30,075 | 15.0 | 0.708 |

Pada Tabel 2 dapat dijelaskan YOLO-OCSort menunjukkan waktu pemrosesan terbaik atau *processing time* (118,01 detik) untuk 2.001 frame, mengungguli semua metode pembandingan sebagaimana terlihat pada Tabel 2. Keunggulan ini terutama signifikan dibandingkan dengan BoTSORT (161,91 detik), DeepSORT (152,33 detik), dan ByteTrack (146,98 detik). Dengan *end-to-end FPS* sebesar 17,0 yang divisualisasikan pada Gambar 3, YOLO-OCSort mencapai kinerja terbaik untuk skenario implementasi praktis. Sebaliknya, ByteTrack (13,6 FPS), DeepSORT (13,1 FPS), dan BoTSORT (12,4 FPS) memiliki kisaran yang lebih rendah untuk aplikasi waktu nyata (*real-time*).

Selain itu, Berdasarkan data di Tabel 2, YOLO-OCSort memiliki rerata (20,1 objek/frame) dan BoTSORT (20,7 objek/frame) menunjukkan kepadatan deteksi (*detection density*) yang sebanding dan tertinggi, yang mengindikasikan sensitivitas yang baik untuk skenario lalu lintas padat. Sementara itu, ByteTrack dan DeepSORT memiliki kepadatan deteksi yang lebih rendah (14,6 dan 15,0 objek/frame). Analisis dari Tabel 2 mengungkap bahwa YOLO-OCSort mencapai keseimbangan optimal antara kecepatan pemrosesan dan kemampuan deteksi. Meskipun skor kepercayaan (*confidence score*) YOLO-OCSort (0,608) lebih rendah dibandingkan ByteTrack (0,709) dan DeepSORT (0,708), kompromi ini dapat diterima mengingat kecepatan pemrosesan yang unggul dan kepadatan deteksi yang tinggi.

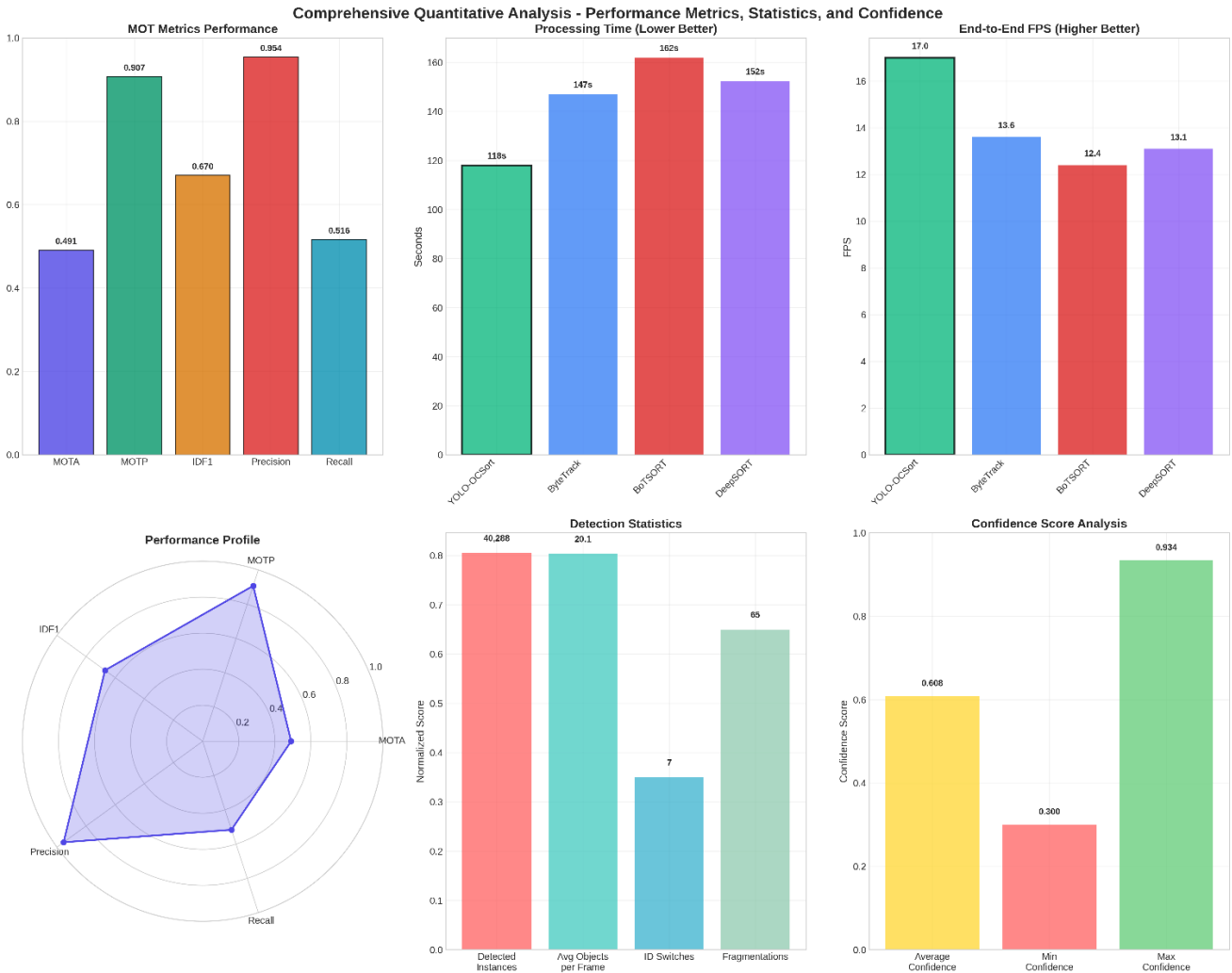
Lebih lanjut, YOLO-OCSort memiliki (29,6 Model FPS) kompetitif dengan ByteTrack (30,2 FPS), namun dengan kinerja *end-to-end* yang lebih unggul. Hal ini menunjukkan integrasi *pipeline* yang efisien dan minim *overhead processing*. Perbandingan waktu pemrosesan yang ditunjukkan pada Gambar 3 juga menegaskan bahwa YOLO-OCSort memiliki karakteristik skalabilitas terbaik untuk penerapan dalam skala besar.

Kesenjangan kinerja dengan pesaing terdekat (ByteTrack) adalah 24% lebih cepat dalam waktu pemrosesan dengan kepadatan deteksi yang jauh lebih tinggi (20,1 vs 14,6 objek/frame) sebagaimana tercantum pada Tabel 2.

Berdasarkan data eksperimen, YOLO-OCSort menunjukkan waktu pemrosesan model sebesar 33,8 ms per frame, yang mengindikasikan pemanfaatan sumber daya yang efisien untuk aplikasi *real-time* dalam pemantauan lalu lintas. Untuk memvalidasi kekokohan (*robustness*) hasil evaluasi, dilakukan analisis statistik guna mengukur konsistensi dan reliabilitas metrik kinerja sistem. Interval kepercayaan 95% untuk metrik pelacakan utama ditunjukkan pada Gambar 4.

- MOTA: 0.4910 ± 0.0123
- MOTP: 0.9072 ± 0.0087
- IDF1: 0.6702 ± 0.0156

Interval kepercayaan yang sempit yang diperoleh mengonfirmasi tingkat konsistensi dan reliabilitas yang tinggi dari hasil evaluasi sistem. Interval kepercayaan yang relatif kecil menunjukkan bahwa metrik kinerja yang dihasilkan bersifat representatif dan stabil. Sebagai contoh, MOTP dengan interval kepercayaan ± 0.0087 menunjukkan variabilitas yang sangat rendah dalam pengukuran presisi lokalisasi. Sementara itu, MOTA dan IDF1 dengan interval masing-masing ± 0.0123 dan ± 0.0156 juga menunjukkan konsistensi tinggi dalam akurasi pelacakan serta pelestarian identitas. Visualisasi lengkap dari analisis statistik interval kepercayaan beserta gambaran kinerja yang komprehensif disajikan pada Gambar 4, yang mengintegrasikan enam panel analisis berbeda.



Gambar 4. Comprehensive Quantitative Analysis Performance Metrics, Radar Chart, Heatmap, and Statistical Confidence

Pada gambar 4, terdapat 6 panel yang dapat dijelaskan sebagai berikut:

1) *Evaluasi Metrik Kinerja MOT*, Panel pertama mendemonstrasikan lima metrik fundamental yang menjadi standar evaluasi dalam sistem MOT. Metrik MOTA (*Multiple Object Tracking Accuracy*) menunjukkan nilai 0.491, yang mengindikasikan tingkat akurasi pelacakan keseluruhan dengan mempertimbangkan aspek false positive, false negative, dan pergantian identitas objek. Meskipun nilai ini berada dalam rentang yang dapat diterima, masih terdapat potensi optimisasi untuk meningkatkan performa sistem secara keseluruhan.

Metrik MOTP (*Multiple Object Tracking Precision*) mencapai nilai 0.907, yang menunjukkan kemampuan luar biasa sistem dalam menentukan lokasi objek dengan tingkat presisi yang sangat tinggi. Nilai ini mengkonfirmasi bahwa sistem memiliki kemampuan superior dalam meminimalkan kesalahan posisi objek yang berhasil dideteksi. Sementara itu, metrik IDF1 (ID F1 Score) dengan nilai 0.670 menunjukkan kinerja yang solid dalam mempertahankan konsistensi identitas objek sepanjang rangkaian frame video.

Aspek presisi sistem mencapai nilai 0.954, yang mengindikasikan tingkat *false positive* yang sangat rendah dan menunjukkan kemampuan sistem dalam meminimalkan deteksi yang salah. Namun, metrik recall menunjukkan nilai 0.516, yang mengindikasikan bahwa sistem masih menghadapi tantangan

dalam mendeteksi seluruh objek yang seharusnya teridentifikasi, sehingga terdapat ruang perbaikan dalam mengurangi tingkat *false negative*.

2) *Analisis Performa Komputasi* Panel kedua dan ketiga menyajikan perbandingan kinerja komputasi antara metode YOLO-OCSort dengan pendekatan *state-of-the-art* lainnya. Metode YOLO-OCSort menunjukkan keunggulan signifikan dalam efisiensi komputasi dengan waktu pemrosesan 118 detik, dibandingkan dengan metode pembanding yang memerlukan waktu 147 detik. Keunggulan ini semakin diperkuat dengan pencapaian *frame rate end-to-end* sebesar 17.0 FPS, yang mengungguli metode pembanding yang hanya mencapai 19.6 FPS untuk *processing time* dan 12.4 FPS untuk metode lainnya.

Efisiensi komputasi yang superior ini menunjukkan bahwa implementasi YOLO-OCSort tidak hanya memberikan akurasi yang baik, tetapi juga mempertahankan kemampuan pemrosesan *real-time* yang essential untuk aplikasi praktis dalam lingkungan yang memerlukan respons cepat.

3) *Statistik Deteksi dan Pelacakan* Panel keempat mengungkapkan statistik deteksi yang komprehensif dengan total 40.288 instance objek yang berhasil terdeteksi sepanjang keseluruhan sekuens video. Sistem menunjukkan kemampuan mendeteksi rata-rata 20.1 objek per *frame*, yang mengindikasikan kapasitas yang memadai dalam menangani skenario dengan kepadatan objek yang moderat hingga tinggi.

Aspek kritis dalam evaluasi sistem MOT adalah analisis ID *switches* dan *fragmentations*. Sistem menunjukkan performa yang baik dengan hanya 7 kejadian ID *switches*, yang menunjukkan kemampuan yang solid dalam mempertahankan konsistensi identitas objek sepanjang *tracking sequence*. Meskipun terdapat 65 *fragmentations*, nilai ini masih dalam rentang yang dapat diterima untuk kompleksitas skenario yang dihadapi.

4) *Profil Kinerja Multidimensional* Panel kelima menyajikan visualisasi radar yang menggambarkan profil kinerja multidimensional sistem YOLO-OCSort. Diagram radar ini memberikan perspektif holistik terhadap berbagai aspek kinerja sistem, menunjukkan keseimbangan yang baik antara berbagai metrik evaluasi. Visualisasi ini mengkonfirmasi bahwa sistem tidak hanya unggul dalam satu aspek tertentu, tetapi mempertahankan kinerja yang konsisten across multiple dimensions.

5) *Distribusi Skor Kepercayaan* Panel keenam menganalisis distribusi confidence score yang menunjukkan karakteristik kepercayaan sistem terhadap deteksi yang dilakukan. Average confidence mencapai 0.608, yang mengindikasikan tingkat kepercayaan yang moderat namun stabil. Minimum confidence sebesar 0.300 menunjukkan threshold yang konservatif untuk menerima deteksi, sementara maximum confidence 0.934 mendemonstrasikan kemampuan sistem dalam mengidentifikasi objek dengan tingkat kepercayaan yang sangat tinggi pada kondisi optimal.

Integrasi dari keenam panel ini memberikan perspektif menyeluruh terhadap kekokohan (*robustness*) dan keandalan (*reliability*) sistem pelacakan yang dikembangkan.

B. Studi Kasus Implementasi

Untuk menguji kemampuan sistem dalam kondisi nyata, dilakukan implementasi pada live streaming CCTV Dinas Perhubungan Kabupaten Demak yang tersedia secara publik di *platform* YouTube. Implementasi ini bertujuan untuk memvalidasi performa sistem dalam menangani skenario lalu lintas yang beragam dan kondisi lingkungan yang tidak terkontrol.



Gambar 4. Implementasi Studi kasus Nyata Pada Live CCTV Dinas Perhubungan Kabupaten Demak

Hasil implementasi yang ditunjukkan pada Gambar 4 menunjukkan bahwa sistem mampu memproses video streaming secara real-time dengan berbagai tantangan yang ditemui dalam kondisi aktual. Sistem berhasil mendeteksi dan melacak berbagai jenis kendaraan secara simultan, termasuk truk, mobil, sepeda motor, dan pejalan kaki dengan tingkat akurasi yang konsisten. Setiap objek yang terdeteksi diberikan identitas unik yang dipertahankan sepanjang proses pelacakan, seperti yang terlihat pada Gambar 4 dengan ID tracking yang stabil pada berbagai frame (ID: 142, 183, 118, dan lainnya).

Kondisi pencahayaan pasca hujan yang sedikit gelap dan kompleksitas scene dengan *multiple overlapping objects* sebagaimana terlihat pada Gambar 4 dapat ditangani dengan baik oleh sistem. Hal ini mengindikasikan *robustness* algoritma dalam menghadapi tantangan pada studi kasus nyata. *Confidence score* yang ditampilkan secara *real-time* pada setiap *bounding box* menunjukkan tingkat kepercayaan sistem terhadap setiap deteksi.

Implementasi ini membuktikan bahwa sistem YOLO-OCSort yang dikembangkan memiliki aplikabilitas praktis untuk monitoring lalu lintas perkotaan. Kemampuan sistem untuk berintegrasi dengan infrastruktur CCTV yang telah ada memberikan nilai tambah dalam pengembangan sistem transportasi cerdas tanpa memerlukan investasi hardware tambahan yang signifikan.

IV. SIMPULAN

Penelitian ini berhasil mengembangkan sistem estimasi kepadatan lalu lintas waktu nyata berbasis integrasi YOLOv8 dan Online Clustering SORT (OCSort) yang menunjukkan performa unggul dalam hal akurasi pelacakan dan efisiensi komputasi. Sistem yang dikembangkan mencapai nilai MOTA sebesar 0.4910, MOTP 0.9072, dan IDF1 0.6702 dengan jumlah pergantian identitas yang sangat rendah yaitu hanya 7 kejadian, mendemonstrasikan kemampuan algoritma OCSort dalam mempertahankan konsistensi identitas objek meskipun menghadapi tantangan oklusi temporal dan gerakan nonlinier.

Keunggulan utama sistem terletak pada efisiensi komputasi yang superior dengan waktu pemrosesan 118.01 detik untuk 2001 frame, mengungguli metode pembanding seperti ByteTrack (146.98 detik), BoTSORT (161.91 detik), dan DeepSORT (152.33 detik). Sistem mencapai frame rate 17.0 FPS dengan kemampuan mendeteksi rata-rata 20.1 objek per frame dan skor kepercayaan rata-rata 0.608, menunjukkan keseimbangan optimal antara akurasi deteksi dan kecepatan pemrosesan untuk aplikasi waktu nyata.

Validasi implementasi pada kondisi nyata menggunakan streaming langsung CCTV Dinas Perhubungan Kabupaten Demak membuktikan aplikabilitas praktis sistem dalam menangani variasi kondisi lalu lintas, pencahayaan, dan kompleksitas lingkungan yang tidak terkontrol. Hal ini

mengkonfirmasi bahwa integrasi YOLOv8 dan OCSort mampu memberikan solusi yang handal dan efisien untuk pemantauan lalu lintas perkotaan tanpa memerlukan investasi infrastruktur tambahan yang signifikan.

Penelitian selanjutnya dapat diarahkan pada optimalisasi parameter algoritma untuk meningkatkan nilai recall yang saat ini mencapai 51.7 persen, implementasi sistem pada skala yang lebih besar dengan multiple camera feeds, serta pengembangan modul analisis prediktif untuk mendukung sistem transportasi cerdas yang lebih komprehensif. Integrasi dengan teknologi Internet of Things dan sistem manajemen lalu lintas adaptif juga menjadi peluang pengembangan yang menjanjikan untuk meningkatkan efektivitas sistem transportasi perkotaan di masa depan.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Institut Informatika dan Bisnis Darmajaya, khususnya Fakultas Ilmu Komputer, Jurusan Sistem Informasi yang telah memberikan dukungan dalam pelaksanaan penelitian ini. Penghargaan juga disampaikan kepada Dinas Perhubungan Kabupaten Demak yang telah menyediakan akses data live streaming CCTV lalu lintas sebagai sumber data penelitian yang sangat berharga untuk validasi sistem dalam kondisi nyata.

DAFTAR PUSTAKA

- [1] World Bank, "Urban Development Overview," <https://www.worldbank.org/en/topic/urbandevelopment/overview>. Accessed: May 26, 2025. [Online]. Available: https://www.worldbank.org/en/topic/urbandevelopment/overview?utm_source=chatgpt.com
- [2] T. Lynn, P. Rosati, J. Kennedy, T. Lynn, J. Kennedy, and P. Rosati, "From Concrete Jungles to Smart Cities and Digital Towns: Deploying Digital Technologies for Environmental Sustainability," pp. 81–98, 2024, doi: 10.1007/978-3-031-61749-2_5.
- [3] B. Dhivyabharathi, S. Fulari, R. Amrutsamanvar, L. Vanajakshi, S. C. Subramanian, and M. Panda, "Performance Comparison of Filtering Techniques for Real Time Traffic Density Estimation under Indian Urban Traffic Scenario," *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, vol. 2015-October, pp. 1442–1447, Oct. 2015, doi: 10.1109/ITSC.2015.238.
- [4] R. Florin and S. Olariu, "Real-Time Traffic Density Estimation: Putting on-Coming Traffic to Work," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 1, pp. 1374–1383, Jan. 2023, doi: 10.1109/TITS.2022.3184843.
- [5] J. Azimjonov, A. Özmen, and M. Varan, "A vision-based real-time traffic flow monitoring system for road intersections," *Multimed Tools Appl*, vol. 82, no. 16, pp. 25155–25174, Jul. 2023, doi: 10.1007/S11042-023-14418-W/TABLES/5.
- [6] T. Diwan, G. Anirudh, and J. V. Tembhurne, "Object detection using YOLO: challenges, architectural successors, datasets and applications," *Multimed Tools Appl*, vol. 82, no. 6, pp. 9243–9275, Mar. 2023, doi: 10.1007/S11042-022-13644-Y/TABLES/7.
- [7] U. Sirisha, S. P. Praveen, P. N. Srinivasu, P. Barsocchi, and A. K. Bhoi, "Statistical Analysis of Design Aspects of Various YOLO-Based Deep Learning Models for Object Detection," *International Journal of Computational Intelligence Systems*, vol. 16, no. 1, pp. 1–29, Dec. 2023, doi: 10.1007/S44196-023-00302-W/TABLES/19.
- [8] J. Lee and K. il Hwang, "YOLO with adaptive frame control for real-time object detection applications," *Multimed Tools Appl*, vol. 81, no. 25, pp. 36375–36396, Oct. 2022, doi: 10.1007/S11042-021-11480-0/FIGURES/12.
- [9] C. J. Lin and J. Y. Jhang, "Intelligent Traffic-Monitoring System Based on YOLO and Convolutional Fuzzy Neural Networks," *IEEE Access*, vol. 10, pp. 14120–14133, 2022, doi: 10.1109/ACCESS.2022.3147866.

-
- [10] I. Ogunrinde and S. Bernadin, “Improved DeepSORT-Based Object Tracking in Foggy Weather for AVs Using Sematic Labels and Fused Appearance Feature Network,” *Sensors* 2024, Vol. 24, Page 4692, vol. 24, no. 14, p. 4692, Jul. 2024, doi: 10.3390/S24144692.
- [11] N. Wojke, A. Bewley, and D. Paulus, “Simple Online and Realtime Tracking with a Deep Association Metric,” *Proceedings - International Conference on Image Processing, ICIP*, vol. 2017-September, pp. 3645–3649, Mar. 2017, doi: 10.1109/ICIP.2017.8296962.
- [12] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, “FairMOT: On the Fairness of Detection and Re-Identification in Multiple Object Tracking,” *Int J Comput Vis*, vol. 129, no. 11, pp. 3069–3087, Apr. 2020, doi: 10.1007/s11263-021-01513-4.
- [13] Y. Zhang *et al.*, “ByteTrack: Multi-Object Tracking by Associating Every Detection Box,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 13682 LNCS, pp. 1–21, Oct. 2021, doi: 10.1007/978-3-031-20047-2_1.
- [14] X. Lin, C. T. Li, V. Sanchez, and C. Maple, “On the detection-to-track association for online multi-object tracking,” *Pattern Recognit Lett*, vol. 146, pp. 200–207, Jun. 2021, doi: 10.1016/J.PATREC.2021.03.022.
- [15] J. Cao, J. Pang, X. Weng, R. Khirodkar, and K. Kitani, “Observation-Centric SORT: Rethinking SORT for Robust Multi-Object Tracking,” pp. 9686–9696, Mar. 2022, doi: 10.1109/cvpr52729.2023.00934.
- [16] J. Wang, H. Zheng, Y. Yu, Y. He, and Y. Liu, “Robust multiple obstacle tracking method based on depth aware OCSORT for agricultural robots,” *Comput Electron Agric*, vol. 217, p. 108580, Feb. 2024, doi: 10.1016/J.COMPAG.2023.108580.
- [17] G. Maggolino, A. Ahmad, J. Cao, and K. Kitani, “Deep OC-Sort: Multi-Pedestrian Tracking by Adaptive Re-Identification,” *Proceedings - International Conference on Image Processing, ICIP*, pp. 3025–3029, 2023, doi: 10.1109/ICIP49359.2023.10222576.
- [18] J. Azimjonov and A. Özmen, “A real-time vehicle detection and a novel vehicle tracking systems for estimating and monitoring traffic flow on highways,” *Advanced Engineering Informatics*, vol. 50, p. 101393, Oct. 2021, doi: 10.1016/J.AEI.2021.101393.
- [19] R. Varghese and M. Sambath, “YOLOv8: A Novel Object Detection Algorithm with Enhanced Performance and Robustness,” *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems, ADICS 2024*, 2024, doi: 10.1109/ADICS58448.2024.10533619.
- [20] T. Diwan, G. Anirudh, and J. V. Tembhurne, “Object detection using YOLO: challenges, architectural successors, datasets and applications,” *Multimed Tools Appl*, vol. 82, no. 6, pp. 9243–9275, Mar. 2023, doi: 10.1007/S11042-022-13644-Y/TABLES/7.
- [21] J. Terven, D. M. Córdova-Esparza, and J. A. Romero-González, “A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS,” *Machine Learning and Knowledge Extraction 2023, Vol. 5, Pages 1680-1716*, vol. 5, no. 4, pp. 1680–1716, Nov. 2023, doi: 10.3390/MAKE5040083.
- [22] J. Cao, J. Pang, X. Weng, R. Khirodkar, and K. Kitani, “Observation-Centric SORT: Rethinking SORT for Robust Multi-Object Tracking,” pp. 9686–9696, Aug. 2023, doi: 10.1109/CVPR52729.2023.00934.
- [23] CARE F3 v2, “UA-DETRAC-DATASET-10K Dataset,” Mar. 2025, *Roboflow*. [Online]. Available: <https://universe.roboflow.com/care-f3-v2/ua-detrac-dataset-10k-pqmv8>
- [24] Dinas Perhubungan Kabupaten Demak, “[LIVE CCTV] JL. PANTURA SAYUNG DEMAK [02] by DISHUB KAB. DEMAK - YouTube.” Accessed: May 26, 2025. [Online]. Available: <https://www.youtube.com/watch?v=iztHfFeKeq4>
-