Jurnal KomtekInfo
https://jkomtekinfo.org/ojs
2024 Vol. 11 No. 4 Hal: 457-463 p-ISSN: 2356-0010, e-ISSN: 2502-8758

Analisis Trade-Off Presisi Recall Modifikasi YOLOv8s Lapisan P2, CBAM, Wise-IoU v3 Deteksi Objek Mikro UAV

Agung Ramadhanu[1]✉, Maheza Novrayuda [2]

[1]Magister Teknik Informatika, Fakultas Ilmu Komputer, Universitas Putra Indonesia YPTK, Jl. Raya Lubuk Begalung, 25221, Padang, Indonesia

[2]Teknik Informatika, Fakultas Ilmu Komputer, Universitas Putra Indonesia YPTK, Jl. Raya Lubuk Begalung, 25221, Padang, Indonesia

✉ agung_ramadhanu@upiyptk.ac.id

Abstract

Detecting micro-sized objects in unmanned aerial vehicle (UAV) images is difficult because each object covers only a few pixels. This study examines whether three modifications of YOLOv8s increase detection precision: an extra P2 detection head (stride 4), the Convolutional Block Attention Module (CBAM), and the Wise-IoU v3 loss. In the VisDrone training labels used here, 84.80% of 22,568 annotated objects are smaller than 32 × 32 pixels. Four models (M1–M4) were trained in a controlled ablation for 20 epochs with batch size 4 on an NVIDIA GeForce RTX 2050 (4 GB) and evaluated on 98 test images. The full model (M4) raised precision from 37.90% to 48.68% (+10.78 points), but recall fell from 27.74% to 19.26%, mAP@50 from 25.86% to 15.96%, and mAP_{small} from 21.98% to 14.16%. A paired bootstrap (1,000 resamples) gave a per-image F1 difference of −5.56 points (95% CI −6.89 to −4.20). On one dense test image, false positives fell from 51 to 13. The training curves show that the P2 models started from a much higher loss and were still improving at epoch 20, so unequal weight initialization is a plausible but untested explanation. M4 ran at 37.1 FPS (26.99 ms) on the RTX 2050. The findings are exploratory and describe a high-precision, low-recall trade-off rather than an overall accuracy gain.

Keywords: Small object detection, UAV, YOLOv8, CBAM, Wise-IoU

Abstrak

Mendeteksi objek berukuran mikro pada citra *unmanned aerial vehicle* (UAV) sulit karena satu objek hanya menempati beberapa piksel. Penelitian ini menguji apakah tiga modifikasi pada YOLOv8s meningkatkan presisi deteksi: tambahan *head* deteksi P2 (*stride* 4), *Convolutional Block Attention Module* (CBAM), dan fungsi kerugian Wise-IoU v3. Pada label data latih VisDrone yang dipakai, 84,80% dari 22.568 objek berukuran kurang dari 32 × 32 piksel. Empat model (M1–M4) dilatih dalam studi ablasi terkontrol selama 20 *epoch* dengan ukuran *batch* 4 pada GPU NVIDIA GeForce RTX 2050 (4 GB), lalu dievaluasi pada 98 citra uji. Model lengkap (M4) menaikkan presisi dari 37,90% menjadi 48,68% (+10,78 poin), tetapi *recall* turun dari. 27,74% menjadi 19,26%, mAP@50 dari 25,86% menjadi 15,96%, dan mAP_{small} dari 21,98% menjadi 14,16%. Uji *paired bootstrap* (1.000 pengambilan ulang) memberi selisih F1 per citra −5,56 poin (selang kepercayaan 95%: −6,89 sampai −4,20). Pada satu citra uji yang padat, *false positive* turun dari 51 menjadi 13. Kurva pelatihan menunjukkan bahwa model P2 mulai dari kerugian yang jauh lebih tinggi dan masih membaik pada *epoch* ke-20, sehingga inisialisasi bobot yang tidak setara menjadi dugaan yang masuk akal tetapi belum diuji. M4 berjalan pada 37,1 FPS (26,99 ms) di RTX 2050. Temuan ini bersifat eksploratori dan menggambarkan *trade-off* presisi tinggi dan *recall* rendah, bukan peningkatan akurasi secara keseluruhan

Kata kunci: Deteksi objek kecil, UAV, YOLOv8, CBAM, Wise-IoU

KomtekInfo is licensed under a Creative Commons Attribution-Share Alike 4.0 International License.

1. Pendahuluan

Kendaraan udara tanpa awak (*unmanned aerial vehicle*, UAV) kini banyak dipakai untuk memantau lalu lintas, mengawasi lahan, dan menilai kerusakan akibat bencana. Dari ketinggian, kendaraan dan pejalan kaki hanya tampak sebagai bercak beberapa piksel. Karena itu, mendeteksi objek pada citra UAV jauh lebih sulit daripada pada citra yang diambil dari permukaan tanah [1]. Pada data latih yang dipakai dalam penelitian ini, 19.138 dari 22.568 objek (84,80%) berukuran kurang dari 32 × 32 piksel, yaitu batas objek kecil pada kriteria MS-COCO [2]. Dalam tulisan ini kelompok tersebut disebut objek mikro.

Keluarga YOLO (*You Only Look Once*) menjadi pilihan utama untuk deteksi objek waktu nyata karena memprediksi kotak dan kelas dalam satu lintasan jaringan [3], [4]. YOLOv8 [5] memakai tiga *detection head* pada tingkat fitur P3, P4, dan P5 dengan *stride* 8, 16, dan 32. Pada *stride* 8, objek berukuran 16 piksel hanya menempati sekitar 2 × 2 sel pada P3, dan objek 8 piksel hanya satu sel, sehingga informasi rincinya mudah hilang. Hal ini mendorong banyak peneliti menambah tingkat fitur yang lebih dangkal, memperkuat fusi fitur, dan mengganti fungsi kerugian regresi kotak [6]–[10]. Cara lain yang tidak mengubah arsitektur adalah memotong citra menjadi irisan pada saat inferensi [11].

Tiga komponen sering dipakai bersama. Pertama, *head* P2 (*stride* 4, peta fitur 160 × 160 pada masukan 640 × 640) menambah resolusi spasial sesuai prinsip piramida fitur [12]. Kedua, *Convolutional Block Attention Module* (CBAM) memberi bobot pada kanal dan lokasi sehingga latar belakang yang ramai dapat ditekan [13]. Ketiga, Wise-IoU [14] mengatur bobot gradien secara dinamis agar kotak berkualitas rendah tidak terlalu mendominasi pelatihan; fungsi ini menggantikan CIoU dan DIoU yang berbasis jarak dan rasio sisi [15], [16]. Studi yang menggabungkan komponen semacam ini umumnya melaporkan kenaikan mAP, tetapi pengaruh tiap komponen terhadap presisi dan *recall* tidak selalu diurai secara terpisah.

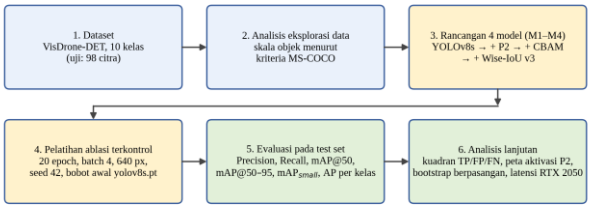
Diterima: 22-10-2024 | Revisi: 24-11-2024 | Diterbitkan: 31-12-2024 | doi: 10.35134/komtekinfo.v11i4.722

Di dalam negeri, kajian YOLOv8 yang kami temukan umumnya berupa perbandingan versi atau penerapan pada satu jenis objek, misalnya deteksi manusia [17], [18] dan deteksi UAV militer [19], di samping penerapan YOLO pada kasus lain seperti pembacaan pelat nomor [20]. Modifikasi *detection head* untuk objek kecil juga mulai dikaji pada tingkat prosiding [21].

Penelitian ini menguji tiga pertanyaan. (RQ1) Apakah penambahan *head* P2, CBAM, dan Wise-IoU v3 pada YOLOv8s meningkatkan presisi deteksi pada citra UAV? (RQ2) Apa dampaknya terhadap *recall*, mAP, dan mAP_{small}? (RQ3) Berapa tambahan beban komputasinya, dan apakah model tetap berjalan waktu nyata pada GPU RTX 2050? Kontribusinya ada empat: (i) analisis skala objek pada data VisDrone; (ii) studi ablasi empat model dengan hiperparameter yang sama; (iii) analisis kuadran TP/FP/FN, peta aktivasi fitur, dan uji *bootstrap* berpasangan; serta (iv) pelaporan hasil apa adanya, termasuk penurunan mAP yang tidak sejalan dengan kenaikan presisi

2. Metodologi Penelitian

Penelitian ini berupa eksperimen terkontrol dengan alur pada Gambar 1. Seluruh angka dalam artikel bersumber dari keluaran *notebook* eksperimen dan berkas CSV tabel yang dihasilkannya.



Gambar 1. Alur penelitian: *dataset*, analisis skala objek, rancangan empat model, pelatihan ablasi, evaluasi pada test set, dan analisis lanjutan.

2.1 Lingkungan Eksperimen

Eksperimen berjalan pada satu *notebook* dengan Python 3.13.2, PyTorch 2.13.0 (CUDA 12.6), dan Ultralytics 8.3.40. Perangkat kerasnya adalah GPU NVIDIA GeForce RTX 2050 dengan VRAM 4 GB. Benih acak dikunci pada nilai 42. Pelatihan memakai 20 *epoch*, ukuran *batch* 4, resolusi masukan 640 × 640 piksel, *early stopping* dengan *patience* 10, dan presisi campuran FP16 (*automatic mixed precision*). Jumlah *worker* diatur 0 agar aman dari masalah *multiprocessing* pada Windows. Semua model diinisialisasi dari berkas bobot pralatih yolov8s.pt.

2.2 Dataset dan Analisis Skala Objek

Data yang dipakai adalah VisDrone-DET [1] dengan 10 kelas: *pedestrian*, *people*, *bicycle*, *car*, *van*, *truck*, *tricycle*, *awning-tricycle*, *bus*, dan *motor*. Evaluasi akhir memakai *test set* berisi 98 citra. Jumlah citra latih dan validasi tidak tercatat pada keluaran *notebook* sehingga tidak dapat dilaporkan di sini (lihat Subbab 3.10). Skala objek dihitung dari berkas label data latih pada skala masukan 640 piksel dengan batas MS-COCO [2]: objek kecil berluas kurang dari 32 × 32 = 1.024 piksel², objek sedang 1.024 sampai kurang dari 96 × 96 = 9.216 piksel², dan objek besar selebihnya.

2.3 Rancangan Arsitektur

Empat model dibandingkan (Tabel 1). M1 adalah YOLOv8s baku. M2 menambah *head* P2 sehingga ada empat *head* (P2–P5), dengan jalur fusi ke P2 pada PANet. M3 menyematkan CBAM pada fitur P2 dan P3 sebelum masuk ke *head*. M4 sama dengan M3, tetapi fungsi kerugian regresi kotaknya Wise-IoU v3. Skema model ditunjukkan pada Gambar 2.

Tabel 1. Konfigurasi dan kompleksitas model pada studi ablasi

Model	Konfigurasi (kerugian)	Head	Param. (M)	GFLOPs
M1	YOLOv8s baku (CIoU + DFL)	3 (P3–P5)	11,13	28,6
M2	+ P2 (CIoU + DFL)	4 (P2–P5)	10,63	36,7
M3	+ P2 + CBAM (CIoU + DFL)	4 (P2–P5)	10,64†	37,0
M4	+ P2 + CBAM (WIoU v3 + DFL)	4 (P2–P5)	10,64†	37,0

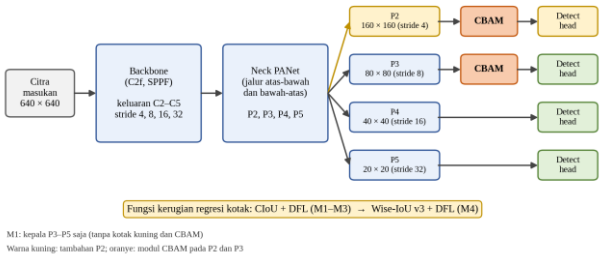
†Jumlah parameter saat arsitektur dibangun. Pada tahap *benchmark* latensi (Tabel 6), *notebook* mencatat 10,63 M untuk M2, M3, dan M4. Selisih 0,01 M tidak dianalisis lebih lanjut.

CBAM terdiri atas dua modul berurutan [13]. Modul atensi kanal membangkitkan bobot kanal dari hasil *average pooling* dan *max pooling* spasial yang dilewatkan pada jaringan *multilayer perceptron* (MLP) bersama (Persamaan 1). Modul atensi spasial menggabungkan hasil *pooling* sepanjang kanal, lalu memprosesnya dengan konvolusi berkernel 7 × 7 (Persamaan 2). Fitur masukan F dihaluskan secara berurutan oleh kedua modul (Persamaan 3). Rasio reduksi kanal MLP yang dipakai adalah 16.

$$M_c(F) = \sigma(\mathrm{MLP}(\mathrm{AvgPool}(F)) + \mathrm{MLP}(\mathrm{MaxPool}(F))) \quad (1)$$

$$M_s(F') = \sigma(f^{7\times7}([\mathrm{AvgPool}(F'); \mathrm{MaxPool}(F')])) \quad (2)$$

$$F' = M_c(F) \otimes F; \quad F'' = M_s(F') \otimes F' \quad (3)$$



Gambar 2. Skema konseptual arsitektur: tambahan *head* P2 (kuning), CBAM pada P2 dan P3 (oranye), serta penggantian fungsi kerugian pada M4. Skema ini tidak menyalin indeks lapisan pada berkas konfigurasi.

2.4 Fungsi Kerugian Wise-IoU v3

Wise-IoU v1 mengalikan kerugian IoU dengan faktor jarak pusat kotak [14] (Persamaan 4 dan 5). Versi 3 menambahkan faktor fokus nonmonotonik yang bergantung pada derajat anomali β (Persamaan 6 sampai 8).

$$L_{\text{WIoUv1}} = R_{\text{WIoU}} \cdot L_{\text{IoU}} \quad (4)$$

$$R_{\text{WIoU}} = \exp\left(\frac{(x - x_{\text{gt}})^2 + (y - y_{\text{gt}})^2}{(W_g^2 + H_g^2)^*}\right) \quad (5)$$

$$\beta = \frac{L_{\text{IoU}}^*}{\bar{L}_{\text{IoU}}} \quad (6)$$

$$r = \frac{\beta}{\delta \alpha^{\beta - \delta}} \quad (7)$$

$$L_{\text{WIoUv3}} = r \cdot L_{\text{WIoUv1}} \quad (8)$$

Di sini (x, y) adalah pusat kotak prediksi, (x_{gt}, y_{gt}) pusat kotak acuan, W_g dan H_g lebar dan tinggi kotak penutup terkecil, tanda $*$ menunjukkan nilai yang dilepas dari graf gradien, L_{IoU}^* kerugian IoU saat ini, dan $\bar{L}_{IoU}$ rata-rata kerugian IoU. Hiperparameter yang dipakai $\alpha = 1{,}9$ dan $\delta = 3{,}0$. Pada M4, kerugian distribusi (*distribution focal loss*, DFL) [22] tetap dipertahankan. Uji numerik awal pada kotak sintetis menghasilkan nilai kerugian 0,8142 dan gradien terhadap kotak prediksi 0,1258, sehingga implementasinya berjalan stabil.

2.5 Protokol Ablasi

Keempat model dilatih dengan hiperparameter yang sama (Subbab 2.1) dan satu benih acak. Bobot terbaik (*best.pt*) menurut metrik validasi Ultralytics dipakai untuk evaluasi. Pada model dengan P2 dan CBAM, sebagian lapisan tidak punya padanan pada berkas yolov8s.pt. Jumlah bobot yang berhasil dimuat tidak dicatat, sehingga kami hanya dapat menduga bahwa M2–M4 memulai pelatihan dengan bobot yang kurang matang dibandingkan M1 (dibahas pada Subbab 3.2 dan 3.9).

2.6 Metrik Evaluasi

Presisi (P), *recall* (R), dan F1 didefinisikan dari jumlah *true positive* (TP), *false positive* (FP), dan *false negative* (FN):

$$P = \frac{\text{TP}}{\text{TP+FP}}; \quad R = \frac{\text{TP}}{\text{TP+FN}} \quad (9)$$

$$F1 = \frac{2\text{PR}}{P + R} \quad (10)$$

$$\text{mAP} = \frac{1}{C}\sum_{c=1}^{C} \text{AP}_c \quad (11)$$

dengan C = 10 kelas. Presisi dan *recall* pada Tabel 2 adalah rata-rata antarkelas pada ambang kepercayaan yang dipilih Ultralytics saat validasi. F1 pada Tabel 2 dihitung penulis dari P dan R rata-rata tersebut, bukan keluaran langsung *notebook*. mAP@50 memakai ambang IoU 0,50, sedangkan mAP@50–95 merata-ratakan ambang 0,50 sampai 0,95 berlangkah 0,05. $\text{mAP}_{\text{small}}$ dihitung hanya pada objek berluas kurang dari 1.024 piksel² sesuai kriteria MS-COCO.

2.7 Analisis Tambahan

Kuadran deteksi. Pada satu citra uji yang padat, deteksi dengan ambang kepercayaan 0,15 dikelompokkan menjadi TP, FP, dan FN. Ambang IoU untuk pencocokan tidak tercatat pada keluaran *notebook*.

Peta aktivasi. Aktivasi lapisan P2 dirata-ratakan lintas kanal lalu dinormalisasi min–maks bersama untuk M2 (lapisan 18) dan M3 (lapisan 19) pada satu citra uji. *Notebook* mencantumkan *Foreground Activation Ratio* (FAR) pada judul gambar, tetapi rumusnya tidak tercantum pada sel *notebook*.

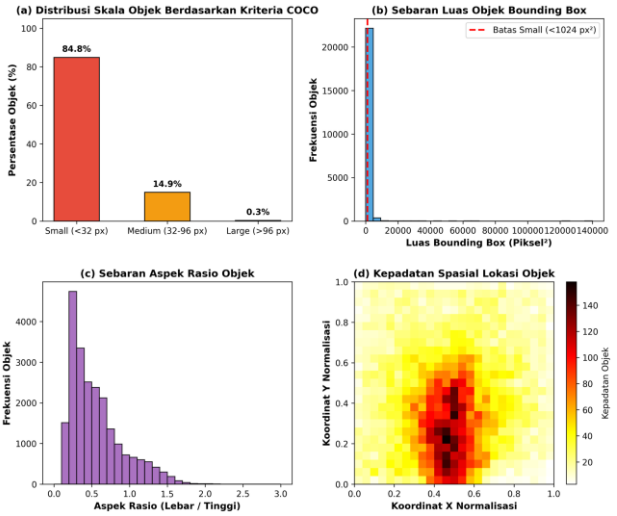
Uji statistik. Skor F1 per citra pada ambang kepercayaan 0,15 dihitung untuk M1 dan M4. Selisih M4 dikurangi M1 diuji dengan *paired bootstrap* [23] sebanyak 1.000 pengambilan ulang citra uji dengan pengembalian, disertai selang kepercayaan persentil 95%. Hipotesis nolnya adalah selisih rata-rata ≤ 0, melawan hipotesis alternatif satu sisi bahwa selisih > 0.

Latensi. Pengukuran memakai *batch* 1 dengan 10 iterasi pemanasan dan 50 iterasi ukur. Dilaporkan waktu prapemrosesan, inferensi tensor, *non-maximum suppression* (NMS), total *end-to-end* (median p50 dan persentil ke-95), *throughput* dalam FPS (1.000 dibagi p50), dan puncak VRAM. Batas waktu nyata ditetapkan 30 FPS (33,33 ms).

3. Hasil dan Pembahasan

3.1 Karakteristik Objek pada Data

Gambar 3 meringkas karakteristik data. Dari 22.568 objek pada label data latih, 19.138 (84,80%) berukuran kecil, 3.342 (14,81%) sedang, dan hanya 88 (0,39%) besar. Hampir seluruh objek berada pada kisaran ukuran yang sulit bagi YOLOv8 baku sehingga rancangan untuk objek mikro memang beralasan. Rasio lebar terhadap tinggi sebagian besar kurang dari 1 (puncak sekitar 0,25–0,30), yaitu kotak yang lebih tinggi daripada lebarnya, sesuai bentuk pejalan kaki. Kepadatan objek terpusat pada lajur tengah citra.

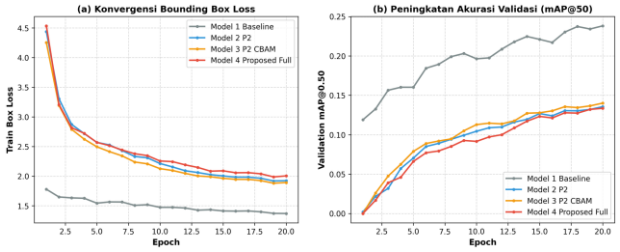


Gambar 3. Karakteristik objek pada label data latih: (a) distribusi skala menurut MS-COCO, (b) sebaran luas kotak, (c) sebaran rasio lebar terhadap tinggi, dan (d) kepadatan lokasi objek. Panel (a) membulatkan nilai sedang dan besar menjadi 14,9% dan 0,3%; teks memakai 14,81% dan 0,39% dari hitungan langsung.

3.2 Konvergensi Pelatihan

Gambar 4 memperlihatkan kurva pelatihan selama 20 *epoch*. Pada *epoch* pertama, *box loss* pelatihan M1 sekitar 1,8, sedangkan M2–M4 mulai dari sekitar 4,3–4,5. mAP@50 validasi M1 sudah sekitar 0,12 pada *epoch* pertama, sementara M2–M4 hampir nol. Pada *epoch* ke-20, *box loss* M1 sekitar 1,4 dan M2–M4 sekitar 1,9–2,0, sedangkan mAP@50 validasi M1 sekitar 0,24 dan M2–M4 sekitar 0,13–0,14. Kurva M2–M4 masih naik pada

epoch terakhir, sehingga belum dapat dikatakan konvergen. Nilai *box loss* M4 juga tidak dapat dibandingkan langsung dengan M2 dan M3 karena M4 memakai Wise-IoU yang skalanya berbeda dari CIoU. Pola ini konsisten dengan dugaan bahwa M2–M4 memulai dari bobot yang kurang matang, tetapi kurva saja belum membuktikannya.



Gambar 4. Konvergensi pelatihan empat model selama 20 *epoch*: (a) *box loss* pelatihan dan (b) mAP@50 pada data validasi. M4 memakai Wise-IoU sehingga skala kerugiannya tidak setara dengan M1–M3.

3.3 Kinerja Kuantitatif Studi Ablasi

Tabel 2 memuat hasil pada 98 citra uji. Dari M1 ke M4, presisi naik 10,78 poin (37,90% menjadi 48,68%), tetapi *recall* turun 8,48 poin, mAP@50 turun 9,90 poin, mAP@50–95 turun 7,09 poin, dan mAP_{small} turun 7,82 poin. Jadi, untuk RQ1 kenaikan presisi memang terbukti pada angka rata-rata antarkelas, dan untuk RQ2 harganya adalah turunnya semua metrik yang mengukur kelengkapan deteksi.

Tabel 2. Kinerja pada test set VisDrone (98 citra), dalam persen

Model	P	R	F1‡	mAP@50	mAP@50–95	mAP_{small}
M1	37,90	27,74	32,03	25,86	14,81	21,98
M2	48,48	18,86	27,16	16,24	8,15	13,75
M3	39,75	20,14	26,73	17,05	8,61	14,37
M4	48,68	19,26	27,60	15,96	7,72	14,16

M1 = YOLOv8s baku; M2 = + P2; M3 = + P2 + CBAM; M4 = + P2 + CBAM + Wise-IoU v3. ‡F1 dihitung dari P dan R pada tabel ini, bukan keluaran langsung *notebook*.

Menurut tahap penambahan, P2 saja (M1 ke M2) menaikkan presisi 10,58 poin tetapi menurunkan *recall* 8,88 poin dan mAP@50 9,62 poin; penurunan terbesar dalam seluruh ablasi terjadi pada tahap ini. CBAM (M2 ke M3) menurunkan presisi 8,73 poin, menaikkan *recall* 1,28 poin, mAP@50 0,81 poin, dan mAP_{small} 0,62 poin. Wise-IoU v3 (M3 ke M4) menaikkan presisi 8,93 poin, tetapi menurunkan *recall* 0,88 poin, mAP@50 1,09 poin, dan mAP_{small} 0,21 poin. Karena tiap model hanya dilatih sekali, selisih sekitar satu poin antara M2, M3, dan M4 belum dapat dibedakan dari variasi antarpelatihan. Selisih terhadap M1 jauh lebih besar dan searah pada semua metrik, sehingga lebih layak dibahas.

3.4 Kinerja per Kelas

Tabel 3 menunjukkan AP@50 pada lima kelas yang tersedia pada berkas keluaran. Pada kelima kelas itu M4 lebih rendah daripada M1: *car* turun dari 74,91% menjadi 66,20%, *motor* dari 33,08% menjadi 25,80%, *people* dari 31,51% menjadi 23,80%, dan *pedestrian* dari 27,36% menjadi 24,50%. Kelas *bicycle* adalah yang terendah di antara kelima kelas pada semua model (9,51% pada M1 dan 3,49% pada M4).

Tabel 3. AP@50 per kelas pada test set (%)

Model	Pedestrian	People	Bicycle	Motor	Car	mAP@50 (10 kelas)
M1	27,36	31,51	9,51	33,08	74,91	25,86
M2	22,99	27,60	2,14	29,64	71,04	16,24
M3	23,76	27,36	2,52	29,04	69,42	17,05
M4	24,50	23,80	3,49	25,80	66,20	15,96

Kolom terakhir adalah mAP@50 rata-rata 10 kelas. Karena mAP@50 merupakan rata-rata sepuluh AP per kelas, rata-rata lima kelas lainnya (*van*, *truck*, *tricycle*, *awning-tricycle*, dan *bus*) dapat dihitung mundur dari Tabel 3 (Tabel 4). Pada M1 rata-rata kelima kelas itu sekitar 16,4%, sedangkan pada M2–M4 hanya 1,8–3,7%. Kelima kelas ini menyumbang sekitar dua pertiga (67%) dari penurunan mAP@50 M1 ke M4. Dengan kata lain, turunnya mAP terutama terjadi pada kelas yang tidak tampil di Tabel 3. AP per kelas untuk kelima kelas tersebut tidak tersimpan pada berkas keluaran *notebook*, sehingga angka ini hanya perkiraan yang mengandung galat pembulatan. Kelas-kelas itu umumnya lebih jarang dan lebih besar dibanding *pedestrian* atau *people*, mungkin sebab mereka lebih peka terhadap bobot awal, tetapi hal ini belum diuji.

Tabel 4. Rerata AP@50 menurut kelompok kelas (%)

Model	Lima kelas pada Tabel 3	Lima kelas lain§	mAP@50 (10 kelas)
M1	35,3	16,4	25,86
M2	30,7	1,8	16,24
M3	30,4	3,7	17,05
M4	28,8	3,2	15,96

§Dihitung mundur: (10 × mAP@50 − jumlah AP lima kelas pada Tabel 3) ÷ 5.

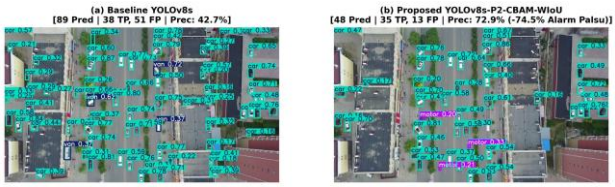
3.5 Analisis Kuadran Deteksi

Tabel 5 dan Gambar 5 menunjukkan satu citra uji yang padat (ambang kepercayaan 0,15). M1 menghasilkan 89 prediksi: 38 TP dan 51 FP, dengan 29 objek terlewat (FN). M4 menghasilkan 48 prediksi: 35 TP dan 13 FP, dengan 32 FN. Presisi pada citra ini naik dari 42,7% menjadi 72,9%. FP berkurang 38 kotak (74,5%), sedangkan TP hilang 3 kotak. Rasio TP terhadap FP naik dari 0,75 menjadi 2,69. Jumlah objek acuan pada citra ini 67 (TP + FN) untuk kedua model.

Tabel 5. Kuadran deteksi pada satu citra uji yang padat

Model	Prediksi	TP	FP	FN	Presisi (%)	TP/FP
M1	89	38	51	29	42,7	0,75
M4	48	35	13	32	72,9	2,69

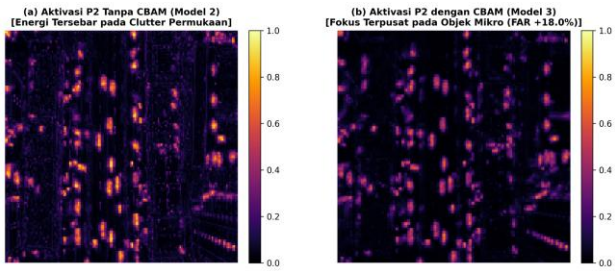
Hasil ini hanya berlaku untuk satu citra. Ia menggambarkan mekanisme turunnya alarm palsu, bukan rata-rata seluruh *test set*. Pada tingkat seluruh set, turunnya *recall* sebesar 8,48 poin menunjukkan bahwa M4 juga kehilangan lebih banyak objek; pada citra ini pun FN naik dari 29 menjadi 32.



Gambar 5. Hasil deteksi pada satu citra uji (ambang kepercayaan 0,15): (a) M1 dengan 89 prediksi (38 TP, 51 FP) dan (b) M4 dengan 48 prediksi (35 TP, 13 FP).

3.6 Peta Aktivasi Fitur P2

Gambar 6 membandingkan aktivasi lapisan P2 tanpa CBAM (M2) dan dengan CBAM (M3) pada satu citra uji. Pada M2, energi aktivasi menyebar ke tekstur permukaan. Pada M3, aktivasi tampak lebih terkonsentrasi di sekitar kendaraan dan area latar tampak lebih gelap. *Notebook* mencatat kenaikan FAR sebesar 17,95% pada gambar ini. Rumus FAR dan cara menandai area latar depan tidak tercantum pada sel *notebook*, sehingga angka itu diperlakukan sebagai indikasi, bukan bukti. Pengamatan ini juga berasal dari satu citra dan dari rata-rata lintas kanal. Perbaikan visual tersebut tidak otomatis berarti deteksi membaik: dari M2 ke M3, presisi turun 8,73 poin dan mAP@50 hanya naik 0,81 poin (Tabel 2).



Gambar 6. Aktivasi lapisan P2 pada satu citra uji: (a) M2 tanpa CBAM dan (b) M3 dengan CBAM. *Notebook* mencatat FAR +17,95%; judul panel (b) pada gambar membulatkannya menjadi 18,0%.

3.7 Uji Statistik Berpasangan

Uji *paired bootstrap* pada skor F1 per citra M4 terhadap M1 memberi selisih −5,56 poin (selang kepercayaan 95%: −6,89 sampai −4,20) dan p = 1,0000 untuk hipotesis satu sisi bahwa M4 lebih baik. Artinya, tidak ada pengambilan ulang yang menunjukkan M4 unggul, dan selang kepercayaannya tidak memuat nol, sehingga pada ukuran ini M1 yang lebih baik. Hasil tersebut sejalan dengan turunnya *recall* dan mAP. F1 per citra memakai ambang kepercayaan 0,15 yang sama untuk kedua model, sedangkan ambang terbaik tiap model dapat berbeda; hasil ini perlu dikonfirmasi dengan ambang lain dan pada lebih dari 98 citra.

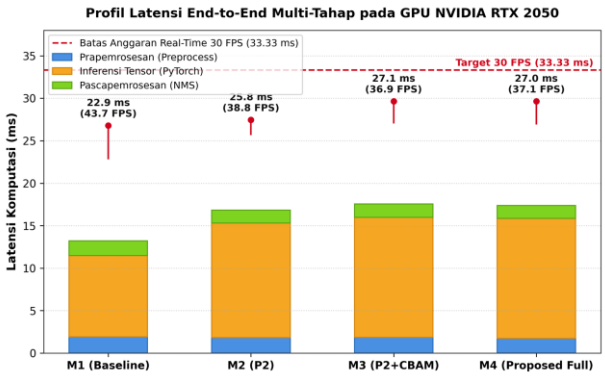
3.8 Efisiensi Komputasi

Tabel 6 memuat hasil latensi. M4 membutuhkan 26,99 ms (p50) per citra atau 37,1 FPS, dibandingkan 22,91 ms (43,7 FPS) untuk M1. Waktu inferensi tensor naik 47% (9,57 ms menjadi 14,11 ms), total latensi naik 17,8%, puncak VRAM naik 8,2% (248,6 MB menjadi 269,0 MB), dan GFLOPs naik 29% (28,6 menjadi 37,0). Semua model melampaui 30 FPS, dengan margin M4 sebesar 23,7%. Latensi p95 M4 adalah 29,64 ms, masih di bawah 33,33 ms. Jumlah parameter justru turun dari 11,13 M ke 10,63 M, kemungkinan karena penyesuaian *head* deteksi pada arsitektur empat *head*, sedangkan GFLOPs naik karena komputasi pada peta fitur beresolusi tinggi; penjelasan ini belum diuji. Jadi, untuk RQ3, tambahan beban komputasi tergolong sedang pada GPU ini.

Tabel 6. Latensi per tahap dan efisiensi pada GPU RTX 2050 (*batch* 1)

Model	Pra (ms)	Infer. (ms)	NMS (ms)	Total p50 (ms)	FPS	VRAM (MB)
M1	1,90	9,57	1,74	22,91	43,7	248,6
M2	1,81	13,50	1,51	25,76	38,8	268,4
M3	1,84	14,13	1,59	27,11	36,9	269,0
M4	1,74	14,11	1,53	26,99	37,1	269,0

Jumlah waktu komponen M1 (1,90 + 9,57 + 1,74 = 13,21 ms) lebih kecil daripada total p50 sebesar 22,91 ms. Selisih sekitar 9,7 ms diduga berasal dari beban pemanggilan *pipeline* Python dan Ultralytics, pemindahan memori dari *host* ke perangkat pada Windows dengan *worker* 0, serta sinkronisasi CUDA. Dugaan ini belum diukur secara terpisah. Selisih serupa muncul pada semua model (M2 8,94 ms; M3 9,55 ms; M4 9,61 ms), sehingga tidak mengubah urutan antarmodel.



Gambar 7. Dekomposisi latensi *end-to-end* per tahap pada GPU RTX 2050. Batang menunjukkan jumlah komponen; titik merah menunjukkan total p50 (batang galat hingga p95).

3.9 Pembahasan

RQ1. Presisi rata-rata naik 10,78 poin pada M4, dan pada satu citra padat jumlah FP turun 74,5%. Namun, kenaikan presisi pada ambang tetap tidak dengan sendirinya berarti kemampuan membedakan latar yang lebih baik. Pada ambang 0,15 yang sama, M4 hanya menghasilkan 48 prediksi dibanding 89 pada M1. Hal ini dapat berarti skor kepercayaan M4 umumnya lebih rendah, sehingga ambang efektifnya lebih ketat. Perbandingan yang lebih adil adalah AP, yang meliputi seluruh ambang, dan AP justru turun (mAP@50 dari 25,86% menjadi 15,96%). Maka klaim yang dapat dipertanggungjawabkan adalah M4 bersifat lebih konservatif pada ambang tersebut, bukan lebih akurat.

RQ2. Harga kenaikan presisi adalah turunnya *recall* (−8,48 poin), mAP@50 (−9,90 poin), mAP_{small} (−7,82 poin), dan F1 per citra (−5,56 poin; selang kepercayaan tidak memuat nol). Penurunan ini terlihat pada semua kelas di Tabel 3 dan, menurut perhitungan mundur, paling besar pada lima kelas yang tidak dilaporkan. Berbeda dengan sejumlah studi yang melaporkan kenaikan mAP dari tambahan lapisan deteksi dangkal dan Wise-IoU pada

VisDrone [7]–[9], studi ini tidak menemukannya. Penyebabnya tidak dapat dipastikan karena kondisi pelatihan tiap studi berbeda; kondisi di sini terbatas pada 20 *epoch*, *batch* 4, dan GPU 4 GB.

Dugaan penyebab. Tiga pengamatan mengarah pada ketidaksetaraan inisialisasi bobot. Pertama, M2–M4 mulai dari kerugian sekitar 4,3–4,5 dan mAP@50 validasi hampir nol, sedangkan M1 mulai dari 1,8 dan 0,12. Kedua, lapisan P2 dan CBAM tidak punya padanan pada yolov8s.pt, sehingga hanya M1 yang berangkat dari bobot pralatih utuh. Ketiga, kurva M2–M4 masih naik pada *epoch* ke-20. Dugaan ini baru dapat diuji dengan (a) pelatihan yang lebih lama, (b) inisialisasi yang setara, misalnya memuat semua bobot yang cocok atau melatih M1 dari awal pula, serta (c) replikasi dengan beberapa benih acak dan uji statistik pada *test set* yang lebih besar. Efek tiap komponen (P2, CBAM, Wise-IoU) juga belum dapat dipisahkan dengan yakin sebelum langkah-langkah tersebut dilakukan.

RQ3. M4 tetap berjalan waktu nyata (37,1 FPS; margin 23,7% di atas 30 FPS) dengan tambahan latensi 17,8% dan puncak VRAM 8,2% dibanding M1. Kesimpulan ini terbatas pada RTX 2050, *batch* 1, dan *pipeline* Ultralytics yang dipakai.

3.10 Keterbatasan Penelitian

Penelitian ini memiliki delapan keterbatasan. (1) *Test set* hanya 98 citra, dan jumlah citra latih serta validasi tidak tercatat pada log keluaran sehingga reprodusibilitas data belum lengkap. (2) Setiap model dilatih sekali dengan satu benih acak; tidak ada replikasi sehingga variasi antarpelatihan tidak diketahui. (3) Pelatihan hanya 20 *epoch* dengan *batch* 4, dan model P2 belum konvergen. (4) Inisialisasi bobot M1 dan M2–M4 tidak setara dan jumlah bobot yang dimuat tidak dicatat (hipotesis pada Subbab 3.9). (5) Nilai kerugian M4 tidak sebanding dengan M1–M3 karena fungsi kerugiannya berbeda. (6) Analisis kuadran dan peta aktivasi masing-masing hanya memakai satu citra; ambang IoU pencocokan dan rumus FAR tidak tercatat, dan AP lima kelas lain hanya diperoleh lewat perhitungan mundur. (7) Latensi diukur pada satu GPU dengan *batch* 1, dan selisih sekitar 9,7 ms antara jumlah komponen dan total belum diukur terpisah. (8) Skor F1 per citra memakai satu ambang kepercayaan (0,15). Karena itu, seluruh temuan bersifat eksploratori.

4. Kesimpulan

Model fusi fitur yang mencapai akurasi 99,64% pada Pada YOLOv8s untuk citra UAV VisDrone, penambahan *head* P2, CBAM, dan Wise-IoU v3 menaikkan presisi rata-rata dari 37,90% menjadi 48,68% (+10,78 poin) dan menekan *false positive* pada satu citra padat dari 51 menjadi 13. Namun, *recall* turun dari 27,74% menjadi 19,26%, mAP@50 dari 25,86% menjadi 15,96%, mAP_{small} dari 21,98% menjadi 14,16%, dan selisih F1 per citra −5,56 poin (selang kepercayaan 95%: −6,89 sampai −4,20). Model tetap waktu nyata pada 37,1 FPS dengan RTX 2050. Dengan demikian, model usulan lebih cocok disebut detektor konservatif berpresisi tinggi daripada detektor yang lebih akurat secara keseluruhan.

Kurva pelatihan dan pola pada kelas yang tidak dilaporkan mengarah pada dugaan ketidaksetaraan inisialisasi bobot, tetapi dugaan ini belum diuji. Penelitian lanjutan perlu melatih lebih lama dengan inisialisasi yang setara, mereplikasi dengan beberapa benih acak, melaporkan AP seluruh kelas, dan menguji pada *test set* yang lebih besar.

Daftar Rujukan

[1] P. Zhu, L. Wen, D. Du, X. Bian, H. Fan, Q. Hu, and H. Ling, "Detection and tracking meet drones challenge," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 7380–7399, 2022, doi: 10.1109/TPAMI.2021.3119563.

[2] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 740–755, doi: 10.1007/978-3-319-10602-1_48.

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[4] 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," *Mach. Learn. Knowl. Extr.*, vol. 5, no. 4, pp. 1680–1716, 2023, doi: 10.3390/make5040083.

[5] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8," ver. 8.0.0, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics.

[6] S. Tan, Z. Duan, and L. Pu, "Multi-scale object detection in UAV images based on adaptive feature fusion," *PLoS ONE*, vol. 19, no. 3, art. e0300120, 2024, doi: 10.1371/journal.pone.0300120.

[7] L. Xu, Y. Zhao, Y. Zhai, L. Huang, and C. Ruan, "Small object detection in UAV images based on YOLOv8n," *Int. J. Comput. Intell. Syst.*, vol. 17, no. 1, 2024, doi: 10.1007/s44196-024-00632-3.

[8] G. Wang, Y. Chen, P. An, H. Hong, J. Hu, and T. Huang, "UAV-YOLOv8: A small-object-detection model based on improved YOLOv8 for UAV aerial photography scenarios," *Sensors*, vol. 23, no. 16, art. 7190, 2023, doi: 10.3390/s23167190.

[9] G. Zhang, Y. Peng, and J. Li, "YOLO-MARS: An enhanced YOLOv8n for small object detection in UAV aerial imagery," *Sensors*, vol. 25, no. 8, art. 2534, 2025, doi: 10.3390/s25082534.

[10] X. Zhai, Z. Huang, T. Li, H. Liu, and S. Wang, "YOLO-Drone: An optimized YOLOv8 network for tiny UAV object detection," *Electronics*, vol. 12, no. 17, art. 3664, 2023, doi: 10.3390/electronics12173664.

[11] F. C. Akyon, S. O. Altinuc, and A. Temizel, "Slicing aided hyper inference and fine-tuning for small object detection," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2022, pp. 966–970, doi: 10.1109/ICIP46576.2022.9897990.

[12] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 936–944, doi: 10.1109/CVPR.2017.106.

[13] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 3–19, doi: 10.1007/978-3-030-01234-2_1.

[14] Z. Tong, Y. Chen, Z. Xu, and R. Yu, "Wise-IoU: Bounding box regression loss with dynamic focusing mechanism," *arXiv preprint* arXiv:2301.10051, 2023, doi: 10.48550/arXiv.2301.10051.

[15] Z. Zheng, P. Wang, D. Ren, W. Liu, R. Ye, Q. Hu, and W. Zuo, "Enhancing geometric factors in model learning and inference for object detection and instance segmentation," *IEEE Trans. Cybern.*, vol. 52, no. 8, pp. 8574–8586, 2022, doi: 10.1109/TCYB.2021.3095305.

[16] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU loss: Faster and better learning for bounding

box regression," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 7, 2020, pp. 12993–13000, doi: 10.1609/aaai.v34i07.6999.

[17] I. P. Sary, S. Andromeda, and E. U. Armin, "Performance comparison of YOLOv5 and YOLOv8 architectures in human detection using aerial images," *Ultima Comput. J. Sist. Komput.*, vol. 15, no. 1, pp. 8–13, 2023, doi: 10.31937/sk.v15i1.3204.

[18] N. M. Muriyah, J. H. Sim, and A. Yulianto, "Evaluating YOLOv5 and YOLOv8: Advancements in human detection," *J. Inf. Syst. Informatics*, vol. 6, no. 4, 2024, doi: 10.51519/journalisi.v6i4.944.

[19] N. R. Sakhdiah and K. Fahmi, "Analisis perbandingan kinerja YOLOv8 dan YOLOv11 untuk deteksi UAV militer secara realtime," *J. Algoritma, Logika dan Komputasi*, vol. 9, no. 2, pp. 14–24, 2026, doi: 10.30813/j-alu.v9i2.10100.

[20] I. H. Al Amin and A. Aprilino, "Implementasi algoritma YOLO dan Tesseract OCR pada sistem deteksi plat nomor otomatis," *J. Teknoinfo*, vol. 16, no. 1, pp. 54–59, 2022, doi: 10.33365/jti.v16i1.1522.

[21] E. D. Hartono and B. Hardiansyah, "Improved YOLOv8 algorithms for object detection," in *Proc. ICOIACE*, Univ. 17 Agustus 1945 Surabaya, Surabaya, Indonesia, Oct. 31–Nov. 1, 2024. [Online]. Available: https://conference.untag-sby.ac.id/index.php/icoiace/article/view/5311.

[22] X. Li, W. Wang, L. Wu, S. Chen, X. Hu, J. Li, J. Tang, and J. Yang, "Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, 2020, pp. 21002–21012.

[23] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*. New York, NY, USA: Chapman & Hall, 1993.