

# Evaluasi Robustness dan Efisiensi Model CNN Modern untuk Deteksi Objek di Bawah Degradasi Visual

A. Widhiariandoko<sup>1</sup>, A. Syukur<sup>2</sup>, dan M.A. Soeleman<sup>3</sup>

Program Magister Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro, Semarang E-mail :  
adhitya218@gmail.com<sup>1</sup>, abdul\_s@dosen.dinus.ac.id<sup>2</sup>, m.arief.soeleman@dsn.dinus.ac.id<sup>3</sup>

**Abstract**—Object detection models based on Convolutional Neural Networks (CNN) typically demonstrate high performance on benchmark datasets but often experience significant degradation in real-world conditions containing visual disturbances. This research aims to analyze and compare the efficiency and accuracy of three representative CNN architectures—Faster R-CNN (two-stage), SSD (classic one-stage), and YOLOv8s (modern one-stage)—under visual degradation. The models were trained from scratch on a COCO dataset subset and evaluated across four scenarios: clean images, Gaussian noise, motion blur, and a combination of both. Performance was measured using mean Average Precision (mAP) for accuracy and Frames Per Second (FPS) for efficiency. The results indicate that YOLOv8s is not only the fastest (476.19 FPS) but also achieved the highest accuracy in clean conditions (mAP 0.2870) and demonstrated superior robustness, with an accuracy drop of only 51.4% under motion blur compared to over 79% for other models. This study contributes to the literature by highlighting the resilience of modern anchor-free architectures against pixel-level noise compared to classic anchor-based approaches. These findings recommend YOLOv8s as a robust solution for real-time industrial surveillance systems prone to visual impairments

**Keywords**—Convolutional Neural Networks (CNN), Model Evaluation, Object Detection, Robustness, Visual Noise, YOLOv8s.

**Abstrak**—Model deteksi objek berbasis Jaringan Saraf Konvolusional (CNN) umumnya menunjukkan performa tinggi pada dataset standar, namun seringkali mengalami penurunan performa signifikan pada kondisi dunia nyata yang mengandung gangguan visual. Penelitian ini bertujuan untuk menganalisis dan membandingkan efisiensi serta akurasi dari tiga arsitektur CNN representatif—Faster R-CNN (dua tahap), SSD (satu tahap klasik), dan YOLOv8s (satu tahap modern)—di bawah kondisi degradasi visual. Model-model tersebut dilatih dari awal (from scratch) pada subset dataset COCO dan diuji pada empat skenario: citra bersih, gangguan Gaussian, motion blur, dan kombinasi keduanya. Evaluasi performa menggunakan metrik mean Average Precision (mAP) untuk akurasi dan Frames Per Second (FPS) untuk efisiensi. Hasil eksperimen menunjukkan bahwa YOLOv8s tidak hanya menjadi model tercepat (476.19 FPS), tetapi juga mencapai akurasi tertinggi pada kondisi bersih (mAP 0.2870) dan menunjukkan ketahanan (robustness) terbaik terhadap semua jenis gangguan. Sebagai contoh, pada gangguan motion blur, akurasi YOLOv8s hanya turun 51.4%, jauh lebih rendah dibandingkan penurunan model lain yang mencapai lebih dari 79%. Penelitian ini memberikan kontribusi dalam membuktikan keunggulan ketahanan arsitektur modern berbasis anchor-free terhadap noise level piksel dibandingkan pendekatan klasik. Temuan ini merekomendasikan penggunaan YOLOv8s untuk sistem pengawasan industri real-time yang rentan terhadap gangguan visual.

**Kata Kunci**—Convolutional Neural Networks (CNN), Deteksi Objek, Evaluasi Model, Gangguan Visual, Ketahanan, YOLOv8s.

## I. PENDAHULUAN

Deteksi objek merupakan salah satu tugas fundamental dalam visi komputer yang bertujuan untuk mengidentifikasi dan melokalisasi objek dalam gambar atau video. Dalam dekade terakhir, pendekatan berbasis Deep Learning, khususnya Convolutional Neural Network (CNN), telah menjadi standar industri dalam penyelesaian tugas ini [1]. Perkembangan arsitektur CNN telah melahirkan berbagai varian model yang menawarkan keseimbangan berbeda antara akurasi dan kecepatan, mulai dari pendekatan dua tahap (two-stage) seperti Faster R-CNN [2] hingga pendekatan satu tahap (one-stage) seperti SSD [3] dan keluarga YOLO [4].

Penerapan CNN telah meluas ke berbagai domain praktis, seperti sistem manajemen parkir cerdas yang menggunakan YOLO untuk mendeteksi kendaraan secara real-time [5]. Namun, tantangan utama muncul ketika model-model ini diterapkan di lingkungan dunia nyata yang tidak ideal. Kondisi lingkungan seperti pencahayaan rendah, cuaca buruk, atau pergerakan kamera seringkali menimbulkan

gangguan visual (visual noise) pada citra masukan. Gangguan ini dapat berupa Gaussian noise akibat sensitivitas sensor kamera pada kondisi minim cahaya, atau motion blur akibat pergerakan objek yang cepat.

Studi terdahulu oleh Rusak et al. [6] menunjukkan bahwa detektor objek modern sangat rentan terhadap kerusakan citra, di mana performa model dapat menurun drastis saat diuji pada data yang terdegradasi. Hal ini diperkuat oleh penelitian Wibowo et al. [7] yang menekankan pentingnya pengujian reliabilitas deteksi objek pada berbagai kondisi lingkungan untuk memastikan kehandalan sistem. Meskipun kerentanan ini telah diketahui, masih terdapat celah penelitian (research gap) terkait perbandingan ketahanan (robustness) antara arsitektur klasik berbasis anchor (seperti Faster R-CNN dan SSD) melawan arsitektur modern berbasis anchor-free (seperti YOLOv8) secara langsung pada jenis gangguan spesifik. Kebanyakan studi komparatif hanya berfokus pada metrik standar pada dataset bersih, tanpa mengevaluasi stabilitas model di bawah tekanan noise sintetik yang mensimulasikan kondisi lapangan.

Oleh karena itu, penelitian ini bertujuan untuk mengevaluasi dan membandingkan secara komprehensif performa tiga model deteksi objek (Faster R-CNN, SSD, dan YOLOv8s) dalam menghadapi degradasi visual. Kontribusi utama penelitian ini adalah: (1) Menyediakan analisis empiris mengenai ketahanan arsitektur modern YOLOv8s dibandingkan pendahulunya; (2) Mengevaluasi trade-off antara efisiensi dan akurasi pada kondisi ekstrim; dan (3) Memberikan rekomendasi praktis untuk pemilihan model pada sistem visi komputer yang beroperasi di lingkungan dinamis.

## II. METODE PENELITIAN

Penelitian ini dirancang sebagai sebuah studi eksperimen kuantitatif komparatif. Alur kerja penelitian dirangkum dalam Gambar 1



Gambar 1 : Diagram alur metodologi penelitian

### A. Arsitektur Model

Tiga model dipilih untuk mewakili pendekatan arsitektur yang berbeda:

1. **Faster R-CNN**: Model dua tahap yang menggunakan *Region Proposal Network* (RPN). Dikenal memiliki akurasi tinggi namun inferensi lambat [2].
2. **SSD (Single Shot MultiBox Detector)**: Model satu tahap klasik yang melakukan prediksi dari beberapa feature map berskala berbeda. Menawarkan kompromi kecepatan dan akurasi [3].
3. **YOLOv8s**: Model satu tahap modern yang menggunakan desain anchor-free dan backbone C2f. Dirancang untuk kecepatan dan akurasi yang sangat tinggi [4].

### B. Dataset dan Parameter Pelatihan

Dataset Penelitian menggunakan subset dari dataset MS COCO (Microsoft Common Objects in Context) [8] sebanyak 50.000 citra untuk pelatihan. Pemilihan subset ini dilakukan untuk menjaga efisiensi waktu pelatihan dengan

tetap mempertahankan representasi keragaman kelas objek. Subset ini mencakup 80 kategori objek sesuai standar COCO. Dataset dibagi secara acak dengan rasio 80:20 untuk set pelatihan dan validasi. Augmentasi data standar (seperti Mosaic pada YOLO dan random flip) diaktifkan selama pelatihan untuk meningkatkan variabilitas data.

Ketiga model dilatih dari awal (trained from scratch) tanpa menggunakan bobot pre-trained untuk memastikan perbandingan yang adil pada proses pembelajaran fitur. Konfigurasi hyperparameter pelatihan disajikan pada Tabel 1. Pelatihan dilakukan menggunakan perangkat keras GPU NVIDIA GeForce RTX 4060 (8GB VRAM) dan CPU Intel Core i5-12700F.

Parameter	Nilai	Keterangan
Jumlah Epoch	10	
Batch Size	4	
Image Size	512x512 (SSD), 512x512 (YOLOv8), 512x512 (Faster R-CNN)	
Device	GPU (NVIDIA RTX 4060)	
Framework	PyTorch (SSD, Faster R-CNN), Ultralytics YOLOv8	

Tabel 1. Konfigurasi *hyperparameter* pelatihan model

### C. Skenario Pengujian dan Simulasi Gangguan

Untuk menguji ketahanan model, dataset uji dimodifikasi dengan menambahkan gangguan visual sintetis yang merepresentasikan kondisi nyata:

1. **Bersih**: Citra uji asli tanpa modifikasi.
2. **Gaussian Noise**: Citra uji diberi gangguan Gaussian dengan variansi  $\sigma^2 = 0.05$  untuk menyimulasikan *noise* sensor.
3. **Motion Blur**: Citra uji diberi *kernel blur* linear dengan panjang 15 piksel untuk menyimulasikan gerakan.
4. **Kombinasi**: Citra uji diberi gabungan gangguan Gaussian dan *motion blur* untuk menyimulasikan skenario ekstrem.



Gambar 2 : Visualisasi sampel dataset uji: (a) Citra bersih, (b) Gaussian noise, (c) Motion blur, dan (d) Kombinasi gangguan

### D. Metrik Evaluasi

Performa model dievaluasi menggunakan dua metrik utama:

1. **Mean Average Precision (mAP)**: Mengukur akurasi deteksi berdasarkan Intersection over Union (IoU). Penelitian ini menggunakan  $mAP@[.50:.95]$  sesuai

standar COCO yang merupakan rata-rata mAP pada berbagai ambang batas IoU.

2. Frames Per Second (FPS): Mengukur kecepatan inferensi model, dihitung dari rata-rata waktu yang dibutuhkan untuk memproses satu bingkai gambar pada GPU yang digunakan.

### III. HASIL DAN PEMBAHASAN

Bagian ini menyajikan hasil eksperimen dan analisis mendalam mengenai performa ketiga model pada berbagai skenario uji. Rangkuman hasil kuantitatif dapat dilihat pada Tabel 2.

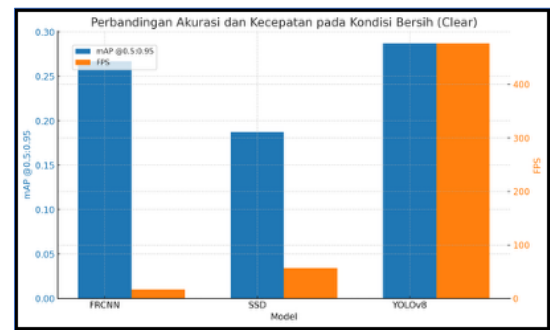
Model	Kondisi Gangguan	Akurasi (mAP @.50)	Akurasi (mAP)	Kecepatan (FPS)
Faster R-CNN	Bersih (Tanpa Gangguan)	0.4406	0.2671	16.92
	Gaussian Noise	0.0118	0.0071	17.33
	Motion Blur	0.1053	0.0538	16.94
	Kombinasi	0.0025	0.0017	17.29
SSD	Bersih (Tanpa Gangguan)	0.3410	0.1872	57.02
	Gaussian Noise	0.0268	0.0130	59.53
	Motion Blur	0.0957	0.0481	58.51
	Kombinasi	0.0098	0.0054	60.43
YOLOv8s	Bersih (Tanpa Gangguan)	0.4250	0.2870	476.19
	Gaussian Noise	0.1440	0.0867	476.19
	Motion Blur	0.2160	0.1390	476.19
	Kombinasi	0.0354	0.0199	476.19

Tabel 2 : Perbandingan performa kuantitatif: Akurasi (mAP) dan kecepatan inferensi (FPS)

#### A. Analisis Akurasi dan Efisiensi pada Kondisi Ideal

Pada kondisi citra bersih, hasil eksperimen menunjukkan fenomena menarik yang membantah asumsi tradisional mengenai trade-off kecepatan dan akurasi. YOLOv8s berhasil mencapai akurasi tertinggi (mAP 0.2870) sekaligus menjadi model yang paling efisien dengan kecepatan inferensi mencapai 476.19 FPS. Kecepatan ini jauh melampaui Faster R-CNN (16.92 FPS) dan SSD (57.02 FPS).

Superioritas YOLOv8s ini dapat ditarik pada desain arsitekturnya yang efisien. Penggunaan modul C2f pada backbone memungkinkan ekstraksi fitur yang lebih kaya dengan parameter yang lebih sedikit dibandingkan VGG-16 yang digunakan SSD atau ResNet pada Faster R-CNN. Selain itu, mekanisme anchor-free menghilangkan beban komputasi untuk memproses ribuan kotak anchor yang tumpang tindih, sehingga mempercepat proses inferensi secara signifikan.



Gambar 3 : Grafik perbandingan trade-off antara akurasi dan kecepatan pada kondisi ideal.

#### B. Evaluasi Robustness terhadap Degradasi Visual

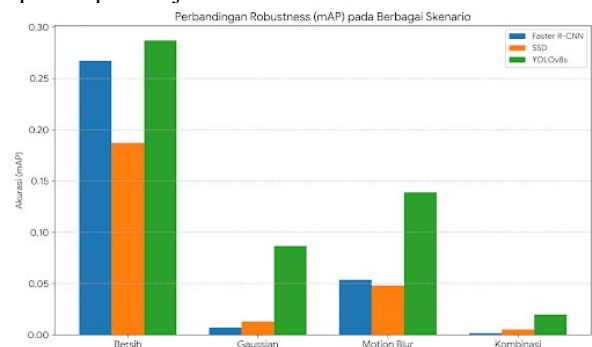
Ketika dihadapkan pada gangguan visual, perbedaan performa antar arsitektur menjadi sangat kontras. Tabel 3 menyajikan persentase penurunan performa (mAP) relatif terhadap kondisi bersih.

Model	Penurunan vs. Gaussian	Penurunan vs. Motion Blur	Penurunan vs. Kombinasi
Faster R-CNN	97.3%	79.9%	99.4%
SSD	93.1%	74.3%	97.1%
YOLOv8s	69.8%	51.4%	93.1%

Tabel 3 : Evaluasi ketahanan (*robustness*): Persentase penurunan mAP pada skenario gangguan visual

Faster R-CNN terbukti menjadi model yang paling rentan, dengan penurunan akurasi mencapai 97.3% pada kondisi Gaussian noise. Hal ini mengindikasikan bahwa Region Proposal Network (RPN), yang bergantung pada deteksi fitur tepi dan tekstur halus untuk mengajukan kandidat objek, sangat sensitif terhadap gangguan level piksel. Ketika noise mengaburkan detail halus tersebut, RPN gagal menghasilkan proposal wilayah yang valid, sehingga proses deteksi terhenti total.

Sebaliknya, YOLOv8s menunjukkan ketahanan (*robustness*) terbaik. Pada kondisi motion blur, akurasinya "hanya" turun sebesar 51.4%, jauh lebih baik dibandingkan model lain. Ketahanan ini kemungkinan besar disebabkan oleh pendekatan anchor-free dan fungsi loss yang lebih modern (seperti CIoU atau DFL), yang membuat model lebih fokus pada pusat massa objek dan fitur global, sehingga tidak terlalu terdistraksi oleh gangguan lokal atau blur pada tepian objek.

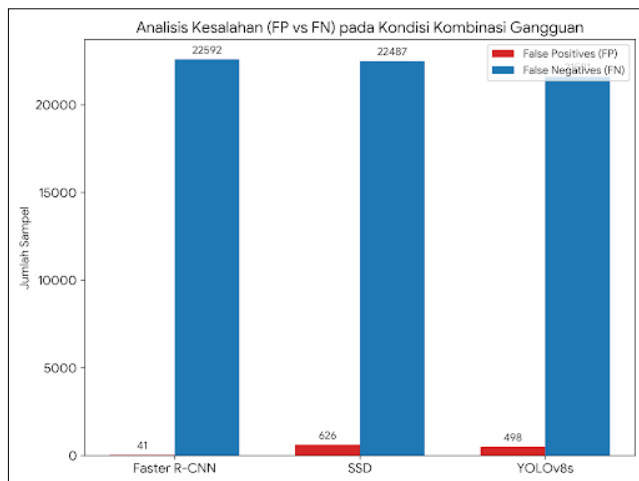


Gambar 4. Grafik perbandingan penurunan akurasi (mAP) pada berbagai skenario gangguan.

### C. Analisis Pola Kesalahan (Error Analysis)

Untuk memahami penyebab penurunan mAP secara lebih mendalam, dilakukan analisis terhadap komponen kesalahan deteksi, yaitu False Positives (FP) dan False Negatives (FN).

Hasil analisis menunjukkan bahwa penyebab utama kegagalan pada Faster R-CNN dan SSD adalah lonjakan drastis pada False Negatives (FN). Pada skenario kombinasi, jumlah FN pada Faster R-CNN meningkat dari 16.791 (bersih) menjadi 22.592. Artinya, model menjadi "buta" dan gagal mengenali keberadaan objek sama sekali. Di sisi lain, SSD menunjukkan kecenderungan untuk menghasilkan False Positives (FP) lebih tinggi pada kondisi motion blur, yang berarti model sering "berhalusinasi" mendeteksi objek yang sebenarnya tidak ada. YOLOv8s mampu mempertahankan keseimbangan terbaik dengan jumlah True Positives (TP) tertinggi dan FP terendah di seluruh skenario, menunjukkan bahwa model ini lebih konservatif dan akurat dalam pengambilan keputusan di lingkungan ber-noise.



Gambar 5. Analisis komponen kesalahan (FP vs FN) pada skenario gangguan kombinasi.

## IV. KESIMPULAN

Penelitian ini telah mengevaluasi performa tiga arsitektur CNN pada kondisi degradasi visual. Berdasarkan hasil eksperimen, dapat disimpulkan bahwa arsitektur modern YOLOv8s menawarkan solusi paling optimal untuk deteksi objek di lingkungan yang menantang. Model ini tidak hanya unggul dalam kecepatan (476.19 FPS) dan akurasi (mAP 0.2870) pada kondisi ideal, tetapi juga menunjukkan tingkat ketahanan (robustness) tertinggi terhadap Gaussian noise dan motion blur. Sebaliknya, arsitektur dua tahap klasik seperti Faster R-CNN terbukti sangat rentan terhadap gangguan visual, menjadikannya kurang ideal untuk aplikasi real-time di lingkungan terbuka.

Implikasi praktis dari temuan ini menyarankan penggunaan YOLOv8s untuk sistem pengawasan industri atau kendaraan otonom yang membutuhkan respons cepat dan toleransi tinggi terhadap gangguan sensor atau lingkungan. Keterbatasan penelitian ini terletak pada penggunaan jenis gangguan sintesis dan fokus pada satu dataset (COCO). Penelitian selanjutnya disarankan untuk

memperluas evaluasi pada kondisi cuaca nyata (hujan, kabut) dan mengeksplorasi teknik augmentasi data untuk meningkatkan ketahanan model lebih lanjut.

## DAFTAR PUSTAKA

- [1] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object Detection with Deep Learning: A Review," *IEEE Trans Neural Netw Learn Syst*, 2019.
- [2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *NeurIPS*, 2015.
- [3] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single Shot MultiBox Detector," *ECCV*, 2016.
- [4] G. Jocher, A. Chaurasia, and J. Qiu., "YOLOv8: State-of-the-art Real-Time Object Detection," *Ultralytics Open Source*, 2023.
- [5] I. Ramadhan and P. Harsadi, "Optimalisasi Sistem Parkir Menggunakan YOLO: Penerapan pada Lingkungan Urban Padat," *JURNAL INFORMATIKA UPGRIS*, vol. 10, no. 2, 2024.
- [6] E. E. Rusak, L. Schott, R. S. Zimmermann, J. Bitterwolf, O. Bringmann, M. Bethge, and W. Brendel, "Benchmarking Robustness in Object Detection: Autonomous Driving when Winter is Coming," 2019, doi: 10.48550/arXiv.1907.07484.
- [7] Budi Cahyo Wibowo, Fajar Nugraha, and Andy Prasetyo Utomo, "Uji Deteksi Objek Bentuk Bola Dengan Menerapkan Metode Circular Hough Transform," Jun. 2021.
- [8] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick., "Microsoft COCO: Common Objects in Context," *ECCV*, 2014.