



Yolov11 Optimization Through Hyperparameter Tuning and Data Augmentation to Improve Vehicle Detection Accuracy at Night

Optimasi Yolov11 Melalui *Hyperparameter Tuning* dan Data Augmentasi untuk Meningkatkan Akurasi Deteksi Kendaraan pada Kondisi Malam Hari

Imam Alfath Zulkarnain^{1*}, Kusrini²

^{1,2}Program Studi Informatika, Universitas Amikom Yogyakarta, Indonesia

E-Mail: ¹imamalfath93@gmail.com, ²Kusrini@amikom.ac.id

Received Aug12th 2025; Revised Sep 12th 2025; Accepted Sep 25th 2025; Available Online Oct 30th 2025

Corresponding Author: Imam Alfath Zulkarnain

Copyright © 2025 by Authors, Published by Institut Riset dan Publikasi Indonesia (IRPI)

Abstract

Vehicle detection at night faces significant challenges due to low lighting, headlight glare, and limited object contrast. Low detection accuracy at night is a major obstacle to developing reliable 24/7 intelligent transportation systems (ITS) and traffic surveillance solutions. This study aims to optimize the YOLOv11 algorithm to improve vehicle detection accuracy under such conditions. Optimization was carried out through hyperparameter tuning, including setting the learning rate (0.001), momentum (0.937), and weight decay (0.0005), as well as applying data augmentation techniques such as saturation and brightness adjustment, translation, scaling, horizontal flipping, mosaic, and mixup. The model was tested in two scenarios: (1) nighttime data and (2) a combination of daytime and nighttime data. The results show that the optimized YOLOv11 achieved a precision of 0.97, a recall of 0.92, and an mAP@0.5 of 0.97 in the nighttime scenario, outperforming YOLOv8 and the baseline YOLOv11. In the combined scenario, the model remained superior with a precision of 0.95, a recall of 0.95, and an mAP@0.5 of 0.98. These findings demonstrate that a combination of hyperparameter tuning and adaptive augmentation is effective in improving nighttime vehicle detection performance without reducing daytime accuracy. This approach is promising for application in computer vision-based traffic monitoring systems that require high-performance consistency both day and night, thereby contributing to enhanced traffic safety and efficiency.

Keyword: Computer Vision, Hyperparameter Tuning, Vehicle Detection, YOLOv11

Abstrak

Deteksi kendaraan pada malam hari menghadapi tantangan signifikan akibat pencahayaan rendah, silau lampu depan, dan kontras objek yang terbatas. Akurasi deteksi yang rendah pada malam hari menjadi penghambat utama dalam pengembangan sistem transportasi cerdas (ITS) dan sistem pengawasan lalu lintas yang andal secara 24/7. Penelitian ini bertujuan mengoptimalkan algoritma YOLOv11 untuk meningkatkan akurasi deteksi kendaraan dalam kondisi tersebut. Optimasi dilakukan melalui penyesuaian hiperparameter, termasuk pengaturan laju pembelajaran (0.001), momentum (0.937), dan *weight decay* (0.0005), serta penerapan teknik augmentasi data seperti penyesuaian saturasi dan kecerahan, translasi, skala, *flipping horizontal*, *mosaic*, dan *mixup*. Model diuji dalam dua skenario: (1) data malam hari dan (2) gabungan data siang dan malam. Hasil penelitian menunjukkan bahwa YOLOv11 yang telah dioptimalkan mencapai *precision* 0.97, *recall* 0.92, dan *mAP@0.5* sebesar 0.97 pada skenario malam hari, melampaui kinerja YOLOv8 dan YOLOv11 *baseline*. Pada skenario gabungan, model tetap unggul dengan *precision* 0.95, *recall* 0.95, dan *mAP@0.5* sebesar 0.98. Temuan ini membuktikan bahwa kombinasi penyesuaian hiperparameter dan augmentasi adaptif efektif meningkatkan kinerja deteksi kendaraan pada malam hari tanpa menurunkan akurasi pada kondisi siang. Pendekatan ini menjanjikan untuk diaplikasikan dalam sistem pemantauan lalu lintas berbasis visi komputer yang memerlukan konsistensi performa tinggi baik di siang maupun malam hari, sehingga berkontribusi pada peningkatan keselamatan dan efisiensi lalu lintas.

Kata Kunci: Deteksi Kendaraan, Penyesuaian Hiperparameter, Visi Komputer, YOLOv11

1. PENDAHULUAN

Deteksi objek, khususnya kendaraan, menjadi komponen penting dalam pengembangan sistem pengawasan lalu lintas, manajemen transportasi cerdas (*Intelligent Transportation System*), dan keamanan jalan [1]. Teknologi ini banyak dimanfaatkan dalam berbagai aplikasi, seperti pemantauan lalu lintas secara *real-time*, sistem tilang elektronik, serta analisis kepadatan kendaraan. Meski demikian, akurasi deteksi kendaraan tidak selalu optimal di semua kondisi, terutama pada malam hari. Rendahnya pencahayaan, silau lampu kendaraan, dan gangguan visual menjadi tantangan utama dalam memperoleh hasil deteksi yang tepat [2]. Tingginya angka kecelakaan lalu lintas malam hari serta kebutuhan sistem pengawasan yang beroperasi 24 jam semakin menguatkan urgensi untuk mengatasi masalah ini. Deteksi yang akurat dalam segala kondisi pencahayaan sangat penting untuk meningkatkan keselamatan jalan, mendukung otomasi kendaraan otonom, dan menciptakan kota yang lebih cerdas dan aman.

Salah satu algoritma deteksi objek populer yang banyak digunakan adalah *You Only Look Once* (YOLO) [3]. Alasan pemilihan YOLO, khususnya varian YOLOv11 dalam penelitian ini, didasarkan pada keunggulan fundamentalnya dibandingkan pendekatan *machine learning* tradisional atau arsitektur deep learning lainnya. Pendekatan *machine learning* konvensional (seperti SVM atau Random Forest) sangat bergantung pada fitur *hand-crafted* yang kurang robust dalam menangani variasi pencahayaan dan sudut objek yang ekstrem. Sementara itu, arsitektur *deep learning* lain seperti R-CNN dan turunannya cenderung memiliki komputasi yang lebih berat dan tidak dilakukan untuk pemrosesan *real-time*. YOLO mengatasi keterbatasan ini dengan menggabungkan kecepatan inferensi tinggi dan akurasi yang kompetitif dalam satu arsitektur *unified network*. YOLO dikenal memiliki kecepatan dan akurasi tinggi dalam mendeteksi objek secara *real-time*. Versi terbarunya, YOLOv11, membawa peningkatan arsitektur dari versi sebelumnya, mencakup optimalisasi model, anchor-free detection, serta penerapan teknik augmentasi modern untuk meningkatkan kemampuan generalisasi.

Namun, meskipun unggul, performa YOLOv11 pada kondisi malam hari masih memerlukan optimasi untuk mengatasi keterbatasan pencahayaan dan variasi kontras objek [4]. Berbagai studi sebelumnya menggunakan metode peningkatan akurasi deteksi pada kondisi minim cahaya, seperti image enhancement [5], *data augmentation* [6], dan penyesuaian parameter pelatihan [7]. Strategi yang menjanjikan adalah menggabungkan model deteksi dengan optimasi berbasis augmentasi dan penyesuaian hiperparameter agar lebih adaptif terhadap data malam hari. Penelitian ini berfokus pada optimasi YOLOv11 arsitektur terbaru yang belum banyak dieksplorasi untuk kasus ini dengan strategi augmentasi yang tepat dan pengaturan parameter pelatihan yang cermat. Fokus pada arsitektur mutakhir dan kombinasi teknik optimasi yang dirancang khusus untuk mengatasi tantangan malam hari inilah yang menjadi kebaruan (*novelty*) dari penelitian ini, yang membedakannya dari penelitian sebelumnya yang umumnya menggunakan model lebih lama atau pendekatan berbasis GAN yang kompleks. Tujuannya adalah untuk meningkatkan akurasi dan keandalan deteksi kendaraan di malam hari, serta mendukung pengembangan sistem pengawasan lalu lintas yang lebih efektif.

2. TINJAUAN LITERATUR

Deteksi objek dalam kondisi cahaya rendah, khususnya untuk aplikasi transportasi cerdas, telah menjadi fokus banyak penelitian. Keluarga algoritma YOLO menjadi pilihan populer karena arsitekturnya yang efisien untuk deteksi *real-time* [3]. Perkembangannya dari YOLOv1 [19] hingga YOLOv11 ditandai dengan peningkatan berkelanjutan dalam akurasi dan kecepatan.

Penelitian sebelumnya mengatasi tantangan deteksi malam hari melalui dua pendekatan utama: transformasi citra dan augmentasi data. Pendekatan transformasi citra sering memanfaatkan *Generative Adversarial Networks* (GAN). Studi [8] menggunakan CycleGAN pada *dataset* BDD100K untuk melakukan translasi domain siang ke malam guna melatih YOLOv5, yang dilaporkan meningkatkan akurasi. [9] mengusulkan AugGAN untuk menghasilkan citra malam yang realistis sekaligus mempertahankan detail objek. Namun, pendekatan berbasis GAN seringkali kompleks, membutuhkan komputasi tinggi, dan berisiko menghasilkan artefak yang mengganggu deteksi [10]. Pendekatan lain yang lebih langsung adalah dengan melakukan *low-light image enhancement* (LLIE) sebagai pra-pemrosesan. [12] mengusulkan *framework* untuk meningkatkan citra gelap dari lensa *fisheye* sebelum deteksi, namun metode LLIE dapat memperkenalkan *noise* dan *halo artifact*.

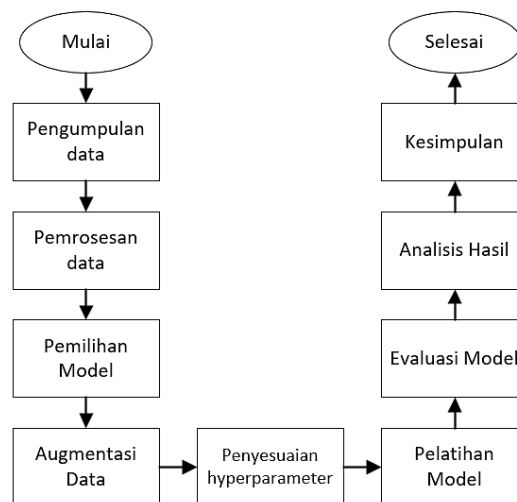
Pada kategori kedua, teknik augmentasi data konvensional dan optimasi model banyak dieksplorasi untuk meningkatkan ketahanan model. Studi [13] dan [14] membuktikan bahwa penyesuaian ruang warna HSV dan transformasi geometri pada YOLOv4 mampu meningkatkan akurasi deteksi malam hari hingga 5%. Penelitian pada YOLOv7 [19] dan YOLOv8 [4], [21] semakin mengonfirmasi pentingnya data augmentasi, khususnya dalam kondisi lingkungan yang buruk. Penelitian [15] (XM-YOLOvIT) dan [16] (LLD-YOLO) juga menyoroti pentingnya desain arsitektur *lightweight* dan spesifik kondisi untuk kinerja *real-time*.

Meskipun berbagai penelitian terdahulu telah menyelidiki deteksi kondisi malam hari menggunakan YOLO, analisis menunjukkan bahwa: (1) mayoritas masih berfokus pada versi YOLO yang lebih lama (v3-v8) [8], [9], [13], [14], [20], [21]; (2) pendekatan yang banyak digunakan seperti GAN [8], [9] dan LLIE [12]

cenderung kompleks dan kurang efisien; (3) optimasi hiperparameter yang komprehensif dan kombinasi augmentasi yang dirancang khusus untuk memaksimalkan kemampuan arsitektur terbaru seperti YOLOv11 masih merupakan celah yang belum banyak dieksplorasi. Berdasarkan analisis ini, penelitian ini berfokus pada optimasi YOLOv11 melalui kombinasi strategi augmentasi data yang ringan (seperti *mosaic*, *mixup*, penyesuaian HSV) dan tuning hiperparameter yang cermat (seperti *learning rate*, momentum, *weight decay*). Pendekatan ini menawarkan solusi yang lebih efisien secara komputasi dan lebih mudah diimplementasikan dibandingkan metode transformasi citra yang kompleks, sekaligus memanfaatkan keunggulan *inherent* arsitektur YOLO terbaru untuk mencapai akurasi deteksi kendaraan malam hari yang superior.

3. METODE PENELITIAN

Alur penelitian pada studi ini dapat dilihat pada Gambar 1. Diagram tersebut memperlihatkan rangkaian tahapan utama, dimulai dari proses pengumpulan data, pra-pemrosesan, pemilihan serta pengaturan model, dilanjutkan dengan pelatihan, hingga tahap evaluasi dan penarikan kesimpulan.



Gambar 1. Alur penelitian

3.1 Dataset

Penelitian ini menggunakan *dataset* publik UA-DETRAC, yang berisi data pengawasan lalu lintas dengan variasi kondisi cuaca dan waktu. *Dataset* mencakup 9.816 gambar beranotasi *bounding box* untuk setiap kendaraan yang terdeteksi. UA-DETRAC dipilih karena relevan dengan tujuan penelitian, yaitu mengoptimalkan deteksi kendaraan pada kondisi minim cahaya di malam hari. Ketersediaan citra malam yang cukup banyak memungkinkan model mempelajari karakteristik visual khas malam hari seperti pantulan lampu, *glare*, dan rendahnya kontras, sementara citra siang digunakan sebagai pembanding untuk menguji generalisasi terhadap variasi pencahayaan.

3.2 Prapemrosesan Data

Tahap awal meliputi pemilihan *subset* citra malam untuk skenario pertama, menyesuaikan fokus penelitian. Seluruh citra diubah ukurannya menjadi 640×640 piksel sesuai rekomendasi YOLOv11, lalu dibagi menjadi data latih (70%), validasi (20%), dan uji (10%) untuk memisahkan proses pelatihan, pemantauan performa, dan evaluasi akhir.

3.3 Model

Model yang digunakan adalah YOLOv11, generasi terkini dari keluarga YOLO buatan *Ultralytics*, yang dirancang untuk deteksi objek *real-time* dengan akurasi tinggi dan efisiensi komputasi yang baik [3]. Arsitekturnya terdiri dari *backbone* untuk ekstraksi fitur, *neck* untuk penggabungan informasi spasial dan kontekstual, serta *head* untuk menghasilkan prediksi akhir [18]. YOLOv11 mengadopsi desain *anchor-free* yang menyederhanakan proses deteksi dan mengurangi jumlah parameter model secara signifikan dibandingkan pendahulunya [19]. Dibandingkan pendahulunya, YOLOv11 memiliki peningkatan signifikan pada efisiensi dan presisi deteksi, terutama dalam menangani objek-objek dengan skala yang bervariasi [20].

3.4 Tuning Model YOLOv11

Tiga model diuji: YOLOv8, YOLOv11 *baseline*, dan YOLOv11 hasil optimasi. Pemilihan nilai-nilai hiperparameter awal didasarkan pada studi literatur yang telah teruji untuk tugas deteksi objek [1, 7].

Penyesuaian hiperparameter meliputi *initial learning rate* 0.001, *lrf (learning rate final)* 0.1, *momentum* 0.937, dan *weight decay* 0.0005. Nilai *momentum* yang digunakan (0.937) merupakan nilai standar yang direkomendasikan dalam keluarga YOLO untuk mempercepat konvergensi [18]. Augmentasi yang digunakan antara lain penyesuaian saturasi (0.4) dan kecerahan (0.2), translasi (0.1), skala (0.5), *flipping horizontal* (0.5), *mosaic* (1.0), dan *mixup* ringan (0.1). Teknik *mosaic* dan *mixup* terbukti efektif dalam meningkatkan generalisasi model untuk skenario objek yang tumpang tindih dan variasi pencahayaan [6, 11]. Pelatihan dilakukan pada resolusi 640x640 piksel selama 30 *epoch* dengan *batch size* 32 menggunakan *optimizer* AdamW, yang memisahkan mekanisme *weight decay* untuk regularisasi optimal [21].

3.5 Evaluasi

Kinerja model dinilai menggunakan metrik standar deteksi objek, yaitu mean *Average Precision* (*mAP*) pada ambang batas *Intersection over Union* (IoU) sebesar 0.5 (*mAP@0.5*), serta metrik *precision* dan *recall* untuk menilai akurasi dan kelengkapan deteksi [9, 14]. *Precision* menunjukkan persentase prediksi positif yang benar, sedangkan *recall* menggambarkan persentase objek aktual yang berhasil teridentifikasi oleh model. Secara matematis, kedua metrik tersebut dapat dinyatakan dengan rumus berikut:

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

di mana: *True Positive* (TP): jumlah prediksi yang benar terhadap objek yang memang ada; *False Positive* (FP): jumlah prediksi yang salah terhadap objek yang sebenarnya tidak ada; *False Negative* (FN): jumlah objek yang tidak berhasil terdeteksi oleh model.

Adapun *mAP@0.5* diperoleh dari rata-rata *Average Precision* (AP) untuk setiap kelas, dengan ambang *IoU* = 0.5. Artinya, suatu prediksi dianggap benar jika tumpang tindih (*overlap*) antara *bounding box* hasil prediksi dan *ground truth* melebihi 50%. Nilai AP sendiri dihitung berdasarkan luas area di bawah kurva *precision* terhadap *recall*.

$$AP = \int_0^1 P(R) dR \quad (3)$$

Dengan $P(R)$ merupakan fungsi *precision* terhadap *recall*. Selanjutnya, *mean Average Precision* (*mAP*) didefinisikan sebagai nilai rata-rata dari AP untuk seluruh kelas yang ada, sehingga memberikan ukuran keseluruhan performa deteksi model pada berbagai kategori objek.

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (4)$$

Dengan N adalah jumlah total kelas yang terdapat dalam *dataset*.

3.6 Lingkungan Eksperimen

Seluruh eksperimen dilaksanakan pada komputer dengan spesifikasi perangkat keras meliputi prosesor Intel Core i5 generasi ke-12, RAM 16 GB DDR5, dan GPU NVIDIA GeForce RTX 3060 berkapasitas VRAM 12 GB. Sistem operasi yang digunakan adalah Windows 11 64-bit. Pengembangan dan pelatihan model dilakukan menggunakan *framework* PyTorch versi 2.7.1, dengan implementasi YOLO melalui *Ultralytics*, serta dilengkapi berbagai pustaka pendukung lainnya.

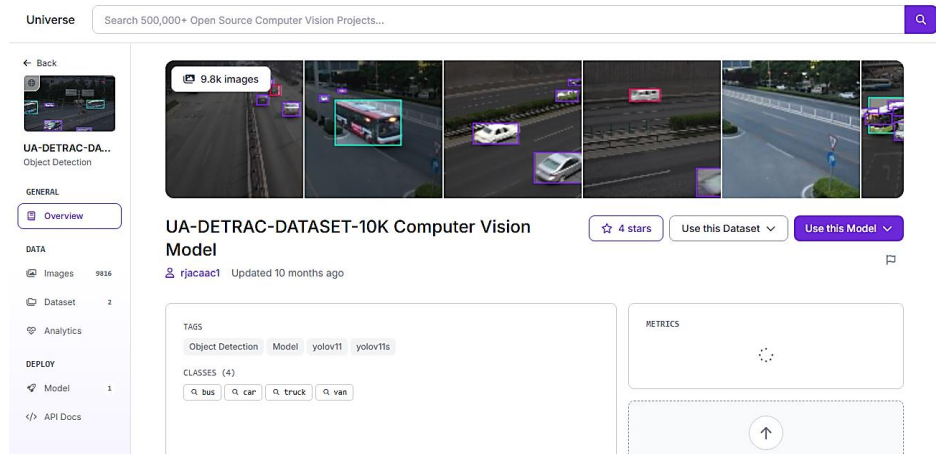
4. HASIL DAN PEMBAHASAN

Untuk mengevaluasi sejauh mana optimasi YOLO11 dapat meningkatkan performa deteksi pada kondisi malam hari sekaligus menguji ketahanannya terhadap perubahan pencahayaan, dilakukan pengujian pada dua skenario berbeda. skenario 1 menggunakan data malam hari secara penuh, sedangkan skenario 2 mengombinasikan data siang dan malam. Pada setiap skenario, tiga model diuji, yaitu YOLO8, YOLO11 versi *baseline*, dan YOLO11 yang telah dioptimasi melalui penyesuaian hiperparameter.

4.1 Pengumpulan Data

Pengumpulan data dalam penelitian ini dilakukan dengan memanfaatkan *dataset* publik UA-DETRAC yang telah tersedia dan terkelola pada platform Universe link: “UA-DETRAC-DATASET-10K Object Detection Model by rjacaac1”, sebuah *repository* terbuka untuk proyek *computer vision*. Pemilihan *dataset* ini didasarkan pada kelengkapannya, anotasi yang andal, dan relevansinya yang tinggi dengan tujuan penelitian untuk deteksi kendaraan.

Seperti yang ditunjukkan pada Gambar 2, *dataset* UA-DETRAC pada *platform Universe Roboflow* dikategorikan sebagai model deteksi objek (*Object Detection*). *Dataset* ini memiliki lebih dari 10.000 gambar dan dianotasi untuk mendeteksi empat kelas kendaraan: *car* (mobil), *bus* (bus), *truck* (truk), dan *van* (van). Model yang tersedia pada platform ini telah dibangun menggunakan arsitektur YOLOv11s, yang menunjukkan keselarasan dengan metode yang digunakan dalam penelitian ini.



Gambar 2. Tempat Pengambilan Data

4.2 Preprocessing Skenario 1 (Data Malam Hari)

Fokus penelitian pada skenario pertama adalah mengoptimalkan deteksi kendaraan secara khusus pada kondisi malam hari. Oleh karena itu, diperlukan *subset* data yang hanya berisi citra-citra malam hari. Proses *preprocessing* untuk skenario ini dimulai dengan pemisahan manual (*manual sorting*) citra malam hari dari keseluruhan *dataset* UA-DETRAC. Pemisahan ini dilakukan melalui *visual inspection* untuk memastikan hanya *frame* yang secara jelas menggambarkan kondisi lingkungan dengan pencahayaan alami minim (kegelapan malam) dengan sumber cahaya dominan berasal dari lampu jalan dan lampu kendaraan yang dimasukkan ke dalam *subset* ini.

Setelah *subset* data malam hari berhasil diisolasi, tahapan *preprocessing* standar diterapkan. Seluruh citra dalam *subset* ini diubah ukurannya (*resized*) menjadi dimensi 640x640 piksel, yang merupakan resolusi *input* standar yang direkomendasikan untuk model YOLOv11 untuk menyeimbangkan antara kecepatan komputasi dan akurasi deteksi. Tahap krusial selanjutnya adalah membagi *dataset* menjadi tiga *subset* untuk fase yang berbeda dalam pengembangan model. Pembagian ini dilakukan secara acak namun terstruktur untuk memastikan bahwa tidak ada kebocoran data antara proses pelatihan, validasi, dan pengujian.

Proporsi pembagian yang digunakan adalah rasio yang umum dalam penelitian *machine learning*: Data Latih (*Training Set* 70%). Digunakan untuk melatih model YOLOv11. Data inilah yang dipelajari oleh model untuk mengenali pola dan fitur kendaraan pada kondisi malam hari. Data Validasi (*Validation Set* 20%). Digunakan untuk mengevaluasi model selama proses pelatihan (*epoch*). Data ini membantu dalam memantau kinerja model pada data yang tidak dilihat selama pelatihan, menginformasikan kapan pelatihan harus dihentikan (*early stopping*), dan sebagai panduan dalam penyetelan hiperparameter. Data Uji (*Test Set* 10%). Digunakan hanya sekali, setelah model selesai dilatih, untuk memberikan evaluasi akhir yang objektif dan tidak bias terhadap performa model yang telah dioptimasi. Hasil metrik (*precision*, *recall*, *mAP*) yang dilaporkan pada bagian hasil berasal dari *subset* ini. Berikut gambar skrip program untuk membagi *dataset* dapat dilihat pada Gambar 3.

4.3 Preprocessing Skenario 2 (Data Gabungan Siang & Malam)

Pada Skenario 2, penelitian bertujuan menguji kemampuan generalisasi model YOLOv11 yang telah dioptimasi dalam mendeteksi kendaraan pada berbagai kondisi pencahayaan. Berbeda dengan skenario 1 yang hanya menggunakan data malam hari, *preprocessing* pada skenario ini menggunakan seluruh *dataset* UA-DETRAC secara utuh tanpa pemisahan antara citra siang dan malam. Pendekatan ini memungkinkan model mempelajari fitur-fitur kendaraan dalam variasi pencahayaan yang luas, sehingga diharapkan dapat berkinerja robust baik di siang maupun malam hari. Seluruh citra dari *dataset* yang mencakup kondisi siang dan malam terkumpul menjadi satu *pool data* homogen.

Seluruh citra dalam *pool data* ini diubah ukurannya menjadi 640x640 piksel untuk konsistensi *input* model YOLOv11 dan efisiensi komputasi. *Pool data* yang telah di *resize* kemudian dibagi secara acak menjadi tiga *subset* dengan proporsi 70% untuk data latih (*training set*), 20% untuk data validasi (*validation set*), dan 10% untuk data uji (*test set*). Pembagian yang dilakukan secara acak ini menjamin setiap *subset*

mengandung representasi proporsional dari kedua kondisi pencahayaan, sehingga evaluasi kinerja model tidak bias terhadap kondisi tertentu. Data latih memastikan model mendapat *exposure* yang cukup terhadap berbagai variasi pencahayaan, data validasi digunakan untuk memantau proses pelatihan dan menyetel hiperparameter, sedangkan data uji mengandung campuran acak dari citra siang dan malam yang merepresentasikan skenario dunia nyata dimana model harus beroperasi secara 24/7. Berikut dapat dilihat pada Gambar 4 skrip program pemisahan atau pembagian datanya.

```
# Path ke dataset
dataset_dir = "dataset/UNSWTHIRDCAMVIDEODATASET" # ubah sesuai lokasi
images_dir = os.path.join(dataset_dir, "JPEGImages")
annotations_dir = os.path.join(dataset_dir, "Annotations")
image_sets_dir = os.path.join(dataset_dir, "ImageSets", "Main")

os.makedirs(image_sets_dir, exist_ok=True)

# Ambil semua nama file gambar (tipe ekstensi)
all_files = [os.path.splitext(f)[0] for f in os.listdir(images_dir) if f.endswith('.jpg')]

# Shuffle untuk random split
random.shuffle(all_files)

# Split ratio
train_ratio = 0.7
val_ratio = 0.2
test_ratio = 0.1

n_total = len(all_files)
n_train = int(n_total * train_ratio)
n_val = int(n_total * val_ratio)

train_files = all_files[:n_train]
val_files = all_files[n_train:n_train+n_val]
test_files = all_files[n_train+n_val:]

# Simpan ke file txt
def save_list(file_list, filename):
    with open(os.path.join(image_sets_dir, filename), 'w') as f:
        for item in file_list:
            f.write(item + '\n')

save_list(train_files, "train.txt")
save_list(val_files, "val.txt")
save_list(test_files, "test.txt")

print(f"Total Images: {n_total}")
print(f"Train: {len(train_files)}, Val: {len(val_files)}, Test: {len(test_files)}")
print(f"File list saved in: {image_sets_dir}")
```

Gambar 3. Membagi Dataset Skenario 1

```
# Path ke dataset
dataset_dir = "dataset/UNSWTHIRDCAMVIDEODATASET" # ubah sesuai lokasi
images_dir = os.path.join(dataset_dir, "JPEGImages")
annotations_dir = os.path.join(dataset_dir, "Annotations")
image_sets_dir = os.path.join(dataset_dir, "ImageSets", "Main")

os.makedirs(image_sets_dir, exist_ok=True)

# Ambil semua nama file gambar (tipe ekstensi)
all_files = [os.path.splitext(f)[0] for f in os.listdir(images_dir) if f.endswith('.jpg')]

# Shuffle untuk random split
random.shuffle(all_files)

# Split ratio
train_ratio = 0.7
val_ratio = 0.2
test_ratio = 0.1

n_total = len(all_files)
n_train = int(n_total * train_ratio)
n_val = int(n_total * val_ratio)

train_files = all_files[:n_train]
val_files = all_files[n_train:n_train+n_val]
test_files = all_files[n_train+n_val:]

# Simpan ke file txt
def save_list(file_list, filename):
    with open(os.path.join(image_sets_dir, filename), 'w') as f:
        for item in file_list:
            f.write(item + '\n')

save_list(train_files, "train.txt")
save_list(val_files, "val.txt")
save_list(test_files, "test.txt")

print(f"Total Images: {n_total}")
print(f"Train: {len(train_files)}, Val: {len(val_files)}, Test: {len(test_files)}")
print(f"File list saved in: {image_sets_dir}")
```

Gambar 4. Membagi Dataset Skenario 2

4.4 Parameter

Konfigurasi parameter memegang peran krusial dalam proses optimasi model untuk mencapai performa deteksi yang optimal. Pada penelitian ini, parameter dikelompokkan menjadi dua bagian utama: Parameter Model yang berkaitan dengan hiperparameter pelatihan dan parameter augmentasi yang mengatur teknik manipulasi data.

Nilai-nilai hiperparameter pada proses pelatihan model YOLOv11 ditentukan melalui kajian literatur dan serangkaian eksperimen pendahuluan untuk memastikan stabilitas konvergensi dan mencegah fenomena *overfitting*. Konfigurasi akhir yang diterapkan adalah sebagai berikut. Nilai *initial learning rate* ditetapkan sebesar 0.001 sebagai titik awal yang optimal untuk memfasilitasi pembaruan bobot model yang stabil tanpa menyebabkan divergensi. *Learning rate* ini dikelola menggunakan scheduler yang menurunkan nilainya secara kosinus hingga mencapai final *learning rate* (lrf) sebesar 0.1 pada *epoch* akhir, yang memungkinkan model untuk melakukan penyempurnaan (*fine-tuning*) yang lebih halus pada tahap akhir pelatihan. Parameter momentum digunakan pada 0.937 untuk mempercepat proses pelatihan dalam arah yang benar dan mengurangi osilasi dengan mempertimbangkan gradien dari langkah-langkah sebelumnya. Nilai *weight decay* sebesar 0.0005 diterapkan sebagai teknik regularisasi L2 untuk membatasi kompleksitas model dengan memberikan *penalty* pada bobot yang bernilai besar, sehingga meningkatkan kemampuan generalisasi model. Pelatihan dilakukan secara lengkap sebanyak 30 *epoch* dengan *batch size* 32 untuk menyeimbangkan antara kebutuhan komputasi dan kapasitas model. Algoritma AdamW (Adam dengan *Decoupled Weight Decay*) dipilih sebagai *optimizer* karena kemampuannya dalam mengelola *weight decay* secara terpisah dari *learning rate*, sehingga memberikan regularisasi yang lebih efektif dan stabil dibandingkan *optimizer* tradisional.

Di sisi lain, teknik augmentasi data diterapkan secara agresif untuk secara artifisial memperkaya variasi *dataset* latih, khususnya mensimulasikan beragam tantangan visual yang ditemui pada kondisi malam hari. Strategi ini dirancang untuk meningkatkan ketahanan (*robustness*) model terhadap perubahan lingkungan. Konfigurasi augmentasi yang digunakan meliputi penyesuaian pada ruang warna dan pencahayaan, yaitu saturasi dengan faktor 0.4 dan kecerahan dengan faktor 0.2 untuk membantu model beradaptasi dengan variasi warna dan intensitas cahaya dari lampu kendaraan dan jalanan. Transformasi geometris juga diterapkan, termasuk translasi sebesar 0.1 dan scaling sebesar 0.5 untuk membuat model tidak sensitif terhadap perubahan posisi dan ukuran objek. Teknik *flipping horizontal* dengan probabilitas 0.5 digunakan untuk memperkenalkan variasi orientasi kendaraan. Selanjutnya, teknik augmentasi modern yang khas pada arsitektur YOLO juga dimanfaatkan secara penuh. Teknik mosaik dengan probabilitas 1.0 menggabungkan empat gambar latih menjadi satu mosaik, yang memaksa model untuk mempelajari konteks objek dalam skenario yang lebih kompleks dan dengan berbagai skala. Terakhir, *mixup* ringan dengan probabilitas 0.1 diterapkan untuk melakukan interpolasi linier antara dua gambar dan labelnya, yang telah terbukti dapat meningkatkan generalisasi dan *smoothness decision boundary* model. Kombinasi dari seluruh teknik augmentasi ini secara kolektif berfungsi untuk mengekspos model pada spektrum kondisi yang sangat luas, sehingga menjadikannya lebih tangguh dalam mendeteksi kendaraan dalam lingkungan pencahayaan rendah yang kompleks.

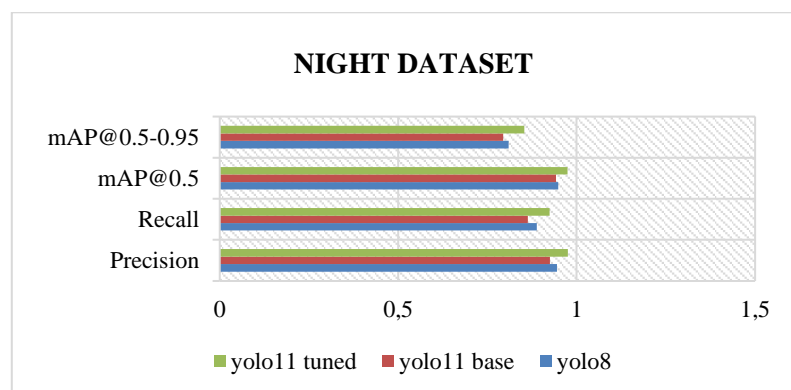
4.5 Hasil Pengujian Skenario 1 (Data Malam Hari)

Hasil pengujian untuk Skenario 1 disajikan pada Tabel 1, dengan perbandingan kinerja model divisualisasikan pada Gambar 5. Pengujian ini memanfaatkan *subset* data malam hari dari *dataset* UA-DETRAC dengan tujuan mengukur kemampuan model mendeteksi kendaraan pada pencahayaan rendah. Berdasarkan hasil evaluasi pada Tabel 1, YOLO11 teroptimasi menunjukkan performa terbaik dibandingkan dua model lainnya, dengan *precision* 0,97, *recall* 0,92, dan *mAP@0.5* sebesar 0,97. Sebagai perbandingan, YOLO8 memperoleh *precision* 0,94, *recall* 0,88, dan *mAP* 0,94, sementara YOLO11 *baseline* mencatat *precision* 0,92, *recall* 0,86, dan *mAP* 0,94.

Tabel 1. Hasil Evaluasi Model pada Skenario 1 (Data Malam Hari)

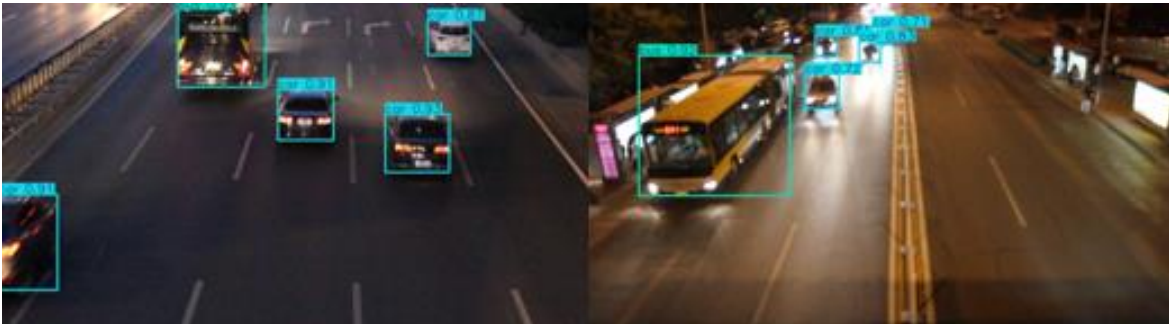
Model	<i>Precision</i>	<i>Recall</i>	<i>mAP</i>
YOLO8	0.94	0.88	0.94
YOLO11	0.92	0.86	0.94
Tuned YOLO11	0.97	0.92	0.97

Peningkatan kinerja yang paling menonjol pada YOLO11 hasil optimasi terlihat pada aspek *recall*, yang naik dari 0,86 pada model *baseline* menjadi 0,92. Peningkatan ini menunjukkan bahwa model teroptimasi mampu mengenali lebih banyak objek kendaraan pada kondisi malam hari, sehingga mengurangi jumlah *false negative*. Visualisasi pada Gambar 5 menampilkan bahwa YOLO11 teroptimasi dapat menjaga konsistensi deteksi meskipun objek berada di area dengan pencahayaan minim, terkena pantulan cahaya lampu, atau mengalami variasi intensitas cahaya dari kendaraan.



Gambar 5. Menampilkan Grafik Perbandingan Nilai *Precision*, *Recall*, Dan *mAP* Pada Skenario 1.

Sementara itu, contoh hasil deteksi kendaraan pada kondisi malam hari menggunakan YOLOv11 yang telah dioptimasi ditunjukkan pada Gambar 6. Visualisasi tersebut memperlihatkan bahwa model mampu melakukan deteksi kendaraan secara konsisten meskipun berada dalam lingkungan dengan pencahayaan terbatas, termasuk kondisi lampu jalan redup, adanya pantulan cahaya, serta variasi intensitas penerangan dari lampu kendaraan.



Gambar 6. Contoh Hasil Deteksi Menggunakan YOLOv11 yang Telah Dioptimasi

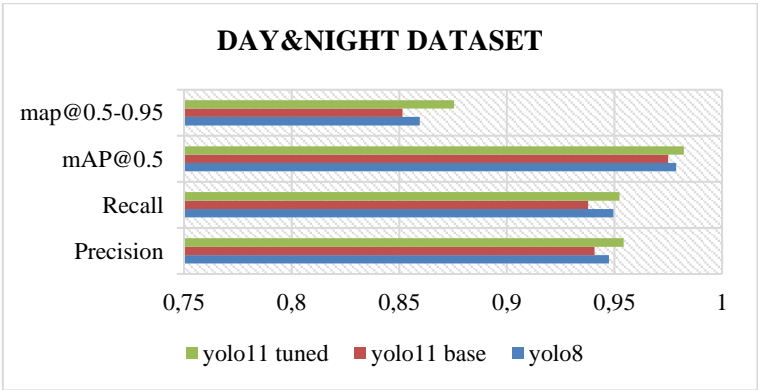
4.6 Hasil Pengujian Skenario 2 (Data Gabungan Siang dan Malam)

Pada Skenario 2, pelatihan dan pengujian dilakukan menggunakan *dataset* gabungan yang mencakup citra pada siang dan malam hari. Tujuan dari skenario ini adalah untuk menilai kemampuan model dalam melakukan generalisasi pada kondisi pencahayaan yang bervariasi. Hasil evaluasi yang disajikan pada Tabel 2 menunjukkan bahwa seluruh model mengalami peningkatan performa dibandingkan Skenario 1. YOLOv11 yang telah dioptimasi kembali menorehkan kinerja tertinggi dengan *precision* sebesar 0,95, *recall* sebesar 0,95, dan *mAP@0.5* sebesar 0,98. Sebagai perbandingan, YOLOv8 mencatat *precision* 0,94, *recall* 0,94, dan *mAP* 0,97, sementara YOLOv11 *baseline* meraih *precision* 0,94, *recall* 0,93, dan *mAP* 0,97.

Tabel 2. Hasil Evaluasi Model pada Skenario 2 (Keseluruhan Dataset)

Model	Precision	Recall	mAP
YOLOv8	0.94	0.88	0.94
YOLOv11	0.92	0.86	0.94
Tuned YOLOv11	0.97	0.92	0.97

Perbedaan performa antar model pada skenario ini lebih kecil dibandingkan dengan Skenario 1, yang menunjukkan bahwa keberagaman kondisi pencahayaan dalam data mampu meningkatkan kemampuan generalisasi model. Meskipun demikian, YOLOv11 yang telah dioptimasi tetap menunjukkan keunggulan, terutama pada nilai *recall* yang tinggi, yang mengindikasikan kemampuannya dalam mendeteksi objek secara konsisten pada kondisi siang maupun malam. Visualisasi hasil deteksi pada Gambar 7 memperlihatkan bahwa model ini dapat mempertahankan tingkat presisi yang tinggi meskipun menghadapi variasi latar belakang, tingkat kepadatan lalu lintas, serta perbedaan intensitas pencahayaan.



Gambar 7. Menampilkan Grafik Perbandingan Nilai *Precision*, *Recall*, Dan *mAP* Pada Skenario 2.

4.7 Analisis Performa Model

Pengujian pada kedua skenario menunjukkan bahwa YOLOv11 hasil optimasi secara konsisten unggul dibandingkan YOLOv8 dan YOLOv11 *baseline*. Peningkatan paling signifikan terlihat pada nilai *recall* di Skenario 1 (data malam hari), di mana skor naik dari 0,86 pada YOLOv11 *baseline* menjadi 0,92 setelah penerapan tuning hiperparameter dan augmentasi data. Hal ini menunjukkan bahwa strategi optimasi yang digunakan berhasil mengurangi jumlah kendaraan yang tidak terdeteksi (*false negative*) pada kondisi pencahayaan rendah. Pada Skenario 2 (data gabungan siang dan malam), selisih performa antar model memang lebih kecil, namun YOLOv11 hasil optimasi tetap mempertahankan keunggulan baik pada *precision* maupun *recall*. Performa yang tetap tinggi di skenario ini membuktikan bahwa augmentasi adaptif dan penyesuaian hiperparameter tidak hanya efektif untuk kondisi malam hari, tetapi juga mampu menjaga

kualitas deteksi di siang hari. Secara keseluruhan, temuan ini selaras dengan hasil penelitian [6], [7], [15], [16] yang menegaskan bahwa kombinasi tuning hiperparameter dan augmentasi data adaptif dapat meningkatkan kemampuan model deteksi objek, khususnya di lingkungan dengan tantangan pencahayaan rendah. Meski demikian, penelitian ini memiliki keterbatasan, seperti penggunaan hanya satu jenis *dataset* (UA-DETRAC). Studi lanjutan disarankan untuk memanfaatkan *dataset* multi-sumber guna menguji skalabilitas metode yang diusulkan.

5. DISKUSI

Hasil pengujian pada kedua skenario menunjukkan bahwa optimasi YOLOv11 melalui kombinasi hiperparameter tuning dan augmentasi data adaptif berhasil meningkatkan akurasi deteksi kendaraan secara signifikan, khususnya dalam kondisi pencahayaan rendah. Peningkatan tertinggi terlihat pada nilai *recall* dalam Skenario 1 (data malam hari), yang naik dari 0,86 pada model *baseline* menjadi 0,92 setelah optimasi. Hal ini menunjukkan bahwa model yang dioptimasi mampu mengurangi jumlah *false negative* secara drastis, sehingga lebih banyak kendaraan yang berhasil terdeteksi meskipun dalam kondisi visual yang menantang seperti silau lampu, kontras rendah, dan bayangan gelap.

Temuan ini sejalan dengan penelitian sebelumnya yang berjudul “*Improved YOLOv8 algorithm for vehicle image target detection based on learning rate optimisation strategy*,” yang menyoroti pentingnya optimasi *learning rate* pada YOLOv8 untuk meningkatkan deteksi kendaraan. Namun, penelitian ini melangkah lebih jauh dengan menggabungkan tidak hanya penyesuaian hiperparameter tetapi juga augmentasi data yang secara khusus dirancang untuk kondisi malam hari. Teknik seperti mosaic dan mixup terbukti efektif dalam memperkenalkan variasi pencahayaan dan konteks yang mirip dengan dunia nyata, sehingga model menjadi lebih robust. Pendekatan ini juga konsisten dengan temuan pada penelitian berjudul “*Research on Real-Time Detection Algorithm for Pedestrian and Vehicle in Foggy Weather Based on Lightweight XM-YOLOVIT*”, dan “*A Multi-Module Network For Robust Vehicle Detection In Low-Light Conditions*,” yang menekankan perlunya arsitektur dan augmentasi yang adaptif terhadap kondisi cahaya rendah.

Keunggulan model yang diusulkan tidak hanya terbatas pada kondisi malam hari. Pada Skenario 2 (gabungan siang dan malam), YOLOv11 yang dioptimasi tetap mempertahankan kinerja terbaik dengan *precision* 0,95, *recall* 0,95, dan *mAP@0.5* 0,98. Ini membuktikan bahwa strategi optimasi yang diterapkan tidak mengorbankan kinerja model dalam kondisi siang hari, sehingga cocok untuk sistem pemantauan lalu lintas yang beroperasi 24/7. Namun, penelitian ini memiliki beberapa keterbatasan. Pertama, eksperimen hanya dilakukan pada satu *dataset* (UA-DETRAC). Meskipun *dataset* ini cukup komprehensif, generalisasi temuan ke *dataset* lain atau kondisi lingkungan yang lebih beragam (seperti kabut, hujan, atau salju) masih perlu diuji lebih lanjut. Kedua, augmentasi yang digunakan meskipun efektif, mungkin belum sepenuhnya menangkap semua variasi kondisi nyata di malam hari, seperti pantulan cahaya ekstrem atau kendaraan dengan lampu tidak standar. Ke depan, penelitian dapat dikembangkan dengan menguji model pada *dataset* multi-sumber dan multi-lingkungan untuk menilai skalabilitasnya. Selain itu, integrasi dengan teknik *image enhancement* khusus kondisi gelap atau penggunaan arsitektur yang lebih ringan untuk aplikasi *real-time* pada perangkat terbatas dapat menjadi arah eksplorasi berikutnya. Secara keseluruhan, optimasi YOLOv11 yang diusulkan menawarkan solusi yang efektif, efisien, dan mudah diimplementasikan untuk meningkatkan deteksi kendaraan malam hari tanpa mengorbankan kinerja siang hari. Temuan ini berkontribusi penting bagi pengembangan sistem transportasi cerdas yang andal dan konsisten dalam berbagai kondisi pencahayaan.

6. KESIMPULAN

Hasil penelitian ini mengungkap bahwa optimasi YOLO11 melalui kombinasi tuning hiperparameter dan strategi augmentasi data mampu secara signifikan meningkatkan akurasi deteksi kendaraan, khususnya pada kondisi malam hari. Berdasarkan pengujian pada *dataset* UA-DETRAC, YOLO11 yang telah dioptimasi mencapai *precision* 0,97, *recall* 0,92, dan *mAP@0.5* sebesar 0,97 pada skenario malam melampaui performa YOLO8 dan YOLO11 *baseline*. Peningkatan ini terutama dipengaruhi oleh penerapan teknik augmentasi seperti penyesuaian warna, translasi, *scaling*, *flipping*, *mosaic*, dan *mixup*, yang memperkaya variasi data latih sehingga model lebih adaptif terhadap pencahayaan rendah. Selain itu, penyesuaian parameter seperti *learning rate*, momentum, dan *weight decay* meningkatkan stabilitas pelatihan, memungkinkan model mempelajari pola visual secara efektif tanpa overfitting. Temuan ini konsisten dengan penelitian sebelumnya yang menegaskan bahwa perpaduan tuning hiperparameter dan augmentasi data mampu meningkatkan kinerja deteksi objek di lingkungan yang menantang. Ke depan, metode ini berpotensi diuji pada *dataset* multi-sumber dan berbagai kondisi lingkungan untuk menilai skalabilitas serta ketahanannya. Secara keseluruhan, optimasi YOLO11 yang diusulkan menawarkan solusi efektif dan praktis untuk deteksi kendaraan di malam hari sekaligus menjaga akurasi pada siang hari, sehingga dapat menjadi kontribusi penting bagi pengembangan sistem pemantauan lalu lintas berbasis visi komputer yang andal di beragam kondisi pencahayaan.

REFERENSI

- [1] G. D. Deepak and S. K. Bhat, "Optimizing YOLOv4 Hyperparameters for Enhanced Vehicle Detection in Intelligent Transportation Systems," *International Journal of Intelligent Transportation Systems Research*, 2025, doi: 10.1007/s13177-025-00519-3.
- [2] M. Humyam, T. Ashfaq, N. Z. Jhanjhi, and M. K. Alsadun, "Traffic Management: Multi-Scale Vehicle Detection in Varying Weather Conditions Using YOLOv4 and Spatial Pyramid Pooling Network," *Electronics (Switzerland)*, vol. 11, no. 17, Sep. 2022, doi: 10.3390/electronics11172748.
- [3] S.-W. Huang, C.-T. Lin, S.-P. Chen, Y.-Y. Wu, P.-H. Hsu, and S.-H. Lai, "AugGAN: Cross Domain Adaptation with GAN-based Data Augmentation."
- [4] Y. Hu, "Improved YOLOv8 algorithm for vehicle image target detection based on learning rate optimization strategy," *Applied and Computational Engineering*, vol. 75, pp. 248-253, Aug. 2024, doi: 10.54254/2755-2721/7520240550.
- [5] H. Kumdake, C. Ongün, and A. Temizel, "Generative Data Augmentation for Vehicle Detection in Aerial Images," Dec. 2020, [Online]. Available: <http://arxiv.org/abs/2012.04902>.
- [6] C. T. Lin, S. W. Huang, Y. Y. Wu, and S. H. Lai, "GAN-Based Day-to-Night Image Style Transfer for Nighttime Vehicle Detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 27, pp. 951-963, Feb. 2021, doi: 10.1109/TITS.2019.2961679.
- [7] A. Murumi and F. Murumi, "Data augmentation: A comprehensive survey of modern approaches," Dec. 01, 2022, Elsevier B.V. doi: 10.1016/j.array.2022.100258.
- [8] V. Ostankovich, R. Yagfarov, M. Rassabin, and S. Gafurov, "Application of CycleGAN-based Augmentation for Autonomous Driving at Night," in *2020 International Conference Nonlinearity, Information and Robotics, NIR 2020*, Institute of Electrical and Electronics Engineers Inc., Dec. 2020. doi: 10.1109/NIR50484.2020.9290218.
- [9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," May 2016, [Online]. Available: <http://arxiv.org/abs/1506.02640>.
- [10] Y. Said, Y. Alassaf, R. Ghodhban, T. Saidani, and O. Ben Rhaïem, "Optimized Convolutional Neural Networks with Multi-Scale Pyramid Feature Integration for Efficient Traffic Light Detection in Intelligent Transportation Systems," *Computers, Materials and Continua*, vol. 82, no. 2, pp. 3005-3018, 2025, doi: 10.32604/cmc.2025.060928.
- [11] D. Sun, F. Dornaika, and N. Barrena, "HSMix: Hard and soft mixing data augmentation for medical image segmentation," *Information Fusion*, vol. 115, Mar. 2025, doi: 10.1016/j.inffus.2024.102741.
- [12] D. Q. Tran, A. Aboah, Y. Jeon, M. Shoman, M. Park, and S. Park, "Low-Light Image Enhancement Framework for Improved Object Detection in Fisheye Lens Datasets," Apr. 2024, [Online]. Available: <http://arxiv.org/abs/2404.10078>.
- [13] Q. Yu, H. Liu, and Q. Wu, "An Improved YOLO for Road and Vehicle Target Detection Model," *Journal of ICT Standardization*, vol. 11, no. 2, pp. 197-216, 2023, doi: 10.13052/jicts2245-800X.1125.
- [14] Z. Yang and L. S. C. Pun-Cheng, "Vehicle detection in intelligent transportation systems and its applications under varying environments: A review," Jan. 01, 2018, Elsevier Ltd. doi: 10.1016/j.imavis.2017.09.008.
- [15] H. Zhang, Y. Gong, F. Yao, and Q. Zhang, "Research on Real-Time Detection Algorithm for Pedestrian and Vehicle in Foggy Weather Based on Lightweight XM-YOLOVIT," *IEEE Access*, vol. 12, pp. 7864-7883, 2024, doi: 10.1109/ACCESS.2023.3344666.
- [16] Q. Zhang, W. Guo, and M. Lin, "LLD-YOLO: a multi-module network for robust vehicle detection in low-light conditions," *Signal Image Video Process*, vol. 19, Aug. 2025, doi: 10.1007/s11760-025-03858-6.
- [17] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 6, pp. 1328-1336, June 2020, [Online]. Available: <https://arxiv.org/abs/1804.02767>.
- [18] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *Computer Vision and Image Understanding*, vol. 212, Sep. 2021, Art. no. 103250, <https://arxiv.org/abs/2004.10934>.
- [19] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," *Pattern Recognition Letters*, vol. 174, pp. 12-19, Oct. 2023, <https://arxiv.org/abs/2207.02696>.
- [20] J. Terven, D. M. Córdova-Esparza, and J.-A. Romero-González, "A Comprehensive Review of YOLO: From YOLOv1 to YOLOv8 and Beyond," *Archives of Computational Methods in Engineering*, vol. 30, no. 4, pp. 2113-2130, May 2023, <https://arxiv.org/abs/2304.00501>.
- [21] J. Cao, T. Zhang, L. Hou, and N. Nan, "An improved YOLOv8 algorithm for small object detection in autonomous driving," *Journal of Real-Time Image Processing*, vol. 21, no. 4, Jul. 2024, doi: 10.1007/s11554-024-01517-6.