

TESIS
INOVASI YOLO11 UNTUK DETEKSI TARGET PESAWAT
MILITER PADA CITRA PENGINDERAAN JARAK JAUH



disusun oleh
SATYO WIDIJANUARTO
24.51.1606
Konsentrasi : Business Intelligence

FAKULTAS ILMU KOMPUTER
UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA
2025

TESIS
INOVASI YOLO11 UNTUK DETEKSI TARGET PESAWAT
MILITER PADA CITRA PENGINDERAAN JARAK JAUH

INNOVATION OF YOLO11 FOR MILITARY AIRCRAFT TARGET
DETECTION IN REMOTE SENSING IMAGES

Diajukan untuk memenuhi salah satu syarat mencapai derajat Pascasarjana
Program Studi S2 Informatika



disusun oleh

SATYO WIDIJANUARTO

24.51.1606

Konsentrasi : Business Intelligence

FAKULTAS ILMU KOMPUTER
UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA
2025

HALAMAN PERSETUJUAN

INOVASI YOLO11 UNTUK DETEKSI TARGET
PESAWAT MILITER PADA CITRA PENGINDERAAN JARAK JAUH

INNOVATION OF YOLO11 FOR MILITARY AIRCRAFT TARGET
DETECTION IN REMOTE SENSING IMAGES

yang disusun dan diajukan oleh

Satyo Widijanuarto

24.51.1606

telah disetujui oleh Dosen Pembimbing Tesis
pada tanggal 3 Desember 2025

Dosen Pembimbing,


Prof. Dr. Ema Utami, S.Si., M.Kom.
NIK. 190302037

HALAMAN PENGESAHAN

INOVASI YOLO11 UNTUK DETEKSI TARGET
PESAWAT MILITER PADA CITRA PENGINDERAAN JARAK JAUH

INNOVATION OF YOLO11 FOR MILITARY AIRCRAFT TARGET
DETECTION IN REMOTE SENSING IMAGES

yang disusun dan diajukan oleh

Satyo Widijanuarto

24.51.1606

Telah dipertahankan di depan Dewan Penguji
pada tanggal 3 Desember 2025

Susunan Dewan Penguji

Nama Penguji

Dr. Andi Sunyoto, M.Kom.
NIK. 190302052

I Made Artha Agastya, S.T., M.Eng., Ph.D.
NIK. 190302352

Prof. Dr. Ema Utami, S.Si., M.Kom.
NIK. 190302037

Tanda Tangan

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer
tanggal 3 Desember 2025

DEKAN FAKULTAS ILMU KOMPUTER

Prof. Dr. Kusriani, M.Kom.
NIK. 190302106



HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : Satyo Widjanuarto
NIM : 24.51.1606

Menyatakan bahwa Tesis dengan judul berikut:

Inovasi YOLO11 Untuk Deteksi Target Pesawat Militer Pada Citra Penginderaan Jarak Jauh

Dosen Pembimbing : Prof. Dr. Ema Utami, S.Si., M.Kom.

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya.
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Dosen Pembimbing.
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini.
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta.
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi.

Yogyakarta, 3 Desember 2025.

Yang Menyatakan,



METERAI
TANDA
UNIVERSITAS AMIKOM YOGYAKARTA
NIDCFAND008006214

Satyo Widjanuarto

HALAMAN PERSEMBAHAN

Tesis ini saya persembahkan dengan penuh rasa hormat kepada Para Penjaga Langit Nusantara. Semoga inovasi ini dapat menjadi setitik sumbangsih bagi kemajuan teknologi pertahanan demi menjaga kedaulatan udara Republik Indonesia. Kepada Ibu Prof. Dr. Ema Utami, M.Kom., selaku pembimbing yang tak kenal lelah memberikan arahan, kritik yang membangun, dan wawasan yang mencerahkan. Bapak Prof. Dr. M. Suyanto, M.M., Bapak Dr. Andi Sunyoto, M.Kom., Bapak I Made Artha Agastya, S.T., M.Eng., Ph.D., Bapak Dhani Ariatmanto, M.Kom, Ph.D, Bapak M. Hanafi, S.Kom, M.Eng., Ph.D, dan Bapak Alva Hendi Muhammad, S.T., M.Eng., Ph.D, atas segala ilmu dan bimbingan yang tak ternilai. Almaterku, Universitas Amikom Yogyakarta. Istriku Puji Lestari, teman berbagi suka dan duka terbaik dalam perjalanan ini.



KATA PENGANTAR

Puji syukur kehadiran Tuhan Yang Maha Esa atas segala berkat dan rahmat karunia-Nya sehingga penulis dapat menyelesaikan tesis yang berjudul "Inovasi YOLO11 untuk Deteksi Target Pesawat Militer pada Citra Penginderaan Jarak Jauh" dengan baik dan tepat waktu. Tesis ini disusun sebagai salah satu syarat untuk memperoleh gelar Magister Komputer pada Fakultas Ilmu Komputer, Universitas Amikom Yogyakarta.

Penyusunan tesis ini tidak akan terwujud tanpa bimbingan, arahan, dan dukungan dari berbagai pihak. Oleh karena itu, dengan segala kerendahan hati, penulis ingin menyampaikan ucapan terima kasih yang tulus dan sebesar-besarnya kepada:

1. Yth. Ibu Prof. Dr. Ema Utami, M.Kom., selaku Pembimbing, yang telah dengan sabar meluangkan waktu, tenaga, dan pikiran untuk memberikan bimbingan, masukan, serta motivasi yang sangat berharga sejak awal hingga akhir penyusunan tesis ini.
2. Yth. Bapak Robert Marco, M.T., Ph.D, selaku Ketua Prodi S2 Komputer Reguler, atas segala arahan, diskusi ilmiah, dan pandangan kritis yang telah memperkaya serta menyempurnakan substansi penelitian ini.
3. Yth. Bapak Dr. Andi Sunyoto, M.Kom., selaku Ketua Sidang Ujian Tesis, beserta segenap Tim Dosen Penguji, atas kesediaan untuk menguji, memberikan pertanyaan, saran, dan kritik yang membangun demi perbaikan kualitas tesis ini.
4. Yth. Ibu Prof. Dr. Kusri, M.Kom., selaku Dekan Fakultas Ilmu Komputer dan seluruh jajaran dosen yang telah memberikan bekal ilmu pengetahuan yang tak ternilai selama masa perkuliahan.

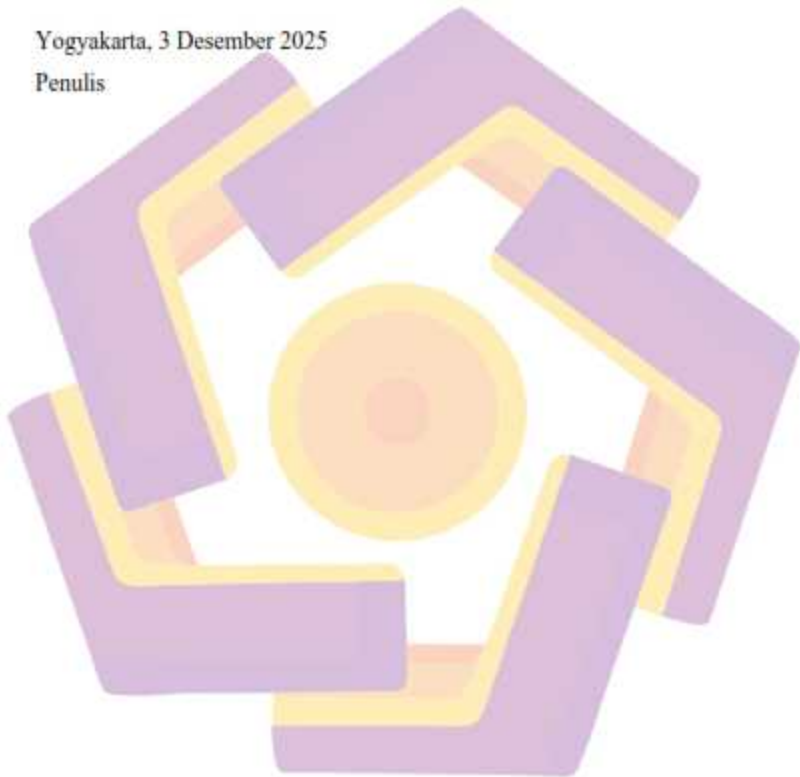
Seluruh staf administrasi di lingkungan Fakultas Ilmu Komputer yang telah memberikan bantuan dan kemudahan dalam berbagai urusan akademik maupun teknis selama proses penelitian berlangsung.

Penulis menyadari sepenuhnya bahwa tesis ini masih jauh dari kesempurnaan dan memiliki banyak kekurangan. Oleh karena itu, kritik dan saran

yang bersifat membangun akan penulis terima demi penyempurnaan di masa mendatang. Semoga tesis ini dapat memberikan manfaat dan kontribusi positif bagi pengembangan ilmu pengetahuan, khususnya di bidang pengolahan citra dan kecerdasan buatan.

Yogyakarta, 3 Desember 2025

Penulis



DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PERSETUJUAN.....	ii
HALAMAN PENGESAHAN.....	iii
HALAMAN PERNYATAAN KEASLIAN TESIS.....	iv
HALAMAN PERSEMBAHAN.....	v
KATA PENGANTAR.....	vi
DAFTAR ISI.....	viii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xi
INTISARI.....	xii
ABSTRACT.....	xiii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah.....	4
1.4 Tujuan Penelitian.....	6
1.5 Manfaat Penelitian.....	6
BAB 2 TINJAUAN PUSTAKA.....	9
2.1 Tinjauan Pustaka.....	9
2.2 Keaslian Penelitian.....	23
2.3 Landasan Teori.....	29
2.3.1 Arsitektur YOLO11.....	29
2.3.2 Hyperparameter Tunning.....	34
2.3.3 Optimizer.....	36
2.3.4 Global Attention Mechanism (GAM) Module.....	43
2.3.5 Modifikasi YOLO11 dengan Penambahan Modul GAM.....	45
BAB 3 METODE PENELITIAN.....	48
3.1 Jenis, Sifat, dan Pendekatan Penelitian.....	48
3.2 Metode Pengumpulan Data.....	48
3.3 Metode Analisis Data.....	53
3.4 Alur Penelitian.....	55
3.5 Evaluasi.....	57
3.5.1 Penggunaan mAP dalam Deteksi Pesawat dengan YOLO.....	58
3.5.2 mAP.....	59
3.5.3 mAP50 dan mAP50-95.....	63
3.5.4 Keunggulan YOLO dalam Deteksi Pesawat:.....	65

BAB 4 HASIL PENELITIAN DAN PEMBAHASAN	66
4.1 Pengumpulan Dataset	66
4.2 Perangkat yang digunakan	66
4.3 Pra Pemrosesan Data	67
4.4 Augmentasi dan Split Data	68
4.5 Pelatihan Model	72
4.6 Hasil Eksperimen	74
4.6.2 Pelatihan YOLO1 In	80
4.6.3 Analisa Hasil Komparatif	84
4.6.4 Hasil Performa Terbaik berdasarkan mAP (Training)	87
4.6.5 Hasil Performa Terbaik berdasarkan mAP (Testing)	89
4.6.6 Confusion Matrix Performa Terbaik (Testing)	92
4.6.7 Perbandingan Hasil Test Visual Label dan Prediksi	94
4.6.8 Hasil Performa Terbaik berdasarkan Kecepatan Inferensi (FPS)	95
4.6.9 Hasil Peningkatan (Gain) Terbaik dengan Modul GAM	96
4.6.10 Hasil Perbandingan Performa dengan Penelitian Sebelumnya	98
4.6.11 Analisis dan Penerapan Perhitungan mAP	100
4.6.12 Analisis dan Perhitungan Akurasi	101
4.6.13 Analisis Superioritas Pelatihan 500 Epoch Berdasarkan Bukti Eksperimental	104
BAB 5 PENUTUP	108
5.1 Kesimpulan	108
5.2 Saran	111
DAFTAR PUSTAKA	113

DAFTAR TABEL

Tabel 2.1 Matriks literatur review dan posisi penelitian.....	23
Tabel 3.1 Modifikasi Nama Kelas	51
Tabel 3.2 Tabel Analisa untuk Learning Rate 0.001	53
Tabel 3.3 Tabel Analisa untuk Learning Rate 0.00147	53
Tabel 3.4 Tabel Analisa untuk Learning Rate 0.00417	54
Tabel 3.5 Tabel Hasil berdasarkan mAP Terbaik	54
Tabel 3.6 Tabel Hasil berdasarkan Kecepatan Terbaik	54
Tabel 3.7 Tabel Hasil Peningkatan (Gain) Terbaik pada mAP50 dan mAP50-95 setelah dimodifikasi.....	55
Tabel 4.1 Modifikasi Nama Kelas	68
Tabel 4.2 Hasil pelatihan dengan learning rate 0,001	80
Tabel 4.3 Hasil pelatihan dengan learning rate 0,00147	82
Tabel 4.4 Hasil pelatihan dengan learning rate 0,00417	83
Tabel 4.5 Hasil performa terbaik berdasarkan mAP (Training)	87
Tabel 4.6 Hasil performa terbaik berdasarkan mAP (Testing)	89
Tabel 4.7 Hasil performa terbaik berdasarkan FPS	96
Tabel 4.8 Hasil peningkatan performa (Gain) terbaik menggunakan modul GAM	97
Tabel 4.9 Perbandingan performa dengan penelitian Chen [19]	99
Tabel 4.10 Perbandingan performa dengan penelitian Wang [66] dan Chen [19]	99
Tabel 4.11 Perbandingan hasil tes performa dengan epoch 500	104
Tabel 4.12 Perbandingan model terbaik dari hasil tes 80 epoch dengan 500 epoch	105

DAFTAR GAMBAR

Gambar 2.1 Arsitektur YOLO11	29
Gambar 2.2 Perbandingan modul C2F pada YOLOv8 dengan modul C3K2 pada YOLO11	30
Gambar 2.3 Modul Spatial Pyramid Pooling Fast (SPPF).....	31
Gambar 2.4 Modul C2 Position Sensitive Attention Block (C2PSA)	33
Gambar 2.5 Global Attention Mechanism	43
Gambar 2.6 Letak Modul GAM pada Arsitektur YOLO11	46
Gambar 3.1 20 tipe pesawat militer (A1-A20)	48
Gambar 3.2 Contoh tiga gambar dari dataset MAR20.....	49
Gambar 3.3 Perbandingan hasil mAP pada dataset MAR20 dari berbagai model	50
Gambar 3.4 Alur Penelitian	56
Gambar 3.5 Ilustrasi skor IOU	61
Gambar 4.1 Contoh dataset MAR20.....	66
Gambar 4.2 Hasil augmentasi yang menghasilkan dua citra baru dengan beberapa keadaan	69
Gambar 4.3 Jumlah Anotasi per Kelas.....	70
Gambar 4.4 Perbandingan antara varian YOLO11	73
Gambar 4.5 Alur pelatihan menggunakan YOLO11 varian n	73
Gambar 4.6 Tangkapan layar proses pelatihan menggunakan YOLO11 asli	79
Gambar 4.7 Tangkapan layar proses pelatihan menggunakan YOLO11 yang telah dimodifikasi	80
Gambar 4.8 Hasil metrik pelatihan/validasi berdasarkan performa terbaik Peringkat 1	88
Gambar 4.9 Confusion matrix dari hasil performa terbaik (testing)	92
Gambar 4.10 Visual label asli (a) vs hasil prediksi model (b) berdasar hasil testing terbaik (mAP50-95).....	94

INTISARI

Penelitian ini berfokus pada pengembangan dan evaluasi algoritma deteksi objek yang superior untuk mengidentifikasi target pesawat militer pada citra penginderaan jarak jauh yang memiliki latar belakang kompleks. Inovasi utama yang diusulkan adalah integrasi empat buah modul Global Attention Mechanism (GAM) ke dalam arsitektur YOLO11, yang bertujuan untuk meningkatkan kemampuan ekstraksi fitur dan menekan interferensi dari latar belakang yang kompleks. Melalui serangkaian eksperimen dan tuning parameter yang sistematis, konfigurasi optimal berhasil dicapai dengan menggunakan optimizer NAdam dan learning rate 0,00147, yang menghasilkan kinerja puncak pada data pengujian dengan nilai mAP50-95 sebesar 80,6% dan mAP50 sebesar 98,6%. Hasil ini menunjukkan peningkatan presisi lokalisasi yang signifikan dibandingkan versi YOLO11 asli (mAP50-95 79% dan mAP50 98,4%) dan secara absolut mengungguli sebelas metode perbandingan state-of-the-art lainnya. Terlebih lagi, model usulan menunjukkan efisiensi komputasi yang luar biasa dengan kecepatan inferensi 159 FPS, lebih dari dua kali lipat kecepatan rata-rata metode perbandingannya. Meskipun terdapat trade-off berupa penurunan kecepatan dibandingkan model asli (250 FPS), analisis kualitatif dan confusion matrix mengonfirmasi ketangguhan model, dengan tantangan utama yang tersisa pada misidentifikasi antara sub-tipe pesawat yang visualnya sangat mirip. Secara keseluruhan, penelitian ini membuktikan bahwa arsitektur YOLO11 yang dimodifikasi dengan GAM merupakan solusi yang sangat efektif dan seimbang, menawarkan presisi deteksi yang superior. Saran untuk penelitian mendatang berfokus pada eksplorasi mekanisme atensi yang lebih ringan dan pengembangan metode untuk meningkatkan akurasi klasifikasi fine-grained.

Kata kunci: YOLO11, parameter tuning, deteksi pesawat militer, penginderaan jarak jauh, Global Attention Mechanism.

ABSTRACT

This research addresses the challenge of developing and evaluating a superior object detection algorithm for identifying military aircraft in remote sensing imagery characterized by complex backgrounds. The primary innovation proposed is the integration of four Global Attention Mechanism (GAM) Modules into the YOLO11 architecture, aimed at enhancing feature extraction capabilities and suppressing background interference. Through a series of systematic experiments and parameter tuning, the optimal configuration was achieved using the NAdam optimizer with a learning rate of 0.00147, yielding a peak performance on the test data with a mAP50-95 of 80.6% and a mAP50 of 98.6%. This result demonstrates a significant improvement in localization precision compared to the original YOLO11 version (79% mAP50-95 and 98.4% mAP50) and decisively outperforms eleven other state-of-the-art comparative methods. Furthermore, the proposed model exhibits remarkable computational efficiency with an inference speed of 159 FPS, more than double the average speed of the compared methods. Although a trade-off exists in the form of reduced speed compared to the original model (250 FPS), qualitative and confusion matrix analyses confirm the model's robustness, with the main remaining challenge being the misidentification between visually similar aircraft subtypes. Overall, this research validates the YOLO11+GAM architecture as a highly effective and balanced solution offering superior detection precision. Suggestions for future research include exploring more lightweight attention mechanisms and developing methods to improve fine-grained classification accuracy.

Keyword: YOLO11, parameter tuning, military aircraft detection, remote sensing images, Global Attention Mechanism.

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Deteksi target pesawat udara pada citra penginderaan jarak jauh merupakan komponen krusial dalam menjamin keselamatan dan keamanan lalu lintas udara. Situasi yang tidak terduga, seperti intrusi di landasan pacu, dapat menyebabkan kecelakaan serius yang mengancam keselamatan penerbangan. Oleh karena itu, sistem deteksi yang akurat dan efisien sangat diperlukan untuk pengoperasian bandara dan manajemen wilayah udara yang aman [1][2]. Dalam dunia militer, kecepatan sistem deteksi dan pengenalan target pesawat tidak hanya penting untuk mencegah ancaman tetapi juga untuk mendukung pengambilan keputusan militer dan pengintaian intelijen [2][3].

Dengan kemajuan teknologi, metode deteksi pesawat semakin canggih. Penggunaan radar apertur sintesis (SAR) dan algoritma deep learning telah meningkatkan kemampuan deteksi dalam berbagai kondisi, termasuk cuaca buruk dan latar belakang yang rumit. Teknologi ini tak hanya meningkatkan akurasi tetapi juga kecepatan deteksi, yang sangat penting untuk aplikasi waktu nyata [2][4][5][6]. Perkembangan teknologi ini membuka kemungkinan baru dalam peningkatan keselamatan serta efisiensi operasional di bandara.

Namun deteksi target pesawat pada citra penginderaan jarak jauh menghadapi sejumlah tantangan signifikan, terutama ketika berhadapan dengan target obyek yang kecil, kompleksitas latar belakang, dan berbagai kondisi

penerbangan. Keterbatasan akurasi dalam mendeteksi pesawat sering kali disebabkan oleh noise latar belakang dan kesulitan dalam mengekstraksi fitur dari citra yang berkualitas rendah. Dengan pesawat yang semakin kecil dan bervariasi, serta lingkungan operasional yang kompleks, pendekatan tradisional tidak lagi memadai untuk memenuhi kebutuhan mendeteksi target dengan efisiensi dan keakuratan yang tinggi. Penelitian sebelumnya telah menunjukkan bahwa banyak algoritma deteksi, seperti variasi YOLO (You Only Look Once), telah diterapkan untuk meningkatkan presisi, meskipun masih terdapat ruang untuk perbaikan dalam hal akurasi dan computational efficiency [7][8][9][10].

Masalah ini penting untuk diselesaikan karena ketidakakuratan dalam deteksi pesawat dapat mengakibatkan konsekuensi serius, terutama dalam konteks militer dan keselamatan penerbangan sipil. Pilihan untuk tidak meningkatkan sistem deteksi dapat membawa risiko peningkatan kecelakaan dan pengawasan yang tidak efektif terhadap lalu lintas dan keamanan udara. Oleh karena itu, inovasi dalam algoritma deteksi menawarkan potensi untuk meningkatkan keselamatan operasional dengan mendukung pengambilan keputusan yang lebih baik dan lebih cepat dalam situasi kritis.

Motivasi dari penelitian ini adalah untuk mengembangkan algoritma deteksi yang lebih akurat dan efisien, yang mampu menangani tantangan-tantangan yang ada dalam mendeteksi pesawat, bahkan dalam kondisi lingkungan yang sulit. Dengan mempertimbangkan kemajuan teknologi terbaru dalam deep learning dan arsitektur jaringan, penelitian ini bertujuan untuk menciptakan model yang tidak hanya meningkatkan akurasi deteksi tetapi juga efisiensi komputasi.

Tujuan penelitian ini adalah untuk mengeksplorasi dan mengembangkan model deteksi yang inovatif berdasarkan algoritma yang sudah ada, dengan fokus pada kerangka YOLO11 yang telah terbukti efektif dalam deteksi objek [11]. Penelitian ini bertujuan untuk menyempurnakan algoritma yang ada dengan menerapkan teknik-teknik baru untuk meningkatkan kemampuan ekstraksi fitur dan menghasilkan akurasi deteksi yang lebih tinggi dalam kondisi tampilan citra yang bervariasi.

Kontribusi penelitian ini diharapkan meliputi pengembangan algoritma deteksi yang lebih ringan dan efisien, serta meningkatkan kemampuan deteksi pesawat dalam citra penginderaan jarak jauh, yang dapat memberikan solusi nyata untuk tantangan dalam industri penerbangan dan pertahanan keamanan. Dengan membandingkan dan mengintegrasikan fitur terbaik dari algoritma yang sudah ada, penelitian ini akan memberikan wawasan baru bagi pengembangan teknologi deteksi pesawat yang lebih adaptif dan responsif terhadap kebutuhan operasional yang berkembang.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah diidentifikasi, rumusan masalah dalam proposal tesis ini adalah sebagai berikut:

1. Bagaimana cara mengembangkan algoritma deteksi berbasis YOLO yang mampu meningkatkan akurasi dan efisiensi dalam mendeteksi pesawat militer di citra penginderaan jarak jauh, terutama dalam kondisi lingkungan dan latar belakang yang kompleks?

2. Bagaimana model YOLO11 yang telah dimodifikasi meningkat dibandingkan YOLO11 asli dalam hal akurasi dan nilai mAP?

1.3 Batasan Masalah

Dalam penelitian ini, batasan masalah akan difokuskan pada beberapa aspek utama untuk menjamin kedalaman dan kejelasan analisis.

1. Penelitian akan menggunakan model deteksi YOLO (You Only Look Once) 11 dari Ultralytics sebagai kerangka utama. Model ini dipilih karena kemampuannya yang telah terbukti dalam deteksi objek secara real-time dan fleksibilitas dalam penyesuaian struktur.
2. Dataset yang digunakan adalah MAR20 (Military Aircraft Recognition) yang terdiri dari 20 kelas pesawat militer. Penggunaan dataset ini memungkinkan peneliti untuk mengeksplorasi kemampuan deteksi dalam konteks spesifik pengenalan pesawat militer dengan latar belakang yang beragam dan kompleks.
3. Penelitian ini akan difokuskan pada optimalisasi model melalui penalaan parameter (parameter tuning) yang meliputi:
 - optimizer: Adam, Adamax, AdamW, NAdam, RAdam, SGD
 - learning rate: 0,001; 0,00147; 0,00417
 - epoch: 80 dan 500
 - batch size : 16
 - momentum : 0,9
 - IoU-threshold : 0,7

- Weight decay : 0,0005
- modifikasi struktur: penambahan 4 buah modul Global Attention Mechanism (GAM).

Modifikasi ini sebagai langkah inovatif untuk meningkatkan akurasi dan efisiensi. Mencakup pengenalan modul perhatian dan fusi fitur yang bertujuan untuk meningkatkan kemampuan model dalam menanggapi tantangan mendeteksi pesawat kecil dalam citra yang memiliki noise latar belakang yang tinggi.

4. Spesifikasi hardware yang digunakan:

- CPU AMD Ryzen 5 5600
- Memory DDR 32 GB
- NVIDIA GeForce RTX 3050 8 GB

5. Spesifikasi software yang digunakan:

- OS Ubuntu 24.04.1 LTS
- Visual Studio Code v.1.99.3
- Ultralytics 8.3.165
- Python 3.12.3 torch 2.5.1+cu124 CUDA

Dengan batasan-batasan ini, penelitian diharapkan mampu menjawab pertanyaan-pertanyaan yang telah dirumuskan dan memberikan kontribusi terhadap pengembangan algoritma deteksi yang lebih bermanfaat dalam bidang penginderaan jarak jauh khususnya untuk deteksi target pesawat militer.

1.4 Tujuan Penelitian

1. Menemukan formulasi parameter tuning terbaik untuk deteksi target pesawat dari beberapa percobaan berikut:
 - Epoch : 80
 - Learning rate : 0,001; 0,00147 dan 0.00417
 - Optimizer : Adam, Adamax, AdamW, NAdam, RAdam, SGD
2. Hasil yang akan dicapai: mAP dan kecepatan deteksi yang terbaik.
3. Menemukan inovasi penelitian dengan mengeksplorasi dan memodifikasi struktur model deteksi YOLO11.

1.5 Manfaat Penelitian

Penelitian ini diharapkan memberikan sejumlah manfaat yang signifikan, baik dalam konteks akademis maupun praktis. Berikut adalah beberapa manfaat utama yang diharapkan dari penelitian ini:

1. Peningkatan mAP, Akurasi dan Efisiensi Deteksi

Dengan mengembangkan algoritma deteksi yang lebih presisi, akurat dan efisien, penelitian ini bertujuan untuk meningkatkan kemampuan mendeteksi pesawat militer dalam citra penginderaan jarak jauh. Hasil ini diharapkan dapat mengurangi tingkat kesalahan deteksi, yang penting untuk memperkuat sistem pertahanan dan strategi militer.

2. Inovasi dalam Pengembangan Model Deteksi

Penelitian ini berkontribusi terhadap inovasi dalam teknologi deteksi objek dengan mengeksplorasi struktur model YOLO11 yang baru. Dengan memodifikasi dan mengoptimalkan arsitektur yang ada,

diharapkan dapat menciptakan model yang lebih baik dalam menangani tantangan deteksi pesawat, khususnya dalam kondisi yang sulit.

3. Optimalisasi Parameter

Melalui penelitian hyperparameter tuning, hasil yang didapatkan akan memberikan gambaran jelas mengenai kombinasi optimum dari epoch, learning rate, dan optimizer. Penemuan ini tidak hanya berguna bagi penelitian ini tetapi juga memberikan panduan untuk penelitian dan aplikasi lain di masa depan dalam pengembangan model deteksi.

4. Dukungan untuk Pengambilan Keputusan Berbasis Data

Hasil penelitian ini akan memperkuat dasar pengambilan keputusan dalam aplikasi militer dan sipil dengan menyediakan data akurat mengenai mAP, Recall, Precision, dan Accuracy. Ini akan membantu pemangku kepentingan dalam mengimplementasikan solusi yang lebih efektif untuk pengawasan lalu lintas udara dan situasi pertahanan keamanan.

5. Bermanfaat untuk Penelitian Lanjutan

Penelitian ini menyediakan basis data yang kuat bagi penelitian lanjutan di bidang deteksi objek dengan penerapan teknologi deep learning. Temuan dan inovasi yang dihasilkan dapat menjadi referensi bagi peneliti lain yang mengeksplorasi teknik dan algoritma baru dalam penginderaan jarak jauh dan bidang terkait.

6. Kontribusi terhadap Bidang Pertahanan dan Keamanan Nasional

Dengan meningkatkan sistem deteksi pesawat, penelitian ini berkontribusi langsung terhadap pertahanan dan keamanan nasional. Pengembangan algoritma yang mampu bekerja dengan baik dalam kondisi yang menantang dapat mengurangi risiko keamanan dan meningkatkan efektivitas dalam memperkuat sistem pertahanan dan strategi militer.

Melalui penerapan manfaat-manfaat ini, penelitian diharapkan dapat memberikan kontribusi nyata baik untuk pengembangan teknologi deteksi pesawat maupun untuk peningkatan pertahanan dan keamanan negara.



BAB 2

TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Penelitian Luo et al. [12] mengembangkan model deteksi target pesawat yang dikenal sebagai YOLOv5-Aircraft, yang merupakan perbaikan dari model YOLOv5 untuk meningkatkan akurasi dan kecepatan deteksi pesawat dalam citra penginderaan jarak jauh. Dengan kemajuan teknologi penginderaan jarak jauh, citra satelit kini dapat menampilkan detail yang lebih jelas, termasuk objek sensitif seperti pesawat, yang penting untuk aplikasi sipil dan militer. Namun, deteksi pesawat dalam citra ini tetap menjadi tantangan karena dipengaruhi oleh faktor eksternal seperti cuaca, pencahayaan, dan kompleksitas latar belakang. Untuk mengatasi masalah ini, model YOLOv5-Aircraft dioptimalkan melalui tiga pendekatan utama: pertama, penambahan kalibrasi centering dan scaling pada modul normalisasi batch untuk memperkuat kemampuan ekstraksi fitur jaringan; kedua, penggantian fungsi loss cross-entropy dengan fungsi loss berbasis smoothed Kullback-Leibler divergence untuk meningkatkan stabilitas konvergensi; dan ketiga, pengenalan modul CSandGlass untuk mengurangi kehilangan informasi dalam proses ekstraksi fitur.

Hasil eksperimen menunjukkan bahwa YOLOv5-Aircraft berhasil meningkatkan akurasi deteksi hingga 85,25% dan kecepatan deteksi mencapai 48,85 FPS, yang menunjukkan peningkatan signifikan dibandingkan dengan model sebelumnya. Selain itu, model ini menunjukkan ketahanan yang baik dalam

menghadapi kondisi kompleks, seperti pencahayaan yang bervariasi dan objek yang terhalang, yang sering menjadi masalah dalam deteksi target kecil. Penelitian ini memberikan kontribusi penting dalam pengembangan algoritma deteksi target pesawat, dengan hasil yang menjanjikan untuk aplikasi dalam pengawasan dan manajemen lalu lintas udara, serta meningkatkan kemampuan deteksi dalam situasi yang rumit.

Penelitian Li et al. [13] mengusulkan algoritma deteksi fitur permukaan yang inovatif untuk citra penginderaan jarak jauh dengan memanfaatkan struktur jaringan YOLOX. Dalam konteks ini, penulis memperkenalkan beberapa mekanisme perbaikan untuk meningkatkan akurasi deteksi objek multi-skala yang sering kali menjadi tantangan dalam citra penginderaan jarak jauh. Pertama, peneliti mengintegrasikan Adaptively Spatial Feature Fusion (ASFF) ke dalam modul ekstraksi fitur PANET untuk mengatasi masalah deteksi objek multi-skala dengan menyaring informasi yang bertentangan di ruang. Kedua, attention mechanism berbasis Convolutional Block Attention Module (CBAM) dirancang untuk lebih menekankan informasi fitur positif dalam peta fitur, sehingga mengurangi dampak informasi yang tidak relevan.

Selanjutnya, penelitian ini memperkenalkan dua fungsi loss baru, yaitu EIoU loss dan VariFocalLoss, yang bertujuan untuk mencerminkan kesalahan jarak yang sebenarnya antara dua target yang diprediksi dan meningkatkan kinerja deteksi multi-target. Hasil eksperimen yang dilakukan pada dataset DOTA menunjukkan bahwa model jaringan yang ditingkatkan mencapai mAP (mean

Average Precision) sebesar 70,75%, yang merupakan peningkatan sebesar 9,08% dibandingkan dengan model asli.

Penelitian ini juga melakukan eksperimen ablation dan perbandingan dengan algoritma deteksi target utama lainnya, yang menunjukkan bahwa metode yang diusulkan efektif dalam meningkatkan akurasi deteksi objek, terutama dalam kategori objek yang padat. Meskipun demikian, penulis mencatat adanya tantangan seperti overfitting dan generalisasi yang buruk, terutama pada data sampel kecil, yang menunjukkan perlunya pengembangan lebih lanjut untuk meningkatkan kinerja deteksi dan klasifikasi.

Penelitian Shen [14] mengembangkan model CA-YOLO (Coordinate Attention-YOLO) yang dioptimalkan untuk deteksi objek dalam citra penginderaan jarak jauh yang kompleks. Model ini bertujuan untuk mengatasi masalah utama yang dihadapi oleh algoritma deteksi objek, seperti kemampuan pembelajaran fitur multi-skala yang lemah dan trade-off antara akurasi deteksi dan kompleksitas parameter model. CA-YOLO dibangun di atas kerangka kerja YOLOv5 dan mengintegrasikan modul perhatian koordinat yang ringan di lapisan dangkal untuk meningkatkan ekstraksi fitur detail dan mengurangi gangguan informasi yang tidak relevan. Selain itu, model ini menerapkan Spatial Pyramid Pooling-Fast (SPPF) dengan modul konstruksi tandem di lapisan yang lebih dalam, serta strategi pooling stokastik untuk menggabungkan informasi fitur kunci multi-skala dari lapisan rendah hingga tinggi, yang mengurangi jumlah parameter model sambil meningkatkan kecepatan inferensi.

Hasil eksperimen menunjukkan bahwa model CA-YOLO mengungguli YOLOv5 dalam hal akurasi deteksi objek multi, dengan peningkatan rata-rata $mAP@0.5$ sebesar 4,8% dan $mAP@0.5:0.95$ sebesar 3,8%. Model ini juga menunjukkan kecepatan inferensi yang luar biasa, rata-rata mencapai 125 FPS, yang menegaskan keunggulannya dalam akurasi deteksi, kemampuan generalisasi, dan efisiensi keseluruhan. Dengan mengoptimalkan mekanisme anchor box dan memodifikasi fungsi loss menjadi $SIoU_loss$, CA-YOLO meningkatkan kemampuan model untuk mendeteksi objek dengan berbagai ukuran dan skala, serta mempercepat konvergensi selama pelatihan. Penelitian ini memberikan kontribusi signifikan dalam pengembangan algoritma deteksi objek untuk aplikasi di berbagai bidang, termasuk pemantauan lingkungan, perencanaan kota, dan keamanan publik, dengan hasil yang menjanjikan untuk meningkatkan akurasi dan efisiensi dalam deteksi objek dalam citra penginderaan jarak jauh.

Penelitian Zhang [15] mengusulkan algoritma YOLOv5 yang ditingkatkan untuk deteksi target pesawat dalam citra penginderaan jarak jauh, yang bertujuan untuk mengatasi masalah deteksi yang sering terjadi, seperti kesalahan deteksi dan ketidakakuratan akibat pengaturan pesawat yang padat, kondisi latar belakang yang kompleks, sudut sikap yang beragam, dan banyaknya target kecil. Dalam pengembangan model ini, beberapa inovasi diterapkan, termasuk penambahan modul mekanisme perhatian koordinat pada jaringan backbone untuk meningkatkan kemampuan ekstraksi fitur dan mengurangi pengaruh informasi yang tidak relevan. Selain itu, lapisan fusi fitur multi-skala diperbaiki untuk meningkatkan kemampuan persepsi global dan ekstraksi fitur dari target pesawat.

Fungsi kerugian EIoU juga digunakan untuk meningkatkan akurasi regresi lokasi, yang berkontribusi pada peningkatan kinerja model secara keseluruhan.

Hasil eksperimen menunjukkan bahwa algoritma yang diusulkan berhasil meningkatkan nilai Average Precision (AP) pada berbagai metrik, termasuk AP50 dan AP75, masing-masing sebesar 3,1% dan 4,9% dibandingkan dengan algoritma YOLOv5 asli. Selain itu, model ini mencapai kecepatan deteksi 34,5 FPS, yang menunjukkan keseimbangan yang baik antara akurasi dan kecepatan. Pengujian dilakukan menggunakan dataset gabungan dari DOTA dan DIOR, yang mencakup 3.306 gambar dengan 25.897 instance pesawat, yang menunjukkan bahwa algoritma ini efektif dalam meningkatkan akurasi deteksi dan mengurangi tingkat kesalahan deteksi dalam kondisi yang menantang.

Secara keseluruhan, penelitian ini memberikan kontribusi signifikan terhadap pengembangan teknologi deteksi objek dalam citra penginderaan jarak jauh, dengan menunjukkan bahwa perbaikan dalam struktur model dan fungsi kerugian dapat secara substansial meningkatkan kinerja deteksi pesawat. Temuan ini memiliki nilai praktis yang tinggi untuk aplikasi di bidang sipil dan militer, seperti pengelolaan lalu lintas udara dan pengintaian militer. Penelitian ini juga membuka peluang untuk pengembangan lebih lanjut dalam optimasi kecepatan deteksi sambil tetap mempertahankan akurasi, dengan mempertimbangkan metode ekstraksi dan fusi fitur yang lebih efektif di masa depan.

Penelitian Huang et al. [16] mengembangkan model deteksi target pesawat dalam citra Synthetic Aperture Radar (SAR) yang dikenal sebagai EST-YOLOv5s, yang dirancang untuk mengatasi tantangan deteksi pesawat dengan skala yang

beragam dan gangguan latar belakang yang kuat. Model ini mengintegrasikan mekanisme Efficient Channel Attention (ECA) ke dalam modul C3 dari jaringan backbone YOLOv5s, yang meningkatkan fitur pencitraan pesawat dan menekan informasi latar belakang yang tidak relevan tanpa menambah jumlah parameter. Selain itu, model ini menggantikan modul bottleneck di modul C3 terakhir dengan Swin Transformer Block, yang menggunakan pendekatan pembagian jendela bergeser untuk meningkatkan kemampuan persepsi global dan mengurangi masalah deteksi objek kecil. Untuk lebih meningkatkan kinerja, model ini juga menerapkan Task-Specific Context Decoupling (TSCODE) head, yang membagi tugas deteksi menjadi dua sub-tugas: klasifikasi objek dan regresi bounding box, sehingga memungkinkan pemanfaatan konteks yang lebih baik.

Hasil eksperimen menunjukkan bahwa model EST-YOLOv5s mencapai tingkat recall sebesar 94,2%, presisi 97,3%, dan mean Average Precision (mAP) @50 sebesar 97,8%, yang masing-masing lebih tinggi 2,3%, 1,7%, dan 1,7% dibandingkan dengan model baseline YOLOv5s. Model ini juga memenuhi persyaratan real-time dengan kecepatan deteksi yang baik dan menunjukkan kemampuan anti-gangguan yang kuat, yang sangat penting dalam konteks pengawasan militer dan pemantauan lingkungan. Penelitian ini memberikan kontribusi signifikan dalam pengembangan algoritma deteksi target pesawat, dengan hasil yang menjanjikan untuk aplikasi di bidang keamanan dan pemantauan, serta meningkatkan kemampuan deteksi dalam kondisi yang kompleks.

Penelitian Liu et al. [10] memperkenalkan algoritma YOLO-Extract, yang merupakan pengembangan dari model YOLOv5, untuk meningkatkan akurasi

deteksi dan klasifikasi target pesawat dalam citra penginderaan jarak jauh. Citra satelit sering kali menampilkan target yang kecil dan padat, serta latar belakang yang kompleks, yang menyulitkan proses deteksi. YOLO-Extract mengoptimalkan struktur model YOLOv5 dengan menghapus lapisan fitur dan kepala prediksi yang memiliki kemampuan ekstraksi fitur yang buruk, serta mengintegrasikan extractor fitur baru yang lebih kuat ke dalam jaringan. Selain itu, penelitian ini mengadopsi mekanisme perhatian koordinat untuk meningkatkan akurasi deteksi dengan memfokuskan perhatian pada area yang lebih luas, sehingga mengurangi beban komputasi dan meningkatkan kemampuan model dalam mengenali target pesawat di latar belakang yang rumit.

Dalam eksperimen yang dilakukan, YOLO-Extract menunjukkan peningkatan yang signifikan dalam kecepatan konvergensi, dengan pengurangan jumlah parameter sebesar 10,526 juta dan pengurangan beban komputasi sebesar 45,3 GFLOPs, sambil meningkatkan mean Average Precision (mAP) sebesar 8,1% dan kecepatan deteksi hingga tiga kali lipat dibandingkan dengan YOLOv5. Penelitian ini juga mengimplementasikan Focal- α EIoU Loss sebagai fungsi loss untuk mempercepat regresi bounding box, yang terbukti efektif dalam meningkatkan akurasi deteksi target kecil dan padat.

Hasil evaluasi menunjukkan bahwa YOLO-Extract tidak hanya lebih cepat dalam konvergensi tetapi juga lebih akurat dalam mendeteksi target pesawat dalam berbagai kondisi, termasuk cuaca buruk dan latar belakang yang kompleks. Dengan demikian, penelitian ini memberikan kontribusi yang signifikan terhadap pengembangan algoritma deteksi target dalam citra penginderaan jarak jauh,

dengan potensi aplikasi yang luas dalam pengawasan militer dan manajemen lalu lintas udara.

Penelitian Liu [8] mengembangkan model YOLO-Class yang bertujuan untuk meningkatkan deteksi dan klasifikasi target pesawat dalam citra penginderaan satelit jarak jauh. Dengan kemajuan teknologi penginderaan jarak jauh, citra satelit menjadi sumber informasi penting mengenai permukaan bumi, namun tantangan seperti sampel data yang tidak seimbang, variasi skala target yang besar, dan occlusion (penutupan) target menyebabkan rendahnya akurasi dan kecepatan deteksi. Model YOLO-Class diusulkan sebagai solusi untuk masalah ini dengan mengoptimalkan model YOLO-Extract [10] yang sebelumnya telah ada. Pertama, penelitian ini menerapkan Representative Batch Normalization (RBN) dan fungsi aktivasi Mish untuk meningkatkan modul konvolusi, serta menggunakan VariFocal loss untuk mengatasi masalah ketidakseimbangan data dalam fungsi loss klasifikasi.

Hasil simulasi menunjukkan bahwa YOLO-Class berhasil meningkatkan akurasi deteksi dari 60,8% menjadi 70,4% dan kecepatan pemrosesan dari 36,16 FPS menjadi 39,598 FPS dibandingkan dengan model YOLO-Extract. Penelitian ini juga mengintegrasikan modul RepVGG dalam backbone untuk lebih meningkatkan akurasi deteksi tanpa menambah jumlah parameter secara signifikan. Eksperimen menunjukkan bahwa model yang diusulkan dapat mendeteksi target pesawat dalam berbagai kondisi latar belakang yang kompleks, cuaca berkabut, dan pencahayaan yang berbeda, serta menunjukkan kinerja yang lebih baik dalam mendeteksi target kecil.

Secara keseluruhan, penelitian ini memberikan kontribusi signifikan dalam pengembangan metode deteksi dan klasifikasi target pesawat dalam citra penginderaan jarak jauh, dengan hasil yang menjanjikan untuk aplikasi di bidang pengintaian militer, manajemen lalu lintas udara, dan navigasi otomatis. Namun, penulis mencatat bahwa meskipun ada peningkatan yang signifikan, masih terdapat ruang untuk perbaikan lebih lanjut dalam akurasi deteksi.

Penelitian Xu et al. [17] mengusulkan algoritma deteksi target pesawat yang ditingkatkan berdasarkan model YOLOv8, yang dirancang untuk mengatasi tantangan deteksi pesawat dalam citra penginderaan jarak jauh, terutama terkait dengan ukuran target yang kecil dan kompleksitas latar belakang. Model ini memperkenalkan C2f_Former sebagai pengganti modul C2f dalam jaringan backbone, yang memanfaatkan konvolusi kernel besar untuk meningkatkan kemampuan ekstraksi fitur dari citra resolusi tinggi dan mengurangi jumlah parameter model. Selain itu, modul EMA (Efficient Multi-Scale Attention) ditambahkan pada bagian Neck untuk memperkuat informasi semantik dan posisi dalam fitur, sehingga meningkatkan kemampuan fusi fitur model. Penelitian ini juga mengoptimalkan fungsi loss dengan memperkenalkan Wise-IoU, yang memberikan strategi penugasan gradien yang lebih baik, sehingga meningkatkan kinerja regresi bounding box.

Hasil eksperimen menunjukkan bahwa model yang ditingkatkan mencapai tingkat recall sebesar 93,1%, yang lebih tinggi 0,9% dibandingkan dengan model sebelum perbaikan, dan rata-rata akurasi deteksi (mAP) sebesar 75,2%, meningkat 1,8% dari model sebelumnya. Selain itu, jumlah parameter model berkurang

sebesar 0,430G, menunjukkan bahwa model ini tidak hanya lebih efisien tetapi juga lebih efektif dalam mendeteksi pesawat dalam citra penginderaan jarak jauh. Penelitian ini memberikan kontribusi signifikan dalam pengembangan algoritma deteksi target pesawat, dengan hasil yang menjanjikan untuk aplikasi dalam pengawasan dan keamanan, serta meningkatkan kemampuan deteksi dalam situasi yang kompleks.

Penelitian Wang et al. [18] mengembangkan algoritma deteksi target untuk kendaraan udara tak berawak (UAV) yang berfokus pada deteksi target kecil, dengan memanfaatkan model YOLOv8 yang telah dimodifikasi. UAV sering kali beroperasi pada ketinggian tinggi, yang menyebabkan citra yang dihasilkan memiliki proporsi target kecil yang signifikan, serta masalah kabur akibat kecepatan terbang yang tinggi. Untuk mengatasi tantangan ini, penulis mengintegrasikan struktur deteksi target kecil (Small Target Connection, STC) ke dalam jaringan, yang berfungsi sebagai jembatan antara fitur dangkal dan dalam, sehingga meningkatkan pengumpulan informasi semantik dari target kecil dan meningkatkan akurasi deteksi.

Selain itu, penelitian ini memperkenalkan mekanisme perhatian global (Global Attention Mechanism, GAM) pada lapisan dasar backbone YOLOv8 untuk mencegah kehilangan informasi fitur citra selama proses pengambilan sampel, yang pada gilirannya meningkatkan kinerja deteksi algoritma. Hasil eksperimen yang dilakukan pada dataset VisDrone2021 menunjukkan bahwa model yang dimodifikasi mencapai nilai mAP (mean Average Precision) sebesar 39,3%,

meningkat 4,4% dibandingkan dengan pendekatan dasar, serta mengungguli algoritma mainstream lainnya seperti SSD dan seri YOLO.

Penelitian ini juga melibatkan eksperimen ablation untuk mengevaluasi kontribusi masing-masing modul yang ditambahkan, yang menunjukkan bahwa penambahan modul STC dan GAM secara signifikan meningkatkan akurasi dan recall deteksi target kecil. Meskipun demikian, penulis mencatat bahwa peningkatan jumlah parameter yang signifikan dapat mempengaruhi kecepatan deteksi, sehingga penelitian lebih lanjut diperlukan untuk menemukan keseimbangan antara akurasi dan efisiensi model. Dengan demikian, penelitian ini memberikan kontribusi penting dalam pengembangan algoritma deteksi target UAV yang lebih efektif, terutama dalam konteks deteksi target kecil.

Penelitian Chen et al. [19] memperkenalkan model DET-YOLO, sebuah inovasi dalam deteksi pesawat militer (MAR20) pada citra penginderaan jarak jauh (RSI), yang bertujuan untuk mengatasi tantangan rendahnya tingkat deteksi dan tingginya tingkat kesalahan deteksi pada data penginderaan jarak jauh yang kompleks. Model ini mengimplementasikan komponen ekstraksi fitur baru yang disebut C2f_DEF, yang menggantikan semua komponen C2f dalam YOLOv8n, sehingga meningkatkan kemampuan model dalam mengenali target kecil di tengah latar belakang yang rumit. Selain itu, DET-YOLO dirancang untuk menjadi lebih ringan dengan menyederhanakan struktur jaringan dalam rangka memenuhi kebutuhan penerapan model yang mudah dan cepat. Penelitian ini juga menyematkan modul DAT (Deformable Attention Transformer) untuk meningkatkan akurasi deteksi dalam lingkungan yang padat dan kompleks, serta

mengoptimalkan fungsi loss dengan menggunakan WIoU-v2 untuk mempercepat konvergensi model tanpa menambah jumlah parameter.

Hasil eksperimen menunjukkan bahwa DET-YOLO mencapai mean Average Precision (mAP) sebesar 94,7% pada dataset MAR20, yang merupakan peningkatan signifikan dibandingkan dengan model-model sebelumnya, dengan hanya 80 epoch pelatihan. Model ini juga menunjukkan kinerja yang baik dalam hal kecepatan deteksi, mencapai 68,97 FPS, yang memenuhi persyaratan deteksi waktu nyata yang diperlukan dalam konteks militer. Penelitian ini mengkonfirmasi bahwa DET-YOLO tidak hanya efektif dalam mendeteksi pesawat militer dalam citra penginderaan jarak jauh, tetapi juga memiliki potensi aplikasi yang luas dalam bidang pengawasan dan keamanan nasional, serta pengembangan teknologi penglihatan komputer dan pengenalan pola.

Dari berbagai penelitian di atas, peneliti akan berfokus pada pengembangan model deteksi target pesawat menggunakan YOLO11, yang merupakan perbaikan dari model-model sebelumnya seperti YOLOv3, YOLOv5, yang telah terbukti memiliki keterbatasan dalam mendeteksi objek kecil dalam citra penginderaan jarak jauh [17]. Penelitian ini menekankan pentingnya peningkatan akurasi dan pengurangan tingkat kesalahan deteksi, terutama dalam lingkungan yang kompleks dan latar belakang yang rumit, yang merupakan tantangan utama dalam deteksi pesawat militer [19]. Dengan mengimplementasikan parameter tuning dan modifikasi arsitektur dengan modul Global Attention Mechanism (GAM) usulan dari penelitian Wang et al. [18], penelitian ini diharapkan mampu meningkatkan ekstraksi fitur dan fusi informasi, yang berkontribusi pada peningkatan akurasi,

recall dan mAP (mean Average Precision). Dengan demikian, penelitian ini tidak hanya berkontribusi pada pengembangan algoritma deteksi yang lebih efisien dan akurat, tetapi juga relevan dalam konteks aplikasi nyata di bidang pengawasan dan keamanan nasional, yang sangat penting untuk memperkuat sistem pertahanan dan strategi militer.

Dalam menilai penelitian-penelitian sebelumnya terkait deteksi pesawat dalam citra penginderaan jarak jauh menggunakan model YOLO, terdapat sejumlah kelemahan yang dapat dijadikan dasar untuk kritik atau justifikasi. Bahwa penelitian yang akan dilakukan merupakan suatu bentuk penyempurnaan atau pengembangan dari penelitian-penelitian tersebut.

Pertama, beberapa penelitian seperti yang dilakukan oleh Chen et al. [19] dan Xu et al. [17] menunjukkan bahwa meskipun model-model yang dihasilkan (DET-YOLO dan CA-YOLO) berhasil meningkatkan akurasi dan kecepatan deteksi, mereka masih menghadapi tantangan terkait generalisasi model di lingkungan yang sangat bervariasi. Penelitian tersebut mungkin tidak sepenuhnya mengatasi isu-isu terkait dengan variasi ukuran target dan kompleksitas latar belakang, yang dapat mempengaruhi performa deteksi pada situasi nyata.

Kedua, meskipun penelitian Chen et al. [17] menggunakan fungsi loss yang lebih baik dalam YOLOv8, peningkatan akurasi yang dicapai tetap berada dalam batasan tertentu, mengindikasikan adanya ruang untuk eksplorasi lebih lanjut dalam optimasi arsitektur jaringan dan fungsionalitas loss yang lebih adaptif. Keterbatasan dalam mengekstraksi fitur dari objek kecil dalam lingkungan yang padat dan

kompleks juga menjadi perhatian, terutama ketika berhadapan dengan citra resolusi rendah atau noise pada citra SAR [16].

Selain itu, kebutuhan akan efisiensi dalam pengolahan data secara real-time menjadi semakin penting dalam konteks aplikasi militer dan pertahanan keamanan, dan meskipun penelitian-penelitian terdahulu seperti DET-YOLO telah menunjukkan kecepatan deteksi yang baik, peningkatan lebih lanjut dalam hal ini dapat dicapai dengan memanfaatkan teknik-teknik terbaru dalam optimasi model dan efisiensi komputasi.

Dengan demikian, penelitian ini bertujuan untuk mengatasi kelemahan-kelemahan tersebut dengan mengintegrasikan inovasi baru dalam model seperti arsitektur YOLO yang lebih mutakhir dan mekanisme perhatian yang lebih efisien, sehingga menghasilkan sistem deteksi pesawat yang tidak hanya lebih akurat tetapi juga lebih adaptif terhadap variasi kondisi yang ada. Penelitian ini akan fokus untuk menyempurnakan dan memperluas hasil yang telah dicapai oleh penelitian sebelumnya, mengarah pada sistem deteksi yang lebih tangguh dan relevan dalam dunia pertahanan dan keamanan.

2.2 Keaslian Penelitian

Tabel 2.1 Matriks literatur review dan posisi penelitian
Inovasi YOLO11 Untuk Deteksi Target Pesawat Militer Pada Citra Penginderaan Jarak Jauh

No	Judul Penelitian	Nama Peneliti, Tahun, Index	Dataset	Metode Penelitian	Hasil	Keunggulan dan Kelemahan	Perbandingan
1	Aircraft Detection in Airport Remote Sensing Images based on YOLOv5	Chengyun Ji, Li Zhang, Ji Li, 9th International Conference on Digital Home (ICDH), 2022	Rareplanes	Mengusulkan YOLOv5-OLS, yang membagi citra input berukuran besar menjadi bagian-bagian kecil yang tumpang tindih untuk dideteksi, guna mengurangi kehilangan informasi target kecil.	Akurasi keseluruhan meningkat dari 66.0% (YOLOv5 standar) menjadi 79.4% (YOLOv5-OLS).	Keunggulan: Secara drastis meningkatkan akurasi deteksi target kecil pada citra resolusi sangat tinggi. Kelemahan: Proses deteksi memerlukan langkah tambahan (pemisahan dan penggabungan).	YOLOv5-OLS (akurasi 79.4%) jauh mengungguli SSD (46.3%), R-CNN (53.2%), dan YOLOv5 standar (66.0%).
2	CA-YOLO: Model Optimization for Remote Sensing Image Object Detection	Lingyun Shen, Baihe Lang, Zhengxun Song, IEEE Access, 2023	RSOD	Mengusulkan CA-YOLO, perbaikan dari YOLOv5 yang menambahkan modal Coordinate Attention (CA) dan Spatial Pyramid Pooling-Fast (SPPF) dengan konstruksi tandem, serta optimasi loss function.	Peningkatan mAP@0.5 sebesar 4.8% dan mAP@0.5:0.95 sebesar 3.8% dibandingkan YOLOv5 asli pada dataset RSOD, dengan kecepatan inferensi rata-rata 125 fps.	Keunggulan: Efisiensi tinggi, akurasi deteksi multi-objek yang lebih baik, dan generalisasi yang kuat tanpa menambah kompleksitas parameter. Kelemahan: Tidak disebutkan secara eksplisit.	CA-YOLO (mAP@0.5 94.2%) lebih baik dari YOLOv3-tiny (84.5%), YOLOv4 (86.5%), YOLOv5s (89.4%), dan YOLOv8 (89.2%) pada dataset RSOD.

No	Judul Penelitian	Nama Peneliti, Tahun, Index	Dataset	Metode Penelitian	Hasil	Keunggulan dan Kelemahan	Perbandingan
3	YOLO-Class: Detection and Classification of Aircraft Targets in Satellite Remote Sensing Images Based on YOLO-Extract	Zhiguo Liu, Yuan Gao, Qianqian Du, IEEE Access, 2023	RarcPlanes dan DOTA.	Mengusulkan model YOLO-Class yang dibangun di atas YOLO-Extract. Menggunakan Representative Batch Normalization, aktivasi Mish, VariFocal loss, dan modul RepVGG untuk meningkatkan akurasi klasifikasi.	Akurasi deteksi (mAP) meningkat dari 0.608 (YOLO-Extract) menjadi 0.704 (YOLO-Class), dan FPS meningkat dari 36.16 menjadi 39.598.	Keunggulan: Meningkatkan akurasi klasifikasi dan deteksi secara bersamaan sambil tetap mempertahankan kecepatan. Kelemahan: Peningkatan akurasi masih memiliki ruang untuk perbaikan lebih lanjut.	YOLO-Class (mAP 0.704) secara signifikan mengungguli model YOLO lainnya seperti YOLOv4 (0.548), YOLOv5 (0.573), YOLOv7 (0.581), dan YOLOv8 (0.616).
4	YOLO-extract: Improved YOLOv5 for Aircraft Object Detection in Remote Sensing Images	Zhiguo Liu, Yuan Gao, Qianqian Du, Meng Chen, Wenqiang Lv, IEEE Access, 2023	DOTA (Dataset for Object Detection in Aerial Images)	Mengusulkan algoritma YOLO-extract yang memodifikasi YOLOv5 dengan beberapa perbaikan utama: 1. Optimasi Struktur: Menghapus feature layer dan prediction head yang kurang efisien dan mengintegrasikan feature extractor baru yang lebih kuat.	Dibandingkan dengan YOLOv5 standar: 1. mAP meningkat sebesar 8.1% (dari 0.878 menjadi 0.959). 2. Kecepatan deteksi (FPS) meningkat 3 kali lipat (dari ~12 FPS menjadi ~32 FPS). 3. Ukuran model lebih ringan:	Keunggulan: • Secara signifikan meningkatkan akurasi (mAP) dan kecepatan (FPS) secara bersamaan. • Model lebih ringan (parameter lebih sedikit), sehingga lebih efisien secara komputasi. • Sangat efektif untuk mendeteksi target pesawat yang kecil, padat, dan terhalang dalam	<ul style="list-style-type: none"> • Terhadap YOLOv5 (baseline): YOLO-extract secara signifikan lebih unggul di semua metrik: mAP (0.959 vs 0.878), Recall (0.925 vs 0.794), FPS (32.2 vs 11.9), dan memiliki parameter yang lebih sedikit (35.59M vs 46.11M). • Terhadap Model Lain (YOLOv3, YOLOv4, YOLOv7, SSD, Faster RCNN): Di lingkungan pengujian yang sama, YOLO-extract menunjukkan nilai mAP dan FPS tertinggi, membuktikan keseimbangan

No	Judul Penelitian	Nama Peneliti, Tahun, Index	Dataset	Metode Penelitian	Hasil	Keunggulan dan Kelemahan	Perbandingan
				<p>2. Integrasi Coordinate Attention (CA): Memasukkan modul atensi untuk meningkatkan fokus pada target pesawat yang padat dan dalam latar belakang yang kompleks.</p> <p>3. Dilated Convolution & Residual: Menggunakan kombinasi mixed dilated convolution dan struktur residual yang didesain ulang untuk meningkatkan ekstraksi fitur pada target skala kecil.</p> <p>4. Loss Function Baru: Merancang Focal-n EIoU Loss untuk menggantikan CloU Loss, guna</p>	<p>Jumlah parameter berkurang 10.526 juta dan GFLOPs berkurang 45.34. Konvergensi lebih cepat dan nilai loss akhir lebih rendah.</p>	<p>latar belakang yang kompleks.</p> <ul style="list-style-type: none"> • Lebih unggul dalam mengatasi gangguan seperti kondisi cuaca (awan/kabut). <p>Kelemahan:</p> <ul style="list-style-type: none"> • Penelitian ini mengakui keterbatasan dalam mendeteksi dan membedakan berbagai tipe pesawat karena kurangnya dataset yang beragam untuk klasifikasi tipe pesawat. 	<p>terbaik antara akurasi dan kecepatan.</p>

No	Judul Penelitian	Nama Peneliti, Tahun, Index	Dataset	Metode Penelitian	Hasil	Keunggulan dan Kelemahan	Perbandingan
				mempercepat konvergensi dan meningkatkan akurasi regresi bounding box.			
5	EST-YOLOv5s: SAR Image Aircraft Target Detection Model Based on Improved YOLOv5s	Min Huang et al., IEEE Access, 2023	SADD (SAR Aircraft Detection Dataset)	Model EST-YOLOv5s yang diusulkan mengintegrasikan mekanisme Efficient Channel Attention (ECA), Swin Transformer Block, dan TSCODE head ke dalam YOLOv5s.	Model yang diusulkan mencapai recall 94.2%, presisi 97.3%, dan mAP@50 97.8%, masing-masing 2.3%, 1.7%, dan 1.7% lebih tinggi dari YOLOv5s dasar.	Keunggulan: Performa tinggi untuk deteksi pesawat pada citra SAR, kemampuan anti-interferensi yang kuat, dan memenuhi persyaratan real-time. Kelemahan: Tidak disebutkan secara eksplisit.	Mengungguli model dasar YOLOv5s secara signifikan pada semua metrik evaluasi pada dataset SADD.
6	CNTR-YOLO: Improved YOLOv5 Based on ConvNext and Transformer for Aircraft Detection in Remote Sensing Images	Fengyun Zhou et al., Electronics (MDPI), 2023	MAR20 dan DOTA.	Mengusulkan CNTR-YOLO, perbaikan dari YOLOv5 dengan mengintegrasikan DenseNet, modul atensi CBAM, dan modul C3CNTR (ConvNext+ Transformer).	Peningkatan mAP sebesar 3.3% pada dataset MAR20 (mencapai 70.1%) dan 2.5% pada dataset DOTA (mencapai 63.7%) dibandingkan YOLOv5L.	Keunggulan: Pemanfaatan informasi lokal dan global yang lebih baik, menghasilkan akurasi lebih tinggi. Kelemahan: Waktu inferensi sedikit lebih lambat dari YOLOv5L.	Mengungguli Faster R-CNN, YOLOv4, dan berbagai varian YOLOv5 dalam hal presisi, recall, dan mAP pada dataset MAR20 dan DOTA.
7	Aircraft Target Detection from Remote Sensing	Dan Zhong et al.,	Dataset kustom berisi 2.007	Mengusulkan YOLOX-DD, modifikasi dari	Akurasi deteksi tertinggi di bawah kondisi	Keunggulan: Sangat tangguh terhadap berbagai	YOLOX-DD (AP 94.8%) mengungguli YOLOv3, Faster R-CNN, YOLOF, dan

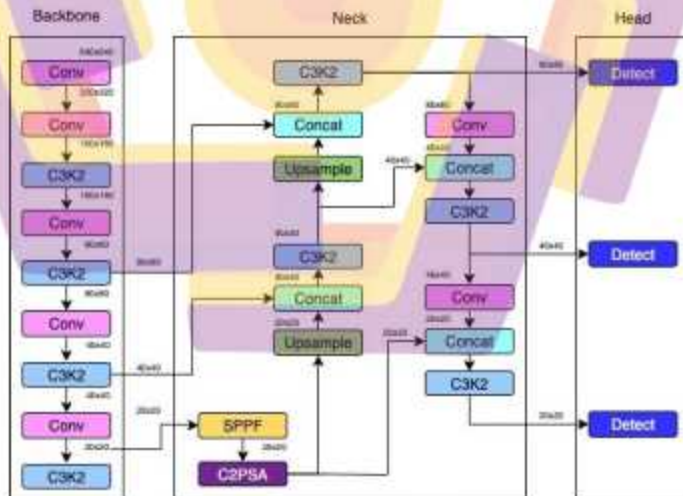
No	Judul Penelitian	Nama Peneliti, Tahun, Index	Dataset	Metode Penelitian	Hasil	Keunggulan dan Kelemahan	Perbandingan
	Images under Complex Meteorological Conditions	Sustainability (MDPI), 2023	gambar pesawat dalam berbagai kondisi cuaca.	YOLOX yang menggunakan depth-separable convolution dan dilated convolution untuk meningkatkan deteksi dalam cuaca buruk.	cuaa kompleks (malam, berdebu, hujan, dll). Parameter model berkurang 2.4 juta, dan akurasi meningkat hingga 9.5% pada skenario tertentu.	kondisi cuaca, model lebih ringan dan efisien. Kelemahan: Fokus utama pada kondisi cuaca ekstrem, performa pada kondisi normal tidak dijelaskan secara detail.	SSD pada dataset uji, terutama dalam skenario cuaca buruk.
8	Comparative analysis of YOLO algorithms for aircraft detection in remote sensing images	Touati Adli et al., Internasional Conference OTEH, 2024	MAR20	Studi komparatif antara YOLOv5, YOLOv7, dan YOLOv8 untuk deteksi pesawat militer.	YOLOv7 menunjukkan performa terbaik dengan mAP 67.7% dan mAP0.5 90.3%.	Keunggulan (YOLOv7): Akurasi dan presisi superior. Kelemahan (YOLOv7): Waktu inferensi sedikit lebih lambat dari YOLOv5m.	YOLOv7 (mAP 67.7%) lebih unggul dari YOLOv5m (64.5%) dan YOLOv8m (65.2%) pada dataset MAR20.
9	Comparison of YOLOv8 Models for Aircraft Detection in Airport Apron Using Digital Image Processing	Nitipoom Wiangkam & Somchat Jirwibhakorn, Engineering & Technology Horizon, 2024	Dataset kustom berisi 1.000 gambar apron bandara dari Google Earth.	Analisis perbandingan berbagai versi model YOLOv8 (n, s, m, nl, nx) untuk deteksi pesawat.	YOLOv8nx menunjukkan kinerja tertinggi (F1-score 0.83), sementara YOLOv8n adalah yang tercepat (0.95 ms).	Keunggulan: Memberikan panduan pemilihan model YOLOv8 sesuai kebutuhan (akurasi vs kecepatan). Kelemahan: Hanya membandingkan varian dalam satu	YOLOv8nx paling akurat namun paling lambat. YOLOv8n paling cepat namun paling tidak akurat. Model lain (s, m, nl) berada di antaranya.

No	Judul Penelitian	Nama Peneliti, Tahun, Index	Dataset	Metode Penelitian	Hasil	Keunggulan dan Kelemahan	Perbandingan
						keluarga model (YOLOv8).	
10	DET-YOLO: An Innovative High-Performance Model for Detecting Military Aircraft in Remote Sensing Images	Xiaoxin Chen, Hui Jiang, Hongxin Zheng, Jiankun Yang, Riqiang Liang, Dan Xiang, Hao Cheng, Zhansi Jiang, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2024	MAR20, NWPU VHR-10, dan NEU-DET.	Mengusulkan DET-YOLO, model deteksi inovatif berbasis YOLOv8n. Mengganti modul C2f dengan C2f DEF, melakukan pruning model, menyematkan modul DAT, dan mengoptimalkan loss function.	Pada dataset MAR20, mencapai mAP0.5 tertinggi (94.7%) dengan hanya 80 epoch pelatihan, sambil tetap ringan dan cepat (68.97 FPS).	Keunggulan: Sangat ringan (parameter dan ukuran model terkecil), cepat, dan akurat, terutama untuk pesawat militer. Kelemahan: Masih ada celah jika dibandingkan dengan model ultra-ringan lainnya dalam hal jumlah parameter.	DET-YOLO (mAP0.5 94.7%) mengungguli semua varian YOLOv5, YOLOv6, dan YOLOv8 yang diuji pada dataset MAR20.

2.3 Landasan Teori

2.3.1 Arsitektur YOLO11

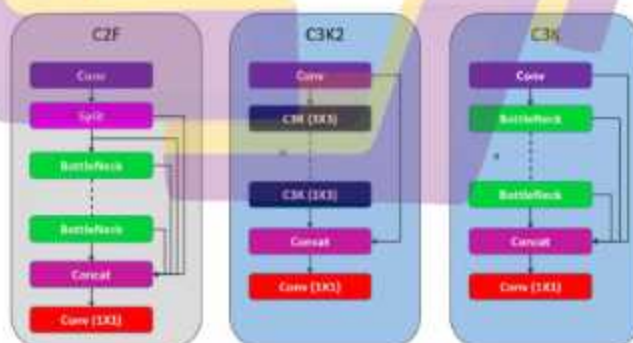
Dari artikel Khanam [20] dan Alif [11] dijelaskan tentang analisis arsitektur dari YOLO11, versi terbaru dari model deteksi objek dalam seri YOLO (You Only Look Once). YOLO11 memperkenalkan beberapa inovasi arsitektural, termasuk blok C3k2 (Cross Stage Partial 3 konvolusi dengan 2 kernel 3x3), SPPF (Spatial Pyramid Pooling Fast), dan C2PSA (Convolutional block with 2 Parallel Spatial Attention), yang meningkatkan kemampuan ekstraksi fitur dan efisiensi komputasi model. Model ini tidak hanya unggul dalam deteksi objek, tetapi juga dalam tugas-tugas lain seperti segmentasi instance, estimasi pose, dan deteksi objek terorientasi.



Gambar 2.1 Arsitektur YOLO11

YOLO11 menunjukkan peningkatan signifikan dalam mean Average Precision (mAP) dan efisiensi komputasi dibandingkan pendahulunya, dengan fokus pada keseimbangan antara jumlah parameter dan akurasi. Selain itu, YOLO11 dirancang untuk berbagai ukuran model, dari nano hingga ekstra besar, sehingga dapat digunakan di berbagai aplikasi, mulai dari perangkat edge hingga lingkungan komputasi berkinerja tinggi [20]. Arsitektur YOLO11 ditunjukkan pada Gambar 2.1.

YOLO11 merupakan pengembangan dari YOLOv8, arsitektur model visi komputer yang dikembangkan oleh Ultralytics. Perbandingan arsitektur antara YOLOv8 dan YOLO11 terlihat jelas dalam Gambar 2.2 yang digunakan untuk ekstraksi fitur. Pada YOLOv8, digunakan C2F (Cross Stage Partial), yang menggabungkan operasi split dan concatenation dengan menggunakan beberapa bottleneck untuk meningkatkan efisiensi pengolahan fitur. Ini bertujuan untuk memaksimalkan aliran informasi antara layer-layer dalam jaringan.



Gambar 2.2 Perbandingan modul C2F pada YOLOv8 dengan modul C3K2 pada YOLO11

Sementara itu, YOLO11 memperkenalkan inovasi dengan penggunaan C3K2 (Cross Stage dengan kernel 3×3), yang memiliki keunggulan dalam menyaring dan mengolah informasi dengan lebih efektif. C3K2 mengkombinasikan beberapa blok C3K dan menghilangkan operasi split, yang memungkinkan penanganan data lebih baik dan menghasilkan output yang lebih informatif. Selain itu, arsitektur YOLO11 juga mengintegrasikan modul Spatial Pyramid Pooling Fast (SPPF), yang tidak hanya meningkatkan efisiensi tetapi juga akurasi deteksi objek. Modul SPPF ditunjukkan pada Gambar 3.

Secara umum, peralihan dari YOLOv8 ke YOLO11 menunjukkan upaya untuk meningkatkan performa model dengan lebih banyak fokus pada efisiensi dan akurasi melalui inovasi struktur yang lebih kompleks. Hal ini membuat YOLO11 lebih unggul dalam berbagai tugas deteksi objek dan aplikasi lainnya dibandingkan dengan YOLOv8.

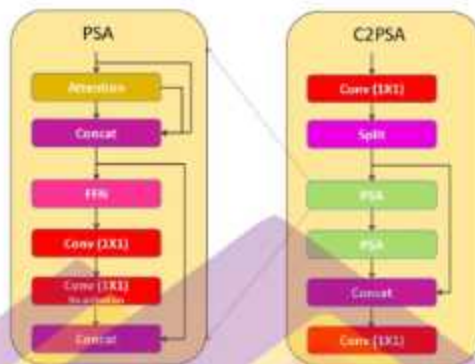


Gambar 2.3 Modul Spatial Pyramid Pooling Fast (SPPF)

Gambar 2.3 menjelaskan konsep Spatial Pyramid Pooling Fast (SPPF) dalam pengolahan citra. Di sisi kiri, terdapat sebuah matriks yang menunjukkan sejumlah nilai numerik yang adalah hasil dari operasi konvolusi. Dari matriks itu

menunjukkan pemrosesan dengan menerapkan Max Pooling, yang merupakan teknik untuk mengurangi dimensi data sambil mempertahankan fitur pentingnya. Matriks 4x4 dibagi menjadi empat kuadran berwarna (merah, hijau, ungu, dan kuning). Masing-masing kuadran ini diambil untuk menemukan nilai maksimum dari setiap subset data. Setelah menerapkan Max Pooling, hasilnya ditampilkan di sebelah kanan, di mana setiap elemen merupakan nilai maksimum yang diambil dari masing-masing kuadran. Misalnya, dari kuadran merah, nilai maksimum adalah 20, dari kuadran ungu adalah 30, dari kuadran hijau adalah 112 dan dari kuadran kuning adalah 37. Proses ini membantu menyederhanakan informasi dengan menekan ukuran matriks, membuatnya lebih efisien untuk analisis selanjutnya.

SPPF berfungsi untuk mengolah matriks ini dengan menggunakan beberapa lapisan, yang diawali dengan konvolusi 1x1 untuk ekstraksi fitur awal. Kemudian, diteruskan ke proses MaxPool2d yang diterapkan dengan berbagai ukuran, di mana tiap lapisan mengabaikan dimensi spasial yang berbeda, memungkinkan model untuk menangkap berbagai skala fitur. Hasil dari beberapa lapisan MaxPool2d ini akan digabungkan (concatenation) sebelum diteruskan kembali ke konvolusi 1x1 untuk menghasilkan representasi akhir. Proses ini mempercepat pengolahan citra sambil mempertahankan informasi penting, meningkatkan kemampuan model dalam mendeteksi objek.



Gambar 2.4 Modul C2 Position Sensitive Attention Block (C2PSA)

Gambar 2.4 menggambarkan dua blok arsitektur dalam pengolahan citra yaitu Position Sensitive Attention (PSA) dan Convolutional Position Sensitive Attention (C2PSA). Proses pada blok PSA dimulai dengan penerapan mekanisme Attention, yang bertujuan untuk menonjolkan informasi penting dari data input. Setelah itu, hasil dari mekanisme Attention di concat dengan informasi tambahan, dilanjutkan dengan feedforward network (FFN), dan diakhiri dengan dua tahap konvolusi 1×1 . Blok ini bertujuan untuk mengoptimalkan representasi fitur sebelum menggabungkannya kembali sebelum lapisan terakhir.

Sementara itu, pada blok C2PSA, proses dimulai dengan konvolusi 1×1 untuk menyiapkan data. Selanjutnya, data dibagi (split) untuk memproses informasi secara paralel, yang kemudian melalui dua tahap PSA berturut-turut, masing-masing bertujuan untuk memperkuat perhatian pada informasi yang relevan. Hasil dari kedua PSA ini kemudian concatenated dan dilakukan konvolusi 1×1 sebagai langkah akhir. Pendekatan C2PSA ini memberikan keunggulan dalam efisiensi pemrosesan dengan memanfaatkan dua attention (PSA), yang dapat meningkatkan

kemampuan model dalam mendeteksi fitur dan mengurangi kemungkinan kehilangan informasi penting.

2.3.2 Hyperparameter Tuning

Hyperparameter tuning merupakan proses yang krusial dalam pengoptimalan model machine learning, termasuk arsitektur YOLO (You Only Look Once), yang sering digunakan untuk tugas deteksi objek. Proses ini melibatkan penyesuaian berbagai parameter untuk meningkatkan kinerja model, seperti akurasi, presisi, dan recall, di berbagai aplikasi.

Hyperparameter tuning berdampak signifikan pada kinerja model YOLO di berbagai domain. Sebagai contoh, dalam konteks deteksi kesalahan printer 3D, pengoptimalan ukuran input dan ukuran batch menghasilkan peningkatan kinerja deteksi, yang mencapai nilai mean Average Precision (mAP) rata-rata sebesar 89,7% dengan ukuran batch 16 dan ukuran input gambar 640 [21]. Dalam klasifikasi multi-tanaman menggunakan gambar UAV, penyetelan ukuran batch dari 10 hingga 60 menunjukkan peningkatan dalam kinerja pengklasifikasi, di mana ukuran batch 60 memberikan keseimbangan terbaik antara akurasi, presisi, dan recall [22].

Penyesuaian ukuran batch dan learning rate merupakan strategi umum dalam hyperparameter tuning. Dalam studi mengenai deteksi kanker payudara, berbagai metode pengoptimalan, termasuk Adaptive Moment Estimation (Adam), Stochastic Gradient Descent (SGD), dan Root Mean Square Propagation (RMSprop), dieksplorasi. Hasil menunjukkan bahwa SGD memberikan kinerja terbaik dalam hal waktu pelatihan dan nilai mAP [23]. Selain itu, dalam bidang

deteksi drone, eksperimen iteratif pada tingkat pembelajaran dan pengoptimalan ukuran batch dilakukan untuk meningkatkan kinerja model tersebut [24].

Augmentasi data merupakan teknik lain yang sering digunakan bersamaan dengan hyperparameter tuning untuk meningkatkan ketahanan model. Dalam konteks deteksi drone, metode augmentasi seperti rotasi, penskalaan, dan penyesuaian warna diterapkan, di mana rotasi menunjukkan peningkatan kinerja yang tertinggi [24]. Begitu juga dalam deteksi bahaya jalan raya, strategi augmentasi telah digunakan untuk mengoptimalkan kinerja model dalam berbagai kondisi [25].

Hyperparameter tuning telah diterapkan dengan sukses dalam berbagai aplikasi, yang menunjukkan fleksibilitas dan efektivitas YOLO. Dalam deteksi jalur pipa bawah air, penyetelan ukuran batch, epochs, dan learning rate meningkatkan akurasi deteksi dalam kondisi yang menantang [26]. Selain itu, dalam deteksi bahaya jalan raya, penyesuaian learning rate dan ukuran batch meningkatkan kemampuan model untuk mendeteksi berbagai bahaya secara efisien [25]. Dalam sistem deteksi senjata, pengoptimalan hyperparameter secara iteratif juga terbukti efektif dalam meningkatkan presisi dan recall, menggambarkan potensi YOLO dalam aplikasi keamanan [27].

Hyperparameter tuning adalah aspek penting dalam memaksimalkan kinerja model YOLO di berbagai domain. Dengan menyesuaikan parameter seperti ukuran batch, learning rate, epoch dan menggunakan teknik augmentasi data secara cermat, peneliti dapat meningkatkan akurasi, presisi, dan recall model secara signifikan. Ini menjadikan YOLO sebagai alat yang ampuh untuk berbagai tugas deteksi objek.

2.3.3 Optimizer

Berbagai algoritma optimasi seperti Adam, AdamW, Nadam, RAdam, Adamax, dan SGD telah dievaluasi secara luas dalam berbagai aplikasi deep learning, khususnya pada arsitektur Convolutional Neural Network (CNN) dan model-model lain seperti YOLOv8 dan Transformer. Beberapa jenis optimizer pada YOLO11:

1. Adaptive Moment Estimation (Adam)

Optimizer Adam (Adaptive Moment Estimation) bekerja dengan menghitung dua momen pertama dari gradien: rata-rata bergerak gradien (momentum pertama) dan rata-rata bergerak kuadrat gradien (momentum kedua). Pada tiap iterasi, Adam memperbarui estimasi ini dengan faktor peluruhan (decay rates) (β_1) dan (β_2), serta melakukan koreksi bias agar estimasi ini akurat terutama pada awal pelatihan. Setelah itu, parameter model diperbarui menggunakan learning rate yang disesuaikan secara adaptif untuk setiap parameter berdasarkan estimasi momen tersebut, yang memungkinkan pembaruan lebih stabil dan efisien dibandingkan metode konvensional seperti SGD biasa [28].

Keunggulan Adam terletak pada kemampuannya untuk mengadaptasi learning rate secara otomatis, membuatnya efektif untuk masalah dengan data besar dan parameter yang kompleks serta tahan terhadap gradien jarang (sparse gradients). Namun, Adam memiliki kelemahan seperti sensitivitas terhadap pilihan hyperparameter dan berpotensi menyebabkan konvergensi dini ke solusi suboptimal, serta

kadang kurang optimal untuk generalisasi dibandingkan optimizer lain dalam beberapa kasus [29]. Oleh karena itu, meski sangat populer dan banyak dipakai, penggunaan Adam tetap memerlukan penyesuaian yang hati-hati untuk mencapai hasil terbaik.

2. Nesterov-accelerated Adaptive Moment Estimation (NAdam)

Optimizer Nesterov-accelerated Adaptive Moment Estimation (NAdam) adalah pengembangan dari Adam yang menggabungkan keuntungan momentum Nesterov dengan mekanisme adaptif Adam. Cara kerjanya mirip dengan Adam, yaitu dengan menghitung rata-rata momentum pertama (gradien) dan momentum kedua (kuadrat gradien) beserta koreksi biasnya, tetapi NAdam menambahkan langkah "lookahead" dari momentum Nesterov yang melakukan perhitungan gradien tidak hanya berdasarkan posisi saat ini, melainkan juga memperkirakan posisi parameter di masa depan. Pendekatan ini menghasilkan update parameter yang lebih responsif dan akurat, memberikan konvergensi yang lebih cepat dan stabil terutama pada fungsi tujuan yang tidak konveks atau kompleks [30].

Keunggulan NAdam terletak pada kemampuannya untuk menangkap informasi momentum secara lebih efektif dibanding Adam biasa, sehingga sering kali mempercepat pelatihan dan meningkatkan performa model. Namun, kelemahannya termasuk peningkatan kompleksitas perhitungan yang sedikit lebih tinggi dan potensi sensitivitas terhadap pemilihan hyperparameter yang sama seperti pada Adam. Tidak semua permasalahan menunjukkan peningkatan signifikan dengan NAdam,

sehingga pemilihan optimizer tetap bergantung pada konteks dan karakteristik data.

3. Rectified Adam (RAdam)

Optimizer Rectified Adam (RAdam) merupakan modifikasi dari Adam yang dirancang untuk mengatasi ketidakstabilan saat tahap awal pelatihan dengan mengoreksi varians adaptif learning rate yang belum stabil. Cara kerjanya mempertahankan mekanisme dasar Adam dengan menggunakan rata-rata momentum pertama dan kedua serta koreksi bias, namun RAdam menambahkan proses "rectification" pada learning rate adaptifnya berdasarkan estimasi varians dari momen kedua yang beradaptasi seiring waktu. Dengan demikian, pada awal pelatihan ketika varians masih tidak stabil, RAdam secara otomatis menurunkan learning rate adaptif agar pembaruan parameter lebih stabil dan menghindari lonjakan besar yang dapat memperlambat konvergensi atau menyebabkan divergensi [31].

Keunggulan RAdam adalah kemampuannya untuk memberikan pelatihan yang lebih stabil dan konvergensi yang lebih cepat tanpa memerlukan proses warm-up atau penyesuaian learning rate manual seperti pada Adam biasa. Hal ini membuat RAdam sangat cocok untuk berbagai tugas pembelajaran mesin dengan konfigurasi hyperparameter yang lebih sederhana.

Namun, kekurangannya adalah tambahan perhitungan untuk rectification yang sedikit meningkatkan kompleksitas komputasi, dan dalam

beberapa kasus spesifik, peningkatan performa dibanding Adam mungkin tidak signifikan, sehingga pemilihan optimizer tetap harus disesuaikan dengan problem dan data yang dihadapi.

RAdam, yang memperbaiki masalah adaptasi learning rate pada awal pelatihan Adam, juga menjadi salah satu opsi yang dipertimbangkan dalam deteksi landmark, meskipun tidak selalu menjadi yang terbaik [32].

4. Adam with Weight Decay (AdamW)

Optimizer Adam with Weight Decay (AdamW) merupakan varian dari Adam yang secara eksplisit memisahkan regularisasi weight decay dari mekanisme update gradien, sehingga memperbaiki cara penanganan regularisasi yang sebelumnya terintegrasi dalam update Adam standar. Cara kerjanya mirip dengan Adam dalam menghitung rata-rata momentum pertama dan kedua gradien serta koreksi biasnya, tetapi AdamW melakukan penalti weight decay langsung pada parameter model secara terpisah setelah melakukan update gradien, bukan menggabungkannya sebagai bagian dari gradien itu sendiri. Pendekatan ini menghindari distorsi update adaptif learning rate yang bisa terjadi pada Adam standar ketika weight decay diintegrasikan ke dalam gradien, sehingga menghasilkan regularisasi yang lebih murni dan stabil [33].

Keunggulan utama AdamW adalah kemampuannya memberikan regularisasi yang lebih efektif dan meningkatkan generalisasi model, khususnya pada jaringan saraf dalam yang sangat dalam dan kompleks. Selain itu, AdamW mengurangi kebutuhan untuk tuning hyperparameter

secara ekstensif dibanding Adam dengan weight decay konvensional. Kelemahannya adalah kompleksitas tambahan dalam implementasi dibanding Adam biasa, meskipun secara komputasi tidak signifikan, dan dalam beberapa kasus spesifik, peningkatan performanya bisa kurang terasa jika weight decay tidak terlalu berperan penting.

5. Adamax

Optimizer Adamax adalah varian dari Adam yang menggunakan norma maksimum (infinity norm) pada estimasi momen kedua daripada norma kuadrat (L2 norm) seperti pada Adam, sehingga membuatnya lebih stabil dan tahan terhadap gradien yang sangat besar. Cara kerjanya masih mengikuti prinsip dasar Adam, yaitu menghitung estimasi momentum pertama (rata-rata gradien) dan momentum kedua, tetapi untuk momentum kedua, Adamax menggunakan nilai maksimum dari absolut gradien yang pernah muncul, sehingga update parameter didasarkan pada estimasi ekstrem (maksimum) dari variasi gradien, bukan rata-rata kuadratnya [28].

Keunggulan Adamax terletak pada kemampuannya untuk bekerja lebih baik dengan gradien sangat besar atau tidak stabil, yang terkadang menyebabkan Adam standar menjadi kurang stabil atau bahkan divergen. Selain itu, Adamax lebih tangguh terhadap problem dengan gradien sparse atau noise tinggi. Namun, kekurangannya adalah Adamax cenderung menggunakan langkah update yang lebih konservatif sehingga potensi konvergensi bisa lebih lambat dibanding Adam. Selain itu, Adamax kurang

populer dan kurang banyak dievaluasi secara luas dibanding Adam, sehingga terkadang kurang pilihan utama untuk aplikasi umum.

Adamax, sebagai ekstensi Adam berbasis infinity norm, juga konsisten menunjukkan performa tinggi pada berbagai tugas klasifikasi gambar dan diagnosis kanker serviks, sering kali menjadi pilihan utama untuk arsitektur CNN seperti VGG-16 dan ResNet [34].

6. Stochastic Gradient Descent (SGD)

Optimizer Stochastic Gradient Descent (SGD) adalah metode optimasi klasik yang secara iteratif memperbarui parameter model dengan menggunakan estimasi gradien dari satu atau beberapa contoh data secara acak pada setiap langkah. Cara kerjanya sederhana: pada tiap iterasi, SGD menghitung gradien fungsi loss terhadap parameter menggunakan subset data (batch kecil atau bahkan satu sampel), kemudian mengurangi parameter tersebut dengan proporsi learning rate dikalikan gradien tersebut. Pendekatan ini memungkinkan SGD untuk melakukan update secara cepat dan efisien, serta memperkenalkan variasi yang dapat membantu keluar dari minimum lokal yang buruk [35].

Keunggulan utama SGD adalah kesederhanaannya, efisiensi komputasi, dan kemampuan untuk bekerja dengan data sangat besar karena hanya memproses sebagian kecil data di tiap iterasi, sehingga sangat scalable. Namun, kekurangannya termasuk sensitivitas terhadap pemilihan learning rate dan konvergensi yang kadang lambat atau tidak stabil karena fluktuasi gradien akibat sampel acak. Selain itu, SGD tanpa modifikasi

momentum bisa tersangkut di minimum lokal dan memerlukan penjadwalan learning rate yang hati-hati untuk mencapai performa optimal.

SGD, sebagai optimizer klasik, tetap relevan dan sering digunakan sebagai baseline. Meskipun umumnya membutuhkan lebih banyak iterasi dan tuning learning rate, SGD dengan momentum dapat menjadi pilihan terbaik ketika tuning hyperparameter terbatas, dan performanya cukup kompetitif pada berbagai tugas [32][31][36]. Studi lain menegaskan bahwa perbedaan performa antara optimizer adaptif seperti Adam, AdamW, Adamax, dan Nadam cenderung kecil jika hyperparameter di-tuning dengan baik, sehingga pemilihan optimizer dapat disesuaikan dengan kebutuhan dan kemudahan tuning learning rate [36][37].

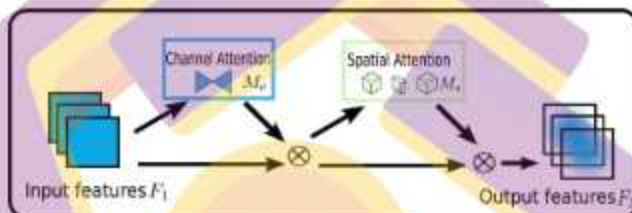
Dalam beberapa studi, Adam menunjukkan performa yang sangat baik dalam deteksi landmark pada citra x-ray lutut dengan nilai mAP tertinggi dibandingkan optimizer lain, meskipun Adamax unggul dalam presisi dan recall, serta AdamW menawarkan waktu pelatihan tercepat [32]. Namun, pada tugas pengukuran suspended solids menggunakan AlexNet, Adam, AdamW, dan Nadam justru menunjukkan performa terburuk dengan akurasi rendah, sedangkan Adamax dan SGD termasuk yang terbaik [38].

Secara umum, pemilihan optimizer sangat bergantung pada karakteristik data, arsitektur model, dan kebutuhan spesifik tugas. Adam, Adamax, dan Nadam sering menjadi pilihan utama untuk tugas-tugas klasifikasi dan deteksi, sementara SGD tetap menjadi opsi handal dengan tuning yang tepat. AdamW dan RAdam

menawarkan keunggulan dalam efisiensi waktu dan stabilitas, namun hasil akhirnya sangat dipengaruhi oleh pengaturan hyperparameter dan aplikasi.

2.3.4 Global Attention Mechanism (GAM) Module

Global Attention Mechanism diperkenalkan oleh Liu et al. [39] dalam artikel berjudul Global Attention Mechanism: Retain Information to Enhance Channel-Spatial Interactions.



Gambar 2.5 Global Attention Mechanism

Arsitektur GAM ditunjukkan pada Gambar 2.5. GAM dirancang untuk meningkatkan interaksi lintas dimensi dalam jaringan saraf dengan mempertahankan informasi dari aspek Channel Attention dan Spatial Attention. GAM menggunakan submodul channel attention yang memanfaatkan permutasi 3D dan multilayer perceptron (MLP) untuk memperkuat spatial attention. Dalam submodul spatial attention, dua lapisan konvolusi digunakan untuk menggabungkan informasi spasial tanpa menggunakan pooling, yang dapat mengurangi informasi penting.

$$F1 \in \mathbb{R}^{C \times H \times W} \quad (2.1)$$

$$F2 = M_c(F1) \odot F1 \quad (2.2)$$

$$F3 = M_s(F2) \odot F2 \quad (2.3)$$

Persamaan untuk Global Attention Mechanism (GAM) dapat dijelaskan sebagai berikut. Diberikan peta fitur input Persamaan (2.1), di mana C adalah jumlah saluran, H adalah tinggi, dan W adalah lebar, proses GAM terdiri dari dua langkah utama:

1. Channel Attention

Pada Persamaan (2.2), M_c adalah peta perhatian saluran yang dihasilkan dari pemrosesan F_1 menggunakan permutasi 3D dan multilayer perceptron (MLP), dan \odot menunjukkan perkalian elemen-wise.

2. Spatial Attention

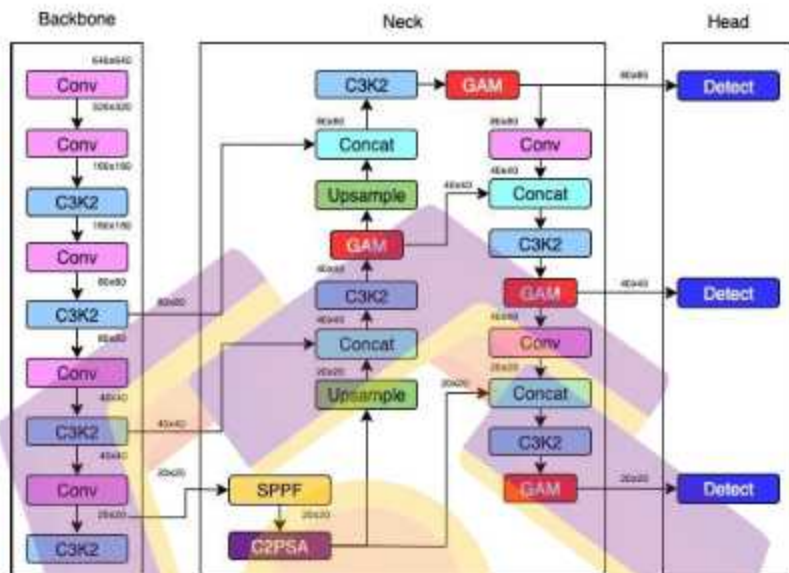
Di Persamaan (2.3), M_s adalah peta perhatian spasial yang dihasilkan dari pemrosesan F_2 menggunakan dua lapisan konvolusi untuk menggabungkan informasi spasial.

Dengan demikian, output akhir F_3 adalah hasil dari penerapan Channel Attention dan Spatial Attention secara berurutan, yang meningkatkan interaksi lintas dimensi antara Channel Attention dan Spatial Attention.

Dari hasil penelitian Liu et al. [39], evaluasi GAM pada dataset CIFAR-100 dan ImageNet-1K menunjukkan bahwa metode ini secara konsisten mengungguli attention mechanism lainnya, seperti Squeeze-and-Excitation Networks (SENet), Bottleneck Attention Module (BAM), dan Convolutional Block Attention Module (CBAM), baik pada arsitektur ResNet maupun MobileNet. Hasilnya menunjukkan bahwa GAM tidak hanya meningkatkan akurasi klasifikasi tetapi juga efisiensi parameter, menjadikannya solusi yang efektif untuk tugas pengenalan gambar.

2.3.5 Modifikasi YOLO11 dengan Penambahan Modul GAM

Penerapan Global Attention Mechanism (GAM) dalam arsitektur YOLO merupakan inovasi penting yang bertujuan untuk meningkatkan akurasi deteksi objek, khususnya pada target kecil yang sering kali sulit dikenali dalam citra yang diambil oleh Satelit atau UAV. Pada eksperimen Wang et al. [18], GAM diintegrasikan ke dalam modul Backbone YOLOv8 untuk menangkap informasi fitur yang krusial dalam berbagai dimensi, sehingga mengurangi kesulitan dalam mendeteksi objek kecil yang terpengaruh oleh faktor seperti blur akibat gerakan cepat dan jarak pengambilan gambar yang jauh. GAM melibatkan dua submodul utama: perhatian saluran (channel attention) dan perhatian spasial (spatial attention), yang bekerja secara berurutan untuk memperkuat representasi fitur. Dengan cara ini, GAM tidak hanya mengurangi kehilangan informasi penting selama proses pengambilan sampel, tetapi juga meningkatkan akurasi deteksi objek, terutama dalam konteks pengambilan gambar oleh UAV. Hasil eksperimen menunjukkan bahwa penambahan GAM tidak hanya meningkatkan nilai mean Average Precision (mAP) sebesar 0,6% dan akurasi sebesar 2%, tetapi juga memperbaiki kemampuan model dalam mengenali target kecil secara signifikan. Dengan demikian, penerapan GAM dalam YOLOv8 tidak hanya meningkatkan kinerja deteksi tetapi juga memberikan kontribusi terhadap pengembangan algoritma deteksi objek yang lebih efisien dan akurat dalam konteks pengawasan dan pemantauan berbasis UAV.



Gambar 2.6 Letak Modul GAM pada Arsitektur YOLO11

Gambar 2.6 menunjukkan empat modul GAM berada dalam struktur Neck YOLO11 yang diimplementasikan dalam penelitian ini. Keempat modul GAM tersebut ditempatkan setelah modul C3K2 di beberapa tahap aliran data. Penempatan 4 modul GAM dalam Neck berbeda dengan yang diterapkan pada penelitian sebelumnya. Modul ini berfungsi untuk menekankan fitur global yang relevan dengan menyesuaikan bobot channel dan spasial, sehingga memperbaiki kemampuan deteksi objek kecil atau kompleks.

Pada struktur Neck, GAM diintegrasikan setelah operasi Upsample, Concat dan C3K2, memungkinkan jaringan untuk mempertahankan informasi kontekstual dari backbone sambil memfilter noise. Setelah C3K2 pada resolusi 40x40 atau 20x20, GAM membantu menyempurnakan fitur sebelum diteruskan ke Detect

Head. Kombinasi ini diharapkan meningkatkan akurasi dan presisi tanpa mengorbankan kecepatan, karena GAM dirancang efisien secara komputasi.

Dengan demikian, penambahan empat modul GAM di Neck menjadikan YOLO11 lebih tangguh dalam menangani variasi skala dan occlusions, terutama dalam skenario deteksi real-time.



BAB 3

METODE PENELITIAN

3.1 Jenis, Sifat, dan Pendekatan Penelitian

Dalam eksperimen, akan dilakukan penelitian dengan YOLO11 yang telah ditingkatkan kemampuannya dengan menambahkan empat buah modul Global Attention Mechanism dan berbagai penalaan parameter dan mengevaluasinya dengan mencari nilai mAP, Akurasi, Recall dan Presisi tertinggi dari beberapa percobaan.

Pendekatan yang akan digunakan dalam penelitian ini adalah eksperimental, di mana peneliti mengatur kondisi kontrol untuk mengamati efek dari penggunaan beberapa parameter tuning yang berbeda dalam sebuah arsitektur YOLO11.

3.2 Metode Pengumpulan Data



Gambar 3.1 20 tipe pesawat militer (A1-A20)

MAR20 adalah dataset untuk penelitian dalam pengenalan pesawat militer pada citra penginderaan jarak jauh, yang sangat penting dalam aplikasi seperti pengintaian, peringatan awal, dan analisis intelijen. MAR20 membantu mengatasi kekurangan dataset yang ada sebelumnya yang relatif sedikit menyediakan label tipe pesawat militer yang detail dan posisi yang akurat, serta menyediakan anotasi berbentuk orientasi yang bermanfaat untuk objek dengan sudut rotasi pesawat. Data diperoleh dari web <https://gcheng-nwpu.github.io>.

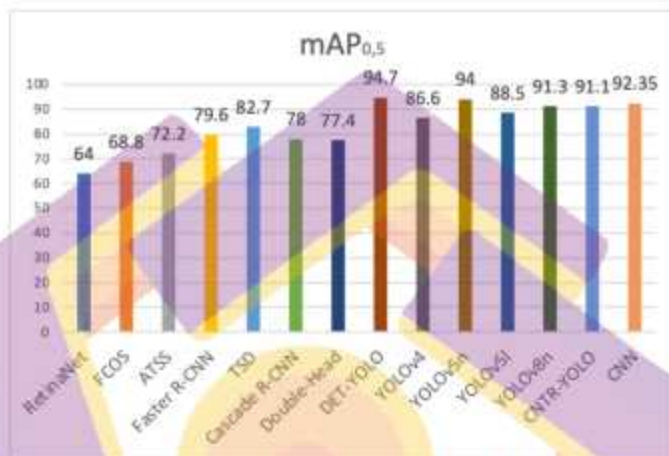
Dataset ini terdiri dari 3.842 citra dan mencakup 20 tipe pesawat militer (lihat Gambar 3.1), dengan total 22.341 instance, yang masing-masing dilengkapi dengan dua jenis anotasi yaitu bounding box horizontal dan bounding box berorientasi. Dataset MAR20 (Gambar 8) dikumpulkan dari 60 pangkalan militer di berbagai negara melalui Google Earth, sehingga mencakup variasi kondisi cuaca, musim, pencahayaan, dan occlusion yang meningkatkan kompleksitas pengenalan pesawat.



Gambar 3.2 Contoh tiga gambar dari dataset MAR20

Dalam penelitian Yu et al. [40], dilakukan evaluasi performa berbagai metode algoritma pengenalan objek pada dataset MAR20. Untuk tugas pengenalan dengan bounding box horizontal, metode TSD menunjukkan performa terbaik. Sedangkan untuk bounding box berorientasi, RoI Transformer memberikan hasil

terbaik. Dari hasil benchmark, pada metode TSD mencapai nilai mean Average Precision (mAP) sekitar 82,7% untuk bounding box horizontal, dan RoI Transformer mencapai mAP sekitar 82,72% untuk bounding box berorientasi.



Gambar 3.3 Perbandingan hasil mAP pada dataset MAR20 dari berbagai model

Gambar 3.3 menunjukkan perbandingan nilai mAP (mean Average Precision) dari berbagai model deteksi objek yang telah diuji pada dataset MAR20. Model DET-YOLO (Chen et al., 2024) mencapai mAP tertinggi sebesar 94,7%, yang menunjukkan performa terbaik dalam akurasi deteksi. Sementara itu, model lain seperti YOLOv5n, CNN (V., Kalahasti and Kotturu, 2024), CNTR-YOLO (Zhou et al., 2023) dan YOLOv8n (Chen et al., 2024) memiliki mAP yang lebih rendah di kisaran 94-91,1%. Grafik ini menggambarkan bahwa beberapa model lebih unggul dalam mengenali objek secara akurat pada dataset MAR20, dengan variasi yang cukup signifikan antar metode yang diuji.

Namun, masih terdapat beberapa gap yang menjadi tantangan penelitian lebih lanjut, yaitu: (1) tingkat kemiripan yang tinggi antar tipe pesawat militer

karena semua tipe merupakan kelas turunan dari pesawat sehingga fitur pembeda antar tipe sulit ditangkap; (2) variasi intrakelas yang besar akibat pengaruh faktor lingkungan seperti cuaca, musim, dan pencahayaan yang menyebabkan perbedaan signifikan visual antar instance tipe yang sama; serta (3) ketidakseimbangan jumlah data antar tipe pesawat tertentu yang berdampak pada performa pengenalan beberapa tipe yang memiliki jumlah data lebih sedikit.

Tabel 3.1 Modifikasi Nama Kelas

Old Class	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
New Class	SU-35	C-130	C-17	C-5	F-16	TU-160	E-3	B-52	P-3C	B-1B
Old Class	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20
New Class	E-8	TU-22	F-15	KC-135	F-22	F-18	TU-95	KC-10	SU-34	SU-24

Untuk mengatasi hal tersebut, maka dalam penelitian ini dilakukan proses preprocessing dan augmentasi agar dataset menjadi lebih bervariasi dan seimbang. Dataset di-preprocessing dengan bantuan aplikasi Roboflow yang di-resize ke ukuran 640x640 dan dilakukan modifikasi nama kelas sesuai dengan nama jenis pesawat sebenarnya. Tabel 3.1, menunjukkan daftar kelas pesawat beserta nama pengganti atau penamaan ulang yang digunakan untuk masing-masing kelas. Misalnya, kelas A1 diberi nama SU-35 yang merupakan pesawat tempur canggih buatan Rusia. Ada berbagai jenis pesawat yang tercantum, mulai dari pesawat tempur seperti F-16 dan F-22, pesawat angkut besar seperti C-130 dan C-17, hingga pesawat pengintai dan tanker seperti E-3 dan KC-135. Penamaan ulang ini digunakan untuk merujuk pada tipe atau model pesawat tertentu agar lebih mudah dikenali dan diklasifikasikan berdasarkan fungsi dan kemampuan masing-masing.

Data ini penting untuk mempermudah identifikasi dan komunikasi dalam operasional maupun strategi militer.

Setelah tahap preprocessing selanjutnya data diaugmentasi dengan proses shear $-15/+15$ (horizontal/vertical), salt and pepper noise 5%, dan Random Gaussian blur 0-2 pixels, hingga didapatkan data sebanyak 9.173 data gambar. Kemudian data dibagi dengan proporsi train 70% (6.604 gambar), valid 8% (733 gambar) dan test 20% (1836 gambar).

Ukuran dataset sangat mempengaruhi proporsi data yang dialokasikan untuk training, validation, dan test. Pada dataset yang besar, proporsi data untuk validasi dan test bisa lebih kecil karena jumlah absolut sampel tetap cukup untuk evaluasi yang andal. Sebaliknya, pada dataset kecil, proporsi validasi dan test kadang perlu diperbesar agar hasil evaluasi lebih terpercaya dan tidak bias [41][42].

Sehingga untuk penelitian ini yang memiliki 9.173 data gambar, proporsi train 72%, sangat cukup data untuk di-train dengan baik. Validasi 8%, memberikan ukuran yang cukup untuk tuning tanpa mengorbankan data training yang banyak. Test 20%, cukup untuk pengujian akhir model dalam kondisi dunia nyata.

Tabel 3.4 Tabel Analisa untuk Learning Rate 0.00417

Model	Epoch	LR	Optimizer	Hasil					Metriks	Hasil					Gain mAP50 (%)	Gain mAP50-95 (%)
				P (%)	R (%)	mAP50 (%)	mAP50-95 (%)	FPS		P (%)	R (%)	mAP50 (%)	mAP50-95 (%)	FPS		
YOLO11n	80	0.00417	Adam						HGAM							
			Adamw													
			AdanVR													
			RAdam													
			NAdam													
			DGB													

Tabel 3.5 Tabel Hasil berdasarkan mAP Terbaik

Peringkat	Model	Optimizer	LR	mAP50-95 (%)	mAP50 (%)	FPS
1						
2						
3						
4						
5						
6						

Tabel 3.5 menunjukkan hasil pilihan eksperimen dari berbagai kombinasi optimizer dan learning rate dalam melatih model, dengan fokus pada nilai mAP terbaik. Kolom-kolom tabel mencakup model, jenis optimizer, nilai learning rate, metrik evaluasi seperti mean Average Precision (mAP50-95 dan mAP50), serta kecepatan pemrosesan (FPS). Tabel ini membantu membandingkan performa model dari segi akurasi dan efisiensi, sehingga memudahkan pemilihan konfigurasi training yang optimal untuk mendapatkan hasil terbaik.

Tabel 3.6 Tabel Hasil berdasarkan Kecepatan Terbaik

No	Model	Optimizer	LR	FPS	mAP50-95 (%)	mAP50 (%)
1						
2						
3						

Tabel 3.6 menyajikan hasil eksperimen dengan fokus pada kecepatan pemrosesan terbaik (FPS) dari berbagai kombinasi optimizer dan learning rate. Setiap baris mencatat konfigurasi training beserta performa model dalam metrik mAP (mAP50-95, mAP50), dan kecepatan (FPS). Tabel ini berguna untuk mengevaluasi trade-off antara kecepatan inferensi dan kualitas prediksi, sehingga membantu memilih parameter yang menghasilkan model dengan respons paling cepat tanpa mengorbankan performa secara signifikan.

Tabel 3.7 Tabel Hasil Peningkatan (Gain) Terbaik pada mAP50 dan mAP50-95 setelah dimodifikasi

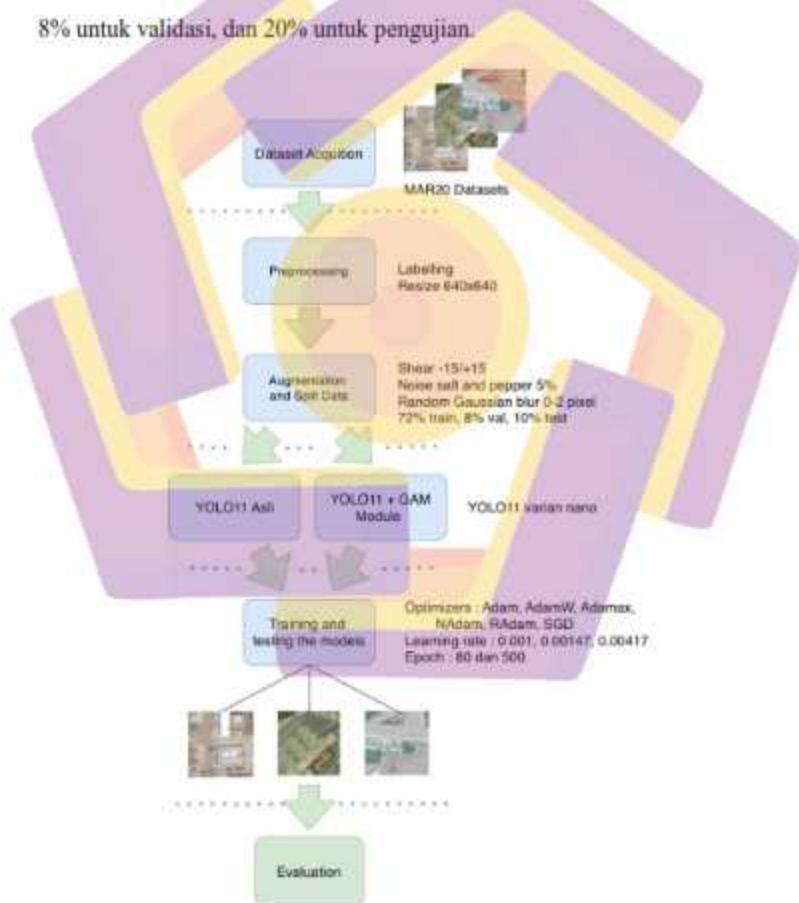
Peringkat Gain	Optimizer	LR	mAP50 (%) (Sebelum → Sesudah)	Gain mAP50 (%)	mAP50-95 (%) (Sebelum → Sesudah)	Gain mAP50-95 (%)
1						
2						
3						
4						
5						

Tabel 3.7 menampilkan hasil eksperimen yang membandingkan kinerja model sebelum dan sesudah modifikasi dengan tambahan teknik GAM pada arsitektur YOLO11n. Setiap baris menunjukkan konfigurasi optimizer, learning rate, dan nilai mAP dari YOLO11n asli dan setelah ditambahkan GAM, diikuti oleh persentase peningkatan (Gain %) dan kecepatan pemrosesan (FPS). Tabel ini memberikan gambaran efektivitas modifikasi GAM dalam meningkatkan akurasi model sekaligus mempertimbangkan dampaknya terhadap kecepatan inferensi.

3.4 Alur Penelitian

Gambar 3.4, menunjukkan alur penelitian yang dimulai dengan pengambilan data citra untuk digunakan dalam analisis deteksi objek. Setelah itu,

proses preprocessing dilakukan yang mencakup langkah resize 640x640 untuk menyesuaikan ukuran gambar agar sesuai dengan input model serta modifikasi kelas menjadi 20 nama pesawat tempur untuk mengubah label dalam dataset. Selanjutnya, teknik augmentasi diterapkan untuk memperkaya variasi gambar dengan menambahkan shear, menambahkan noise, dan menggunakan Gaussian blur. Setelah augmentasi, dataset dibagi menjadi tiga bagian: 72% untuk pelatihan, 8% untuk validasi, dan 20% untuk pengujian.



Gambar 3.4 Alur Penelitian

Model yang digunakan adalah YOLO11 tipe n, yang dilatih dalam dua pendekatan: pelatihan menggunakan model YOLO11 asli dan pelatihan yang dimodifikasi. Dalam dua pendekatan ini, beberapa pengaturan penting seperti optimizer (Adam, Adamax, AdamW, NAdam, RAdam, SGD) dan learning rate bervariasi dari 0,001; 0,00147 dan 0,00417 untuk menemukan kombinasi terbaik. Epoch juga bervariasi, dengan beberapa percobaan menggunakan 80 dan 500 iterasi.

Setelah proses pelatihan selesai, model dievaluasi berdasarkan metrik kinerja yang meliputi mean Average Precision (mAP), akurasi, recall, presisi dan kecepatan. Proses evaluasi ini bertujuan untuk menentukan efektivitas model dalam mendeteksi objek pada citra penginderaan jauh. Penelitian resmi berakhir setelah evaluasi model, memberikan wawasan tentang peningkatan yang dicapai serta aspek-aspek yang masih memerlukan perbaikan.

3.5 Evaluasi

Dalam tugas deteksi target, metrik seperti Recall (R), Precision (P), dan Average Precision (AP) biasanya digunakan untuk evaluasi. Recall mewakili proporsi target yang terdeteksi dengan benar dari total jumlah target, yang dihitung menggunakan rumus yang ditunjukkan dalam Persamaan (3.1). Presisi mewakili proporsi target yang terdeteksi dengan benar dari total target yang diprediksi, seperti yang ditunjukkan dalam Persamaan (3.2).

$$R = \frac{TP}{TP + FN} \times 100\% \quad (3.1)$$

$$P = \frac{TP}{TP + FP} \times 100\% \quad (3.2)$$

$$AP = \int_0^1 P_i(R_i) dR_i \quad (3.3)$$

$$mAP = \frac{\sum_{i=1}^C AP_i}{C} \times 100\% \quad (3.4)$$

$$FPS = \frac{1000}{\text{processing time per frame (ms)}} \quad (3.5)$$

Di mana TP mewakili jumlah target yang teridentifikasi dengan benar, FP adalah jumlah target yang terdeteksi secara keliru, dan FN adalah jumlah target yang tidak terdeteksi. Average Precision (AP) adalah ukuran dari area di bawah kurva Precision-Recall (P-R), di mana recall dipetakan pada sumbu x dan precision dipetakan pada sumbu y. Rumus perhitungan untuk Presisi Rata-Rata (AP) diberikan oleh Persamaan (3.3).

Untuk mendapatkan mean Average Precision (mAP) untuk beberapa kelas, nilai AP untuk setiap kelas dirata-ratakan. Rumus untuk menghitung mAP ditunjukkan dalam Persamaan (3.4).

Rumus kecepatan pada hasil YOLO biasanya mengacu pada FPS (Frames Per Second), yaitu banyaknya frame gambar yang bisa diproses dalam satu detik. Rumus menghitung FPS ditunjukkan dalam Persamaan (3.5). Dimana waktu pemrosesan per frame merupakan total waktu dari semua tahapan preprocess time, inference time, postprocess time.

3.5.1 Penggunaan mAP dalam Deteksi Pesawat dengan YOLO

Dalam mendeteksi pesawat menggunakan YOLO, *mean Average Precision* (mAP) digunakan sebagai metrik utama untuk mengukur kinerja model. mAP

memberikan gambaran menyeluruh tentang seberapa baik model mendeteksi dan mengklasifikasikan objek (pesawat) pada berbagai kondisi dan skala, serta membandingkan performa antar model secara objektif.

Alasan Penggunaan mAP dalam Deteksi Pesawat dengan YOLO:

1. Akurasi Deteksi yang Menyeluruh

mAP mengukur rata-rata presisi pada berbagai tingkat recall, sehingga memberikan penilaian yang komprehensif terhadap kemampuan model dalam mendeteksi pesawat, baik yang berukuran kecil, padat, maupun dalam latar belakang yang kompleks [8][19][43][10].

2. Standar Perbandingan Antar Model YOLO

mAP memudahkan perbandingan performa antara berbagai versi dan modifikasi YOLO, serta dengan algoritma deteksi lain, untuk memastikan model yang dipilih benar-benar unggul dalam tugas deteksi pesawat [19][43][44].

3. Sensitivitas terhadap Kesalahan Deteksi

mAP sensitif terhadap kesalahan seperti deteksi ganda, salah deteksi, dan missed detection, sehingga sangat cocok untuk aplikasi kritis seperti deteksi pesawat pada citra satelit atau remote sensing [19][8][10][45].

3.5.2 mAP

Mean Average Precision (mAP) adalah metrik evaluasi standar emas (gold standard) untuk mengukur performa model deteksi objek pada YOLO. Nilai mAP tidak dihitung secara manual, melainkan dihasilkan oleh script evaluasi dari

framework pelatihan YOLO setelah model selesai dilatih dan diuji pada validation set atau test set. Metrik ini secara komprehensif mengukur seberapa baik model dalam melokalisasi (menemukan) objek dan mengklasifikasi (mengidentifikasi) objek tersebut dengan benar.

Cara Menghitung mAP:

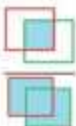
Perhitungan mAP adalah proses multi-langkah yang kompleks. Berikut adalah uraian langkah demi langkahnya:

1. Intersection over Union (IoU)

Dalam tugas deteksi objek, sistem menghasilkan output berupa bounding box—yaitu kotak yang menandai lokasi prediksi objek tertentu dalam citra. Kotak ini menggambarkan posisi objek di dalam gambar. Untuk menilai kinerja model deteksi objek yang telah dilatih, salah satu metode evaluasi yang umum digunakan adalah Intersection over Union (IoU). Metode ini membandingkan dua bounding box yakni hasil prediksi dari model dan bounding box ground truth (label sebenarnya). IoU menghitung tingkat kesamaan antara keduanya dengan mengukur proporsi area irisan terhadap gabungan total area kedua kotak. Dengan adanya nilai IoU, kita dapat menghitung berbagai metrik evaluasi lainnya, seperti precision, recall, dan sebagainya. Persamaan intersection over union sebagai berikut:

$$IoU = \frac{area(BB_{prediksi} \cap BB_{groundTruth})}{area(BB_{prediksi} \cup BB_{groundTruth})} \quad (3.6)$$

Persamaan Intersection Over Union dapat diilustrasikan dalam gambar berikut:

$$IoU = \frac{\text{area irisan}}{\text{area gabungan}} = \frac{\text{img}}{\text{img}} \quad (3.7)$$


Berdasarkan ilustrasi di atas, terlihat bahwa rumus untuk menghitung nilai IoU merupakan rasio antara luas area irisan dengan luas area gabungan dari dua bounding box. Skor Intersection over Union (IoU) diperoleh dengan membagi luas irisan tersebut dengan luas gabungan keduanya. Setelah skor IoU didapatkan, kualitasnya dinilai berdasarkan seberapa besar tumpang tindih atau seberapa dekat posisi antara bounding box hasil prediksi dan bounding box ground truth: semakin besar irisan (atau semakin dekat keduanya), semakin baik skor IoU yang dihasilkan. Berikut adalah contoh ilustrasi skor IoU yang dikategorikan baik, lumayan dan buruk.



Gambar 3.5 Ilustrasi skor IOU

Berdasarkan Gambar 3.5, dapat disimpulkan bahwa skor akan semakin tinggi seiring dengan semakin dekatnya posisi bounding box prediksi

terhadap bounding box ground truth, yang berarti area irisan antara keduanya juga semakin luas.

2. Menentukan True Positive (TP), False Positive (FP), dan False Negative (FN)

Untuk setiap prediksi, kita menilainya berdasarkan ambang batas IoU (misalnya, 0.5):

- True Positive (TP): Deteksi yang benar. Terjadi jika:
 1. $\text{IoU} \geq \text{Ambang Batas}$ (misal, 0.5)
 2. Kelas yang diprediksi BENAR (misal, prediksi "F-16" untuk objek F-16 asli).
- False Positive (FP): Deteksi yang salah. Terjadi jika:
 1. $\text{IoU} < \text{Ambang Batas}$ (lokasi salah).
 2. Kelas yang diprediksi SALAH (lokasi benar, tapi salah identifikasi, misal prediksi SU-35 untuk objek SU-34).
 3. Model mendeteksi objek yang sebenarnya tidak ada.
- False Negative (FN): Objek yang gagal dideteksi. Terjadi jika ada objek asli di citra, tetapi model tidak membuat prediksi untuknya.

3. Menghitung Precision dan Recall

- Precision (Presisi): Mengukur akurasi prediksi. "Dari semua deteksi yang saya buat, berapa persen yang benar?"
 - Rumus: $\text{TP} / (\text{TP} + \text{FP})$

- Recall (Cakupan): Mengukur seberapa banyak objek asli yang berhasil ditemukan. "Dari semua objek yang seharusnya ada, berapa persen yang berhasil saya temukan?"
 - Rumus: $TP / (TP + FN)$

4. Kurva Precision-Recall (P-R Curve)

Dengan memvariasikan ambang batas skor kepercayaan (confidence score) dari 0 hingga 1, kita bisa mendapatkan banyak pasangan nilai Precision dan Recall. Jika digambarkan dalam grafik, ini akan membentuk Kurva P-R.

5. Average Precision (AP)

AP adalah luas di bawah Kurva P-R untuk satu kelas spesifik. Nilai AP yang tinggi menunjukkan bahwa model dapat mempertahankan presisi yang tinggi bahkan saat mencoba menemukan lebih banyak objek (recall tinggi).

6. Mean Average Precision (mAP)

mAP adalah rata-rata dari nilai AP di semua kelas.

- Rumus: $(AP_{\text{kelas1}} + AP_{\text{kelas2}} + \dots + AP_{\text{kelasN}}) / N$
(di mana N adalah jumlah total kelas).

3.5.3 mAP50 dan mAP50-95

Penilaian mAP pada YOLO ada 2 macam, mAP50 dan mAP50-95. mAP50 adalah singkatan dari mean Average Precision yang dihitung pada satu ambang batas IoU, yaitu 0.50, yang berarti sebuah deteksi dari model akan dianggap benar jika kotak prediksinya memiliki tumpang tindih minimal 50% ($IoU \geq 0.50$) dengan kotak sebenarnya. IoU (Intersection over Union) adalah metrik yang sering dipakai dalam computer vision, terutama pada tugas deteksi objek dan segmentasi, untuk

mengukur seberapa baik prediksi (bounding box atau mask) cocok dengan ground truth (kebenaran dasar). Metrik ini lebih fokus pada kemampuan model untuk mendeteksi keberadaan objek. Selama kotaknya cukup tumpang tindih (50%), model sudah dianggap berhasil. Ini adalah standar yang lebih longgar dan biasa digunakan dalam pengukuran deteksi obyek. Sebagai contoh jika model mendeteksi pesawat, tetapi kotaknya sedikit terlalu besar dan mencakup sebagian latar belakang, selama IoU-nya masih di atas 0,50, deteksi itu tetap dihitung sebagai deteksi yang benar.

mAP50-95 adalah metrik yang jauh lebih ketat dan komprehensif. Ini adalah nilai rata-rata dari mAP yang dihitung pada 10 ambang batas IoU yang berbeda, mulai dari 0.50 hingga 0.95 dengan interval 0.05. Artinya, Perhitungan mAP dilakukan secara terpisah untuk $\text{IoU} \geq 0.50$, $\text{IoU} \geq 0.55$, $\text{IoU} \geq 0.60$, ..., hingga $\text{IoU} \geq 0.95$. Kemudian, hasil dari ke-10 perhitungan mAP tersebut dirata-ratakan untuk mendapatkan satu nilai akhir. Metrik ini tidak hanya menilai apakah model bisa mendeteksi objek, tetapi juga seberapa presisi model dalam melokalisasi objek tersebut (seberapa pas kotaknya). Intinya, mAP50 menjawab pertanyaan "Apakah model menemukan objeknya?", sedangkan mAP50-95 menjawab pertanyaan "Seberapa sempurna model menemukan dan melingkupi objeknya?".[46][47][48][49]

3.5.4 Keunggulan YOLO dalam Deteksi Pesawat:

1. Deteksi Real Time

YOLO dirancang untuk deteksi objek secara cepat dan efisien, sangat penting untuk aplikasi yang membutuhkan respons waktu nyata seperti pemantauan bandara atau militer [19][50][8][10].

2. Kemampuan Deteksi Multi-Skala

Berbagai modifikasi YOLO (misal YOLOv5, YOLOv8, YOLO-Extract) meningkatkan kemampuan mendeteksi pesawat dengan ukuran dan orientasi berbeda, serta dalam kondisi latar belakang yang rumit [8][10][19][43][45].

3. Peningkatan mAP Lewat Inovasi

Berbagai penelitian menunjukkan bahwa modifikasi pada YOLO secara konsisten meningkatkan nilai mAP, menandakan deteksi yang lebih akurat dan andal [8][10][19][43][45][44].

Dari pertimbangan di atas, maka mAP digunakan dalam analisis deteksi pesawat dengan YOLO karena mampu memberikan penilaian akurasi yang menyeluruh, sensitif terhadap berbagai jenis kesalahan deteksi, dan memudahkan perbandingan antar model.

BAB 4

HASIL PENELITIAN DAN PEMBAHASAN

4.1 Pengumpulan Dataset

Dataset yang digunakan adalah MAR20 (Military Aircraft Recognition) yang terdiri dari 20 kelas pesawat militer [40]. Penggunaan dataset ini memungkinkan peneliti untuk mengeksplorasi kemampuan deteksi pesawat militer dengan latar belakang yang beragam dan kompleks. Dataset ini terdiri dari 3.842 citra dan mencakup 20 tipe pesawat militer, dengan total 22.341 instance, yang masing-masing dilengkapi dengan dua jenis anotasi yaitu bounding box horizontal dan bounding box berorientasi. Dataset MAR20 dikumpulkan dari 60 pangkalan militer di berbagai negara melalui Google Earth, sehingga mencakup variasi kondisi cuaca, musim, pencahayaan, dan berbagai rintangan yang meningkatkan kompleksitas identifikasi pesawat. Contoh dataset ditampilkan dalam Gambar 4.1.



Gambar 4.1 Contoh dataset MAR20

4.2 Perangkat yang digunakan

Berikut ini adalah spesifikasi perangkat keras dan perangkat lunak yang digunakan untuk melakukan penelitian.

1. Spesifikasi perangkat keras:
 - a. CPU AMD Ryzen 5 5600
 - b. Memory DDR 32 GB
 - c. NVIDIA GeForce RTX 3050 8 GB
2. Spesifikasi perangkat lunak:
 - a. OS Ubuntu 24.04.1 LTS
 - b. Visual Studio Code v.1.99.3
 - c. Ultralytics 8.3.165
 - d. Python 3.12.3 torch 2.5.1+cu124 CUDA

4.3 Pra Pemrosesan Data

Dalam penelitian ini, aplikasi Roboflow digunakan untuk membantu proses pra pemrosesan data agar dataset menjadi lebih baik dan optimal. Dataset diresize ke ukuran 640x640 dan dilakukan modifikasi nama kelas sesuai dengan nama jenis pesawat sebenarnya. Tabel 4.1, menunjukkan daftar kelas pesawat beserta nama pengganti atau penamaan ulang yang digunakan untuk masing-masing kelas.

Misalnya, kelas A1 diberi nama SU-35 yang merupakan pesawat tempur canggih buatan Rusia. Ada berbagai jenis pesawat yang tercantum, mulai dari pesawat tempur seperti F-16 dan F-22, pesawat angkut besar seperti C-130 dan C-17, hingga pesawat pengintai dan tanker seperti E-3 dan KC-135. Penamaan ulang ini digunakan untuk merujuk pada tipe atau model pesawat tertentu agar lebih mudah dikenali dan diklasifikasikan berdasarkan fungsi dan kemampuan masing-masing. Data ini penting untuk mempermudah identifikasi dan komunikasi dalam operasional maupun strategi militer.

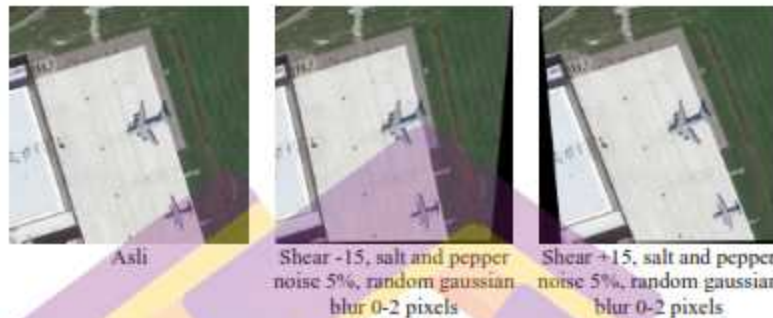
Tabel 4.1 Modifikasi Nama Kelas

Kelas Lama	Kelas Baru
A1	SU-35
A2	C-130
A3	C-17
A4	C-5
A5	F-16
A6	TU-160
A7	E-3
A8	B-52
A9	P-3C
A10	B-1B
A11	E-8
A12	TU-22
A13	F-15
A14	KC-135
A15	F-22
A16	F-18
A17	TU-95
A18	KC-10
A19	SU-34
A20	SU-24

4.4 Augmentasi dan Split Data

Augmentasi data adalah teknik dalam pembelajaran mesin untuk memperluas dataset pelatihan dengan membuat versi modifikasi dari gambar yang sudah ada. Tujuannya adalah untuk meningkatkan jumlah data, membantu model belajar variasi alami, dan mencegah overfitting. Augmentasi dapat dilakukan dengan menerapkan transformasi seperti membalik (flip), merotasi (rotate), mengubah skala, menggeser, memotong, atau menyesuaikan warna dan kecerahan pada gambar asli. Pada penelitian ini, teknik augmentasi yang digunakan adalah shear [51], salt & papper noise [52], dan Gaussian blur [53]. Tiga augmentasi ini meniru kondisi nyata pada proses deteksi pesawat militer menggunakan citra

penginderaan jarak jauh, seperti: distorsi perspektif (shear), gangguan sensor/transmisi (salt & pepper), dan blur (defocus/gerak).



Gambar 4.2 Hasil augmentasi yang menghasilkan dua citra baru dengan beberapa keadaan

Contoh visual Gambar 4.2 menunjukkan bagaimana satu citra dapat menghasilkan banyak citra baru dengan posisi dan keadaan yang beragam. Dengan mengombinasikan beberapa jenis augmentasi dengan proses shear -15/+15 (horizontal/vertical), salt and pepper noise 5%, dan Random Gaussian blur 0-2 pixels, satu citra dapat menghasilkan banyak variasi baru. Implementasi augmentasi data dalam tesis ini menggunakan Roboflow.

Setelah tahap augmentasi akhirnya didapatkan data sebanyak 9.173 data gambar. Kemudian data dibagi sesuai proporsi penelitian Chen [19] menjadi data latih 72% (6.604 gambar), data valid 8% (733 gambar) dan data tes 20% (1.836 gambar).

Ukuran dataset sangat mempengaruhi proporsi data yang dialokasikan untuk training, validation, dan test. Pada dataset yang besar, proporsi data untuk validasi dan test bisa lebih kecil karena jumlah absolut sampel tetap cukup untuk

evaluasi yang andal. Sebaliknya, pada dataset kecil, proporsi validasi dan test kadang perlu diperbesar agar hasil evaluasi lebih terpercaya dan tidak bias [41][42].

Sehingga untuk penelitian ini yang memiliki 9.173 data gambar, proporsi train 72%, sangat cukup data untuk dilatih dengan baik. Validasi 8%, memberikan ukuran yang cukup untuk tuning tanpa mengorbankan data training yang banyak. Test 20%, cukup untuk pengujian akhir model dalam kondisi dunia nyata.

Gambar 4.3 menyajikan distribusi jumlah anotasi untuk setiap kelas pesawat yang digunakan dalam dataset penelitian ini. Grafik ini secara jelas menunjukkan adanya ketidakseimbangan kelas (class imbalance) yang signifikan, di mana beberapa kelas memiliki representasi yang sangat tinggi sementara yang lain jauh lebih sedikit. Sebagai contoh, kelas FA-18 memiliki 5.863 anotasi, yang merupakan jumlah terbanyak, sedangkan kelas KC-10 hanya diwakili oleh 763 anotasi. Meskipun tampak sebagai sebuah kelemahan, distribusi yang tidak seimbang ini tidak selalu perlu disamakan dan justru seringkali mencerminkan distribusi target di dunia nyata, sehingga melatih model untuk lebih robust.



Gambar 4.3 Jumlah Anotasi per Kelas

Pendekatan ini sejalan dengan praktik dalam penyusunan dataset berskala besar yang menjadi tolok ukur dalam penelitian deteksi objek. Studi kasus yang

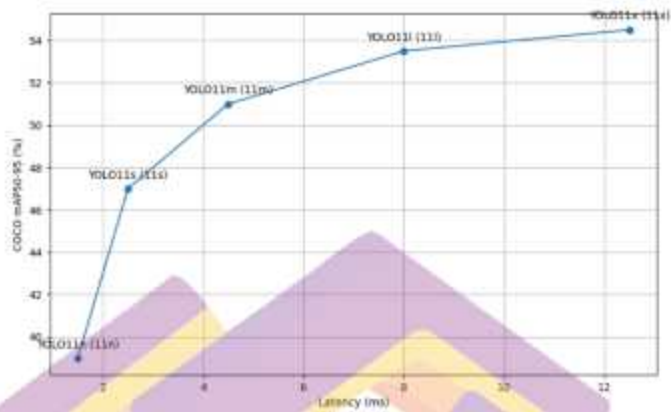
paling relevan adalah dataset Microsoft COCO (Common Objects in Context), yang secara sengaja dirancang untuk memiliki distribusi kelas yang natural dan tidak seimbang. Para pengembang COCO berargumen bahwa distribusi kelas yang mengikuti frekuensi alami objek di dunia nyata (long-tail distribution) memberikan tantangan yang lebih realistis bagi algoritma deteksi objek, karena model harus belajar untuk mendeteksi objek yang sering muncul maupun yang jarang muncul [54]. Oleh karena itu, ketidakseimbangan pada dataset penelitian ini dapat dipandang sebagai representasi yang lebih akurat dari frekuensi kemunculan berbagai jenis pesawat militer dalam citra penginderaan jarak jauh, sehingga melatih model agar mampu berkinerja baik dalam aplikasi dunia nyata yang sesungguhnya.

Selain argumen bahwa distribusi kelas yang tidak seimbang mencerminkan frekuensi target di dunia nyata, ketidakseimbangan pada dataset ini dapat diterima secara metodologis karena volume data yang besar dan penerapan strategi pembagian data yang cermat. Dengan total 9.173 gambar yang menghasilkan 51.265 anotasi, bahkan kelas minoritas seperti KC-10 memiliki ratusan sampel (763 anotasi), jumlah yang cukup untuk dipelajari oleh jaringan neural modern. Kunci utamanya terletak pada pembagian data dengan proporsi 72% untuk pelatihan, 8% untuk validasi, dan 20% untuk pengujian yang dilakukan secara stratified (berstrata). Metode stratified splitting ini memastikan bahwa distribusi kelas yang tidak seimbang dari dataset asli dipertahankan secara proporsional di ketiga subset data. Hal ini sangat krusial karena subset pelatihan yang representatif memungkinkan model untuk belajar fitur dari semua kelas tanpa bias yang

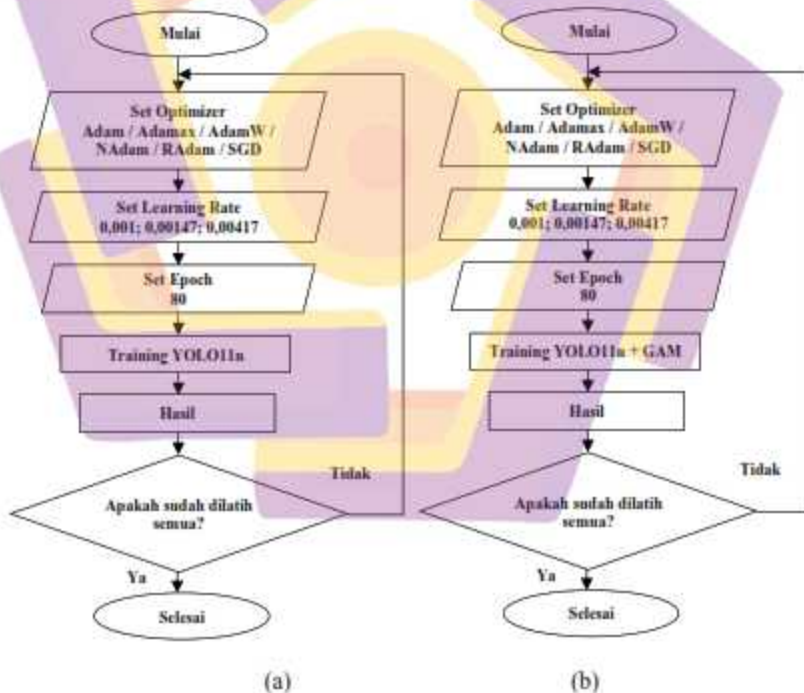
berlebihan, sementara subset validasi dan pengujian yang representatif memastikan bahwa evaluasi kinerja model benar-benar mencerminkan kemampuannya untuk menggeneralisasi pada semua kelas, baik yang mayoritas maupun minoritas. Praktik ini sejalan dengan studi sistematis mengenai masalah ketidakseimbangan kelas pada Convolutional Neural Networks (CNNs), yang menekankan bahwa evaluasi yang andal pada set pengujian yang representatif adalah fundamental untuk menarik kesimpulan yang valid tentang performa model [55]. Dengan demikian, pendekatan ini memastikan bahwa ketidakseimbangan data menjadi tantangan yang terukur, bukan kelemahan metodologis.

4.5 Pelatihan Model

Metodologi eksperimen yang digunakan dalam penelitian ini melibatkan analisis komparatif varian YOLO11n (nano) untuk mengevaluasi trade-off kinerja antara kecepatan dan akurasi. Gambar 4.4 mengilustrasikan hubungan antara latensi dan skor mAP50-95 pada dataset COCO [20]. YOLO11n dicirikan oleh latensi yang lebih rendah, beroperasi sekitar 1,5 ms, dengan skor mAP50-95 sekitar 39%. Sebaliknya, YOLO11s, meskipun menunjukkan latensi yang sedikit lebih tinggi sekitar 2,5 ms, menunjukkan peningkatan akurasi yang signifikan dengan skor mAP50-95 sekitar 47%. Dari gambaran penilaian komparatif ini menunjukkan bahwa YOLO11n digunakan untuk aplikasi yang memprioritaskan kecepatan, dan YOLO11s dipakai untuk keakurasian yang lebih tinggi tapi sayangnya kurang efisien. Dari alasan tersebut maka kami memilih tipe n karena lebih memprioritaskan kecepatan namun berusaha tetap mempertahankan akurasi yang tinggi dengan melakukan parameter tuning dan modifikasi arsitektur.



Gambar 4.4 Perbandingan antara varian YOLO11



Gambar 4.5 Alur pelatihan menggunakan YOLO11 varian n

Gambar 4.5 (a) menerangkan tentang alur menggunakan YOLO11 varian n. Alur dimulai dengan inisialisasi, diikuti dengan pengaturan optimizer (Adam, Adamax, AdamW, NAdam, RAdam, atau SGD), learning rate 0,001; 0,00147 dan 0,00417, serta 80 epoch. Model YOLO11n kemudian dilatih dan hasilnya dievaluasi. Proses ini berulang untuk setiap optimizer yang dipilih hingga semua kombinasi pelatihan selesai, setelah itu proses berakhir. Hampir sama dengan proses (a), Gambar 4.5 (b) memulai dengan inisialisasi lalu diikuti dengan pengaturan optimizer, learning rate dan epoch yang sama persis. Namun yang membedakan adalah menggunakan YOLO11 varian n yang telah dimodifikasi.

4.6 Hasil Eksperimen

Sebelum menjalankan proses pelatihan YOLO11, terlebih dahulu arsitektur dimodifikasi dengan melakukan penambahan modul Global Attention Mechanism. Penambahan modul dilakukan dengan memodifikasi file-file berikut:

1. Modul Global Attention Mechanism (GAM) [39] terdapat di:

`ultralytics/nn/modules/attention.py`

Script modul Global Attention Mechanism:

```
# Global Attention Mechanism
# https://arxiv.org/abs/2112.05561
class GAM_Attention(nn.Module):
    def __init__(self, c1, c2, group=True, rate=4):
        super(GAM_Attention, self).__init__()

        self.channel_attention = nn.Sequential(
            nn.Linear(c1, int(c1 / rate)),
            nn.ReLU(inplace=True),
            nn.Linear(int(c1 / rate), c1),
        )

        self.spatial_attention = nn.Sequential(
            nn.Conv2d(c1, c1 // rate, kernel_size=7, padding=3, groups=rate)
            if group
            else nn.Conv2d(c1, int(c1 / rate), kernel_size=7, padding=3),
```

```

nn.BatchNorm2d(int(c1 / rate)),
nn.ReLU(inplace=True),
nn.Conv2d(c1 // rate, c2, kernel_size=7, padding=3, groups=rate)
if group
else nn.Conv2d(int(c1 / rate), c2, kernel_size=7, padding=3),
nn.BatchNorm2d(c2),
)

def forward(self, x):
    b, c, h, w = x.shape
    x_permute = x.permute(0, 2, 3, 1).view(b, -1, c)
    x_att_permute = self.channel_attention(x_permute).view(b, h, w, c)
    x_channel_att = x_att_permute.permute(0, 3, 1, 2)
    x = x * x_channel_att

    x_spatial_att = self.spatial_attention(x).sigmoid()
    x_spatial_att = channel_shuffle(x_spatial_att, 4) # last shuffle
    out = x * x_spatial_att
    return out

# GAM Attention End

```

2. `ultralytics/nn/modules/__init__.py`

Berupa modul jaringan saraf Ultralytics.

Modul ini menyediakan akses ke berbagai komponen jaringan saraf yang digunakan dalam model Ultralytics, termasuk blok konvolusi, mekanisme atensi, komponen transformer, serta head deteksi/segmentasi.

3. `ultralytics/nn/tasks.py`

4. `ultralytics/cfg/models/11/yolo11-GAM-Attention.yaml`

Potongan script implementasi 4 buah GAM pada struktur YOLO11:

```

# YOLO11+GAM neck-head
head:
  - [-1, 1, nn.Upsample, [None, 2, "nearest"]]
  - [[-1, 6], 1, Concat, [1]] # cat backbone P4
  - [-1, 2, C3k2, [512, False]] # 13
  - [-1, 2, GAM_Attention, [512,512]] # modul GAM 1

  - [-1, 1, nn.Upsample, [None, 2, "nearest"]]
  - [[-1, 4], 1, Concat, [1]] # cat backbone P3
  - [-1, 2, C3k2, [256, False]] # 16 (P3/8-small)
  - [-1, 2, GAM_Attention, [256,256]] # modul GAM 2

```

```

- [-1, 1, Conv, [256, 3, 2]]
- [[-1, 13], 1, Concat, [1]] # cat head P4
- [-1, 2, C3k2, [512, False]] # 19 (P4/16-medium)
- [-1, 2, GAM_Attention, [512,512]] # modul GAM 3

- [-1, 1, Conv, [512, 3, 2]]
- [[-1, 10], 1, Concat, [1]] # cat head P5
- [-1, 2, C3k2, [1024, True]] # 22 (P5/32-large)
- [-1, 2, GAM_Attention, [1024,1024]] # modul GAM 4

- [[16, 19, 22], 1, Detect, [nc]] # Detect(P3, P4, P5)

```

Sumber : Penelitian Aroj Subedi [56] yang telah dimodifikasi.

Source code : <https://github.com/arojsbedi/Improved-YOLOv8s/tree/main>

Setelah proses modifikasi selesai, dilanjutkan dengan melakukan proses pelatihan YOLO11 dengan berbagai parameter. Script untuk melakukan proses training YOLO adalah sebagai berikut:

1. Script untuk pelatihan YOLO11 asli

```

lyolo task=detect mode=train model=yolov11n.yaml
data="/MAR20.v3i.yolov11/data.yaml" optimizer=SGD lr0=0.001
epochs=80 imgsz=512 plots=True save=True val=True
project="/Dataset/MAR20/Eksperiment/runs80n/SGD-001/"

```

2. Script untuk pelatihan YOLO11 + GAM

```

lyolo task=detect mode=train model=yolov11n-GAM-Attention.yaml
data="/MAR20.v3i.yolov11/data.yaml" optimizer=AdamW lr0=0.001
epochs=80 imgsz=512 plots=True save=True val=True
project="/Dataset/MAR20/Eksperiment-GAM/runs80n/AdamW-001/"

```

Berikut adalah penjelasan script perintah melatih model YOLO11 untuk tugas deteksi objek:

- `yolo task=detect`: Ini menentukan bahwa tugas yang akan dilakukan adalah deteksi objek. YOLO (You Only Look Once) adalah keluarga model deep learning yang populer untuk tugas deteksi objek real-time.
- `mode=train`: Ini mengindikasikan bahwa model akan dilatih (bukan digunakan untuk inferensi atau validasi).
- `model=yolo11n.yaml`: Ini adalah jalur ke file konfigurasi YAML yang mendefinisikan arsitektur model. Dalam kasus ini, `yolo11n.yaml` menunjukkan bahwa ini adalah varian "nano" dari YOLO11 yang akan digunakan sebagai titik awal untuk pelatihan. Jika ingin melatih dengan model yang telah dimodifikasi modul GAM, maka `model=yolo11n.yaml` diganti menjadi `model=yolo11n-GAM-Attention.yaml`.
- `data="/MAR20.v3i.yolov11/data.yaml"`: Ini menunjuk ke file konfigurasi YAML yang berisi informasi tentang dataset Anda. File ini biasanya mendefinisikan jalur ke gambar pelatihan dan validasi, nama kelas, dan jumlah kelas.
- `optimizer=SGD`: Ini menentukan optimizer yang akan digunakan selama pelatihan. Optimizer dapat diganti sesuai dengan rancangan pelatihan, misal: Adam, RAdam, NAdam, atau lainnya.
- `lr0=0.001`: Ini menetapkan initial learning rate (tingkat pembelajaran awal) ke 0,001. Learning rate adalah hyperparameter yang mengontrol seberapa besar kita menyesuaikan bobot model dalam kaitannya dengan gradient loss. Lr0 bisa diganti sesuai rencana pelatihan, misal 0,001; 0,00147; atau 0,00417.

- `epochs=80`: Ini mengatur jumlah epoch untuk pelatihan. Satu epoch berarti seluruh dataset pelatihan telah dilewatkan maju dan mundur melalui jaringan saraf satu kali. Epoch dapat disesuaikan dengan rancangan pelatihan mulai dari 80, atau 500.
- `imgsz=512`: Ini menentukan ukuran gambar masukan untuk model. Dalam hal ini, gambar akan diubah ukurannya menjadi 512x512 piksel sebelum dimasukkan ke model.
- `plots=True`: Ini berarti bahwa plot pelatihan (seperti `loss curves`, `precision-recall curves`, dll.) akan dihasilkan dan disimpan selama pelatihan.
- `save=True`: Ini menginstruksikan sistem untuk menyimpan checkpoint model (bobot) selama atau setelah pelatihan.
- `val=True`: Ini berarti bahwa model akan dievaluasi pada dataset validasi setelah setiap epoch (atau secara berkala, tergantung pada implementasinya) untuk memantau kinerja dan mencegah overfitting.
- `project="./Dataset/MAR20/Eksperiment/runs80n/SGD-001/"`: Ini adalah jalur ke direktori tempat semua output pelatihan (bobot model, plot, log, dll.) akan disimpan. Struktur jalur ini menyarankan bahwa ini adalah bagian dari eksperimen di bawah dataset "MAR20", untuk run 80 epoch dari varian "n" (nano) model YOLO11, menggunakan optimizer SGD dengan learning rate 0,001.


```

train  n  param  module  arguments
0      -1  -1  465  ultralytics.nn.modules.conv.Conv  [7, 16, 3, 2]
1      -1  -1  4672 ultralytics.nn.modules.conv.Conv  [16, 32, 3, 2]
2      -1  -1  6640 ultralytics.nn.modules.block.C3k2  [32, 64, 3, False, 0.25]
3      -1  -1  16992 ultralytics.nn.modules.conv.Conv  [64, 64, 3, 2]
4      -1  -1  26980 ultralytics.nn.modules.block.C3k2  [64, 128, 3, False, 0.25]
5      -1  -1  147712 ultralytics.nn.modules.conv.Conv  [128, 128, 3, 2]
6      -1  -1  37040 ultralytics.nn.modules.block.C3k2  [128, 128, 3, True]
7      -1  -1  295424 ultralytics.nn.modules.conv.Conv  [128, 256, 3, 2]
8      -1  -1  340112 ultralytics.nn.modules.block.C3k2  [256, 256, 3, True]
9      -1  -1  164080 ultralytics.nn.modules.block.SPPF  [256, 256, 1]
10     -1  -1  249720 ultralytics.nn.modules.block.C3P4  [256, 256, 1]
11     -1  -1  0  torch.nn.modules.upsampling.Upsample  [None, 2, 'nearest']
12     [-2, 8]  -1  0  ultralytics.nn.modules.conv.Conv  []
13     -1  -1  111396 ultralytics.nn.modules.block.C3k2  [384, 128, 3, False]
14     -1  -1  109380 ultralytics.nn.modules.attention.SPP_Attn2  [768, 128, 512]
15     -1  -1  0  torch.nn.modules.conv.conv1d.Conv1d  [None, 2, 'nearest']
16     [-2, 4]  -1  0  ultralytics.nn.modules.conv.Conv2d  [1]
17     -1  -1  32096 ultralytics.nn.modules.block.C3k2  [256, 64, 3, False]
18     -1  -1  27956 ultralytics.nn.modules.attention.SPP_Attn2  [64, 64, 256]
19     -1  -1  16992 ultralytics.nn.modules.conv.Conv  [64, 64, 3, 2]
20     [-2, 12]  -1  0  ultralytics.nn.modules.conv.Conv2d  [1]
21     -1  -1  38720 ultralytics.nn.modules.block.C3k2  [192, 128, 3, False]
22     -1  -1  130164 ultralytics.nn.modules.attention.SPP_Attn2  [384, 128, 512]
23     -1  -1  147712 ultralytics.nn.modules.conv.Conv  [128, 128, 3, 2]
24     [-2, 10]  -1  0  ultralytics.nn.modules.conv.Conv2d  [1]
25     -1  -1  37040 ultralytics.nn.modules.block.C3k2  [192, 128, 3, True]
26     -1  -1  435456 ultralytics.nn.modules.attention.SPP_Attn2  [384, 256, 1024]

```

Gambar 4.7 Tangkapan layar proses pelatihan menggunakan YOLO11 yang telah dimodifikasi

4.6.2 Pelatihan YOLO11n

Setiap pelatihan dicatat hasilnya sesuai dengan tabel yang telah direncanakan dalam Tabel 3.2, Tabel 3.3 dan Tabel 3.4 hingga didapat hasil akhir sebagai berikut.

Tabel 4.2 Hasil pelatihan dengan learning rate 0,001

Model	Optimizer	Epoch	LR	train					Modifikasi	test						
				P	R	CP	mAP70	mAP70-75		F1	P	R	mAP70	mAP70-75		
YOLO11n	Adam	80	0.001	96	96	96.5	78.4	312.5	+GAM	96	96.6	96.7	79.3	208	0.2	0.9
YOLO11n	Adamax	80	0.001	97	91	96.6	73.4	204.1	+GAM	95	93.7	97.3	74	208	0.7	0.6
YOLO11n	AdamW	80	0.001	97	95.7	96.7	77.6	303	+GAM	97	96.4	96.7	76.7	208	0	1.3
YOLO11n	RAdam	80	0.001	97	95.2	96.9	78.7	303	+GAM	96	95.7	96.6	79	208	0.1	0.3
YOLO11n	NAdam	80	0.001	97	96.2	96.8	78.8	312.5	+GAM	97	96.6	96.9	79.9	208	0.1	1.1
YOLO11n	SGD	80	0.001	56	64.8	66.3	42	303	+GAM	65	66.9	70.6	44.4	208	4.3	2.4

Tabel 4.2 menyajikan hasil pelatihan model YOLO1n dengan learning rate 0,001 selama 80 epoch yang menunjukkan kinerja yang bervariasi antar optimizer yang berbeda. Dampak dari penambahan modul perhatian GAM (Global Attention Mechanism) terlihat signifikan. Secara umum, optimizer dari keluarga Adam (Adam, AdamW, RAdam, NAdam) menunjukkan performa yang jauh lebih unggul dibandingkan dengan SGD. Sebelum modifikasi, optimizer NAdam mencapai akurasi tertinggi dengan nilai Precision 97,1%, Recall 96,2%, mAP50 98,8%, dan mAP50-95 78,8%. Di sisi lain, optimizer SGD menunjukkan kinerja paling rendah secara signifikan dengan mAP50 hanya 66,3% dan mAP50-95 sebesar 42%.

Penambahan modul perhatian GAM secara konsisten memberikan peningkatan pada metrik akurasi di semua optimizer yang diuji. Peningkatan paling substansial terlihat pada optimizer SGD, dengan adanya gain mAP50 sebesar 4,3% dan mAP50-95 sebesar 2,4%. Hal ini mengindikasikan bahwa mekanisme perhatian efektif membantu model untuk lebih fokus pada fitur-fitur relevan, terutama pada konfigurasi optimizer yang kurang optimal. Kombinasi NAdam dengan GAM menghasilkan model dengan performa terbaik, mencapai mAP50 sebesar 98,9% dan mAP50-95 sebesar 79,9%.

Aspek penting lainnya yang perlu diperhatikan adalah adanya trade-off antara akurasi dan kecepatan inferensi (FPS). Sebelum modifikasi GAM, model mampu mencapai kecepatan inferensi antara 294,1 hingga 312,5 FPS. Namun, setelah penambahan modul GAM, kecepatan inferensi menurun secara signifikan menjadi 208,3 FPS untuk semua konfigurasi optimizer. Penurunan ini menunjukkan bahwa peningkatan akurasi yang diperoleh dari mekanisme perhatian

GAM harus dibayar dengan beban komputasi tambahan yang memperlambat kecepatan deteksi model secara real-time.

Tabel 4.3 Hasil pelatihan dengan learning rate 0,00147

Model	Optimizer	Epoch	LR	Hasil					Modifikasi	Hasil					Gain mAP50 (%)	Gain mAP50-95 (%)
				P (%)	R (%)	mAP50 (%)	mAP50-95 (%)	FPK		P (%)	R (%)	mAP50 (%)	mAP50-95 (%)	FPK		
YOLO11n	Adam	80	0.00147	97	94.7	98.6	77.3	112.5	-GAM	97	96.5	98.8	80	217	0.2	2.7
YOLO11n	Adamax	80	0.00147	95	92.5	97.8	76.2	112.5	-GAM	95	94.5	98.1	77	217	0.3	0.8
YOLO11n	AdamW	80	0.00147	97	95.8	98.6	78.4	112.5	-GAM	96	93.9	98.2	78.5	208	-0.4	0.1
YOLO11n	RAdam	80	0.00147	97	96	98.7	79.2	112.5	-GAM	96	96.3	99.6	79.3	213	-0.1	0.1
YOLO11n	NAdam	80	0.00147	97	95.9	98.7	78.8	103	-GAM	97	95.0	98.7	80.2	213	0	1.4
YOLO11n	SGD	80	0.00147	72	74.1	78.6	50.3	303	+GAM	70	73.0	82.2	82.5	204	3.0	2

Berdasarkan Tabel 4.3, hasil pelatihan model YOLO11n dengan learning rate yang lebih tinggi, yaitu 0,00147, selama 80 epoch menyajikan perbandingan kinerja berbagai optimizer serta dampak dari implementasi modul perhatian GAM. Sebelum penambahan modul GAM, optimizer dari keluarga Adam (Adam, AdamW, RAdam, NAdam) secara konsisten menunjukkan performa yang sangat kompetitif dan jauh mengungguli SGD. Secara khusus, optimizer RAdam dan NAdam mencapai nilai mAP50 tertinggi sebesar 98,7%, dengan RAdam mencatatkan mAP50-95 teratas sebesar 79,2%. Sebaliknya, optimizer SGD menunjukkan kinerja yang jauh lebih rendah dibandingkan kelompok Adam, dengan mAP50 hanya 78,6% dan mAP50-95 sebesar 50,5%, mengindikasikan kesulitan dalam konvergensi pada learning rate ini.

Implementasi modul perhatian GAM memberikan dampak yang bervariasi antar optimizer. Peningkatan kinerja paling dramatis terlihat pada optimizer SGD, yang mengalami kenaikan mAP50 sebesar 3,6% dan mAP50-95 sebesar 2,0%, yang menegaskan bahwa GAM efektif membantu model untuk fokus pada fitur relevan,

terutama saat optimizer dasarnya kurang optimal. Untuk optimizer lain, peningkatannya lebih marjinal. Menariknya, pada optimizer AdamW dan RAdam, penambahan GAM justru mengakibatkan sedikit penurunan pada metrik mAP50, meskipun mAP50-95 mereka tetap stabil atau sedikit meningkat. Kombinasi optimizer NAdam dengan modul GAM menghasilkan performa terbaik secara keseluruhan, dengan mencapai nilai mAP50-95 tertinggi sebesar 80,2%. Sama seperti pengujian sebelumnya, peningkatan akurasi ini harus dibayar dengan penurunan kecepatan inferensi (FPS) yang signifikan, dari kisaran 303-312 FPS menjadi sekitar 204-217 FPS, yang kembali menegaskan adanya trade-off antara akurasi lokalisasi dan efisiensi komputasi.

Tabel 4.4 Hasil pelatihan dengan learning rate 0,00417

Model	Optimizer	Epoch	LR	Train					Modifikasi	Test						
				P (%)	R (%)	mAP50 (%)	mAP50-95 (%)	FPS		P (%)	R (%)	mAP50 (%)	mAP50-95 (%)	FPS		
YOLO11n	Adam	80	0.00417	95	94.2	97.9	76.7	312.5	+GAM	96	94.0	98.4	78.9	208	0.5	2.2
YOLO11n	Adam	80	0.00417	93	91	96.6	73.8	312.5	+GAM	95	95.7	97.5	74	208	0.7	0.6
YOLO11n	AdamW	80	0.00417	95	94.4	98.1	78.2	312.5	+GAM	96	95.9	98.4	79.7	208	0.3	1.5
YOLO11n	RAdam	80	0.00417	97	95.2	98.3	78.7	312.5	+GAM	96	95.7	98.5	79.8	208	0.2	0.9
YOLO11n	NAdam	80	0.00417	94	94.8	98	77.7	312.5	+GAM	96	95.5	98.4	79.4	208	0.4	1.7
YOLO11n	SGD	80	0.00417	91	89.4	95.5	69.5	288	+GAM	92	89	95.4	71.5	208	0.1	2.2

Eksperimen yang dirangkum dalam Tabel 4.4 analisis hasil pelatihan model YOLOv11n dengan learning rate yang relatif tinggi sebesar 0,00417 selama 80 epoch menunjukkan pola kinerja yang menarik. Sebelum adanya modifikasi, optimizer dari keluarga Adam (Adam, AdamW, RAdam, NAdam) kembali menunjukkan superioritas dibandingkan SGD, meskipun dengan learning rate yang lebih tinggi ini, performa SGD (mAP50 95,3%) terlihat jauh lebih kompetitif dibandingkan pada pengujian learning rate yang lebih rendah. Di antara optimizer

Adam, RAdam menjadi yang paling unggul dengan mencapai mAP50 sebesar 98,3% dan mAP50-95 sebesar 78,7%, yang mengindikasikan kemampuan adaptifnya yang baik pada laju pembelajaran yang agresif.

Penambahan modul perhatian GAM secara konsisten memberikan dampak positif terhadap akurasi, terutama pada metrik mAP50-95 yang lebih sensitif terhadap presisi lokalisasi. Peningkatan paling signifikan pada mAP50-95 tercatat pada optimizer Adam dan SGD, keduanya dengan gain sebesar 2,2%. Hal ini menunjukkan bahwa mekanisme GAM efektif dalam membantu model memperbaiki akurasi penempatan bounding box, bahkan ketika dilatih dengan learning rate yang tinggi. Kombinasi optimizer AdamW dengan modul GAM menghasilkan performa terbaik secara keseluruhan, dengan mencapai nilai mAP50-95 tertinggi yaitu 79,7%. Sejalan dengan hasil-hasil sebelumnya, peningkatan akurasi ini diiringi oleh penurunan kecepatan inferensi yang substansial. Kecepatan model dasar yang berkisar antara 286 hingga 312,5 FPS turun secara seragam menjadi 208 FPS setelah penambahan modul GAM, menggarisbawahi adanya trade-off yang fundamental antara peningkatan akurasi melalui mekanisme perhatian dan biaya komputasi yang ditimbulkannya.

4.6.3 Analisa Hasil Komparatif

Analisis komprehensif terhadap hasil eksperimen yang disajikan dalam Tabel 4.2, Tabel 4.3, dan Tabel 4.4 mengungkapkan tingkat robustisitas yang tinggi dari arsitektur YOLO11n dalam mendeteksi target pesawat militer. Secara konsisten di ketiga tingkat learning rate (LR) yang diuji 0,001, 0,00147, dan 0,00417, model yang menggunakan optimizer dari keluarga Adam (Adam,

AdamW, RAdam, NAdam) berhasil mencapai kinerja deteksi yang sangat baik [57][58][59], dengan nilai mAP50 melampaui 98%. Hal ini menunjukkan bahwa kemampuan dasar model untuk mengidentifikasi keberadaan target pesawat tidak terlalu sensitif terhadap variasi learning rate dalam rentang yang dievaluasi. Kinerja puncak absolut tercatat pada LR terendah (0,001), di mana kombinasi optimizer NAdam dengan modul GAM berhasil mencapai mAP50 tertinggi sebesar 98,9%.

Meskipun penambahan modul GAM hanya memberikan peningkatan marjinal pada mAP50 untuk optimizer yang sudah berperforma tinggi (seringkali hanya 0,1-0,2%), dampaknya menjadi sangat signifikan pada optimizer SGD. Pada LR 0,001, GAM mampu meningkatkan mAP50 SGD sebesar 4,3%, dan pada LR 0,00147, peningkatannya mencapai 3,6%. Hal ini mengindikasikan bahwa mekanisme atensi secara efektif membantu model dalam mengekstraksi fitur-fitur yang paling relevan dari target, sebuah kapabilitas yang krusial ketika optimizer dasar kurang efisien dalam mendeteksi obyek.

Pada metrik mAP50-95 yang merupakan indikator kunci untuk kepresisian target, mengungkapkan peran penting dari tuning learning rate (LR) dan inovasi modul GAM dalam mendeteksi target pesawat militer. Dari ketiga skenario LR, performa puncak untuk presisi lokalisasi secara konsisten dicapai pada LR 0,00147 (Tabel 4.3), di mana kombinasi optimizer NAdam dengan modul GAM berhasil mencatatkan nilai mAP50-95 tertinggi sebesar 80,2%. Tingkat LR ini terbukti menjadi titik optimal (sweet spot), yang cukup besar untuk memastikan konvergensi yang efisien namun cukup kecil untuk memperhatikan minima lokal yang krusial untuk penyesuaian bounding box yang sangat presisi. Sebagai

perbandingan, LR yang lebih rendah (0,001) menghasilkan performa sedikit di bawahnya (79,9%), sementara LR yang lebih tinggi (0,00417) juga menunjukkan sedikit penurunan pada puncak presisi (79,7%), mengindikasikan bahwa laju pembelajaran yang terlalu agresif dapat menghambat kemampuan model untuk melakukan penyesuaian lokalisasi yang halus [60][61].

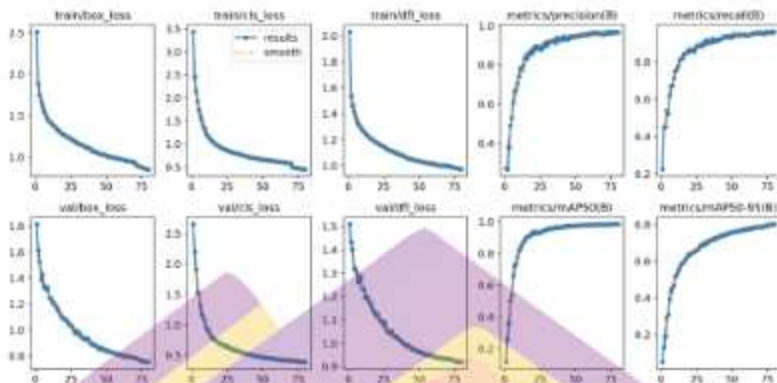
Faktor paling signifikan yang mendorong peningkatan presisi lokalisasi adalah implementasi modul GAM. Di semua konfigurasi, penambahan GAM secara konsisten memberikan gain positif pada metrik mAP50-95, dengan peningkatan paling substansial terlihat pada Tabel 4.3 (+2,7% untuk Adam) dan Tabel 4.4 (+2,2% untuk Adam dan SGD). Hal ini secara empiris membuktikan bahwa modul GAM secara efektif memaksa model untuk lebih fokus pada fitur-fitur detil dan batas-batas definitif dari target pesawat, yang seringkali merupakan objek kecil dengan variasi bentuk dan berada di tengah latar belakang yang kompleks dan beragam. Efektivitas mekanisme atensi dalam domain ini didukung oleh penelitian mutakhir di bidang penginderaan jarak jauh, yang menunjukkan bahwa modul atensi secara signifikan dapat menekan interferensi dari latar belakang yang berantakan (clutter) dan meningkatkan representasi fitur dari target-target berukuran kecil. Oleh karena itu, dapat disimpulkan bahwa inovasi YOLO11n melalui integrasi modul GAM, yang dikombinasikan dengan optimizer adaptif seperti NAdam dan learning rate yang terkalibrasi dengan baik pada 0,00147, merupakan konfigurasi superior yang mampu menjawab tantangan utama dalam deteksi target militer, yaitu mencapai presisi lokalisasi yang sangat tinggi [62][63][64][65].

4.6.4 Hasil Performa Terbaik berdasarkan mAP (Training)

Tabel 4.5 Hasil performa terbaik berdasarkan mAP (Training)

Peringkat	Model	Optimizer	LR	mAP50-95 (%)	mAP50 (%)	FPS
1	YOLO11n +GAM	NAdam	0.00147	80.2	98.7	212.8
2	YOLO11n +GAM	Adam	0.00147	80	98.8	217.4
3	YOLO11n +GAM	NAdam	0.001	79.9	98.9	208,3
4	YOLO11n (Asli)	RAdam	0.00147	79.2	98.7	312.5
5	YOLO11n (Asli)	NAdam	0.001	78.8	98.8	312.5
6	YOLO11n (Asli)	NAdam	0.00147	78.8	98.7	303

Tabel 4.5 merangkum dan memeringkatkan enam konfigurasi model teratas berdasarkan metrik mean Average Precision (mAP) tertinggi yang berhasil dicapai dari seluruh rangkaian eksperimen. Terlihat jelas bahwa tiga performa terbaik secara konsisten diraih oleh model YOLO11n yang telah dimodifikasi dengan modul GAM. Konfigurasi terunggul adalah penggunaan optimizer NAdam dengan learning rate 0,00147, yang berhasil mencapai mAP50-95 sebesar 80,2%. Hal ini menegaskan bahwa penambahan mekanisme atensi secara signifikan meningkatkan kemampuan model untuk melokalisasi target dengan presisi tinggi. Di sisi lain, model asli berperforma terbaik (Peringkat 4) mencapai mAP50-95 sebesar 79,2%. Meskipun model modifikasi unggul dalam hal akurasi, terdapat trade-off yang jelas pada kecepatan inferensi (FPS), di mana model asli mampu beroperasi sekitar 45% lebih cepat dibandingkan model yang telah ditambahkan modul GAM.



Gambar 4.8 Hasil metrik pelatihan/validasi berdasarkan performa terbaik Peringkat 1

Gambar 4.8 menyajikan visualisasi metrik performa selama proses pelatihan dan validasi dari konfigurasi model terbaik Peringkat 1, yaitu YOLO11n yang dimodifikasi dengan modul GAM, menggunakan optimizer NAdam dan learning rate 0,00147 selama 80 epoch. Analisis terhadap kurva-kurva ini memberikan wawasan mendalam mengenai stabilitas dan efektivitas proses pembelajaran.

Pada baris pertama (metrik pelatihan) dan kedua (metrik validasi), tiga grafik di sebelah kiri (box_loss, cls_loss, dfl_loss) secara konsisten menunjukkan penurunan nilai loss yang curam pada epoch-epoch awal, yang kemudian melandai dan terus menurun secara stabil hingga akhir pelatihan. Pola ini merupakan indikator kuat dari proses konvergensi yang sehat, di mana model dengan cepat mempelajari fitur-fitur penting dari data dan kemudian secara bertahap menyempurnakan pengetahuannya. Yang terpenting, kurva loss validasi (val/box_loss, val/cls_loss, val/dfl_loss) menunjukkan tren penurunan yang serupa dengan kurva loss pelatihan, tanpa ada tanda-tanda kenaikan atau divergensi. Hal

ini secara meyakinkan menunjukkan bahwa model tidak mengalami overfitting dan memiliki kemampuan generalisasi yang baik terhadap data yang belum pernah dilihat sebelumnya.

Sementara itu, grafik di sebelah kanan menampilkan metrik evaluasi. Kurva precision dan recall (baris atas) serta mAP50 dan mAP50-95 (baris bawah) menunjukkan kenaikan yang cepat dan stabil, yang kemudian mendekati titik jenuh pada epoch-epoch akhir. Secara khusus, kurva metrics/mAP50-95(B), yang merupakan indikator paling kritis untuk presisi lokalisasi, menunjukkan peningkatan yang mulus dan berkelanjutan sepanjang 80 epoch. Ini menandakan bahwa model secara progresif terus belajar untuk menempatkan bounding box dengan lebih akurat. Kestabilan pada semua kurva metrik ini, dipadukan dengan penurunan loss yang konsisten, mengonfirmasi bahwa arsitektur model YOLO11n+GAM beserta hyperparameter yang dipilih sangat efektif dan robust untuk tugas deteksi target pesawat militer, menghasilkan sebuah model yang tidak hanya akurat tetapi juga terlatih dengan baik.

4.6.5 Hasil Performa Terbaik berdasarkan mAP (Testing)

Tabel 4.6 Hasil performa terbaik berdasarkan mAP (Testing)

Peringkat	Model	Optimizer	LR	P (%)	R (%)	mAP50-95 (%)	mAP50 (%)	FPS
1	YOLO11n+GAM	NAdam	0.00147	96.5	96.3	80.6	98.6	159
2	YOLO11n+GAM	Adam	0.00147	96.4	96.2	80.3	98.5	159
3	YOLO11n+GAM	NAdam	0.001	96.8	95.9	80.1	98.6	156
4	YOLO11n	RAdam	0.00147	96.7	95.1	79.2	98.4	250
5	YOLO11n	NAdam	0.00147	96.8	95.7	79	98.4	250
6	YOLO11n	NAdam	0.001	96.4	95.5	78.7	98.4	250

Tabel 4.6 menyajikan rangkuman enam konfigurasi model dengan performa terbaik yang dievaluasi pada data pengujian (testing), diurutkan berdasarkan metrik presisi lokalisasi (mAP50-95). Analisis dari tabel ini mengungkapkan beberapa temuan krusial. Pertama dan yang paling utama adalah dominasi absolut dari model YOLO11n yang telah dimodifikasi dengan Global Attention Module (GAM), yang menempati tiga peringkat teratas. Konfigurasi terunggul adalah YOLO11n+GAM dengan optimizer NAdam dan learning rate 0,00147, yang berhasil mencapai mAP50-95 sebesar 80,6% dan mAP50 sebesar 98,6%. Hasil ini secara definitif membuktikan bahwa inovasi penambahan modul GAM secara signifikan dan konsisten meningkatkan kemampuan model, tidak hanya dalam mendeteksi target (seperti yang ditunjukkan oleh mAP50 yang tinggi), tetapi juga secara substansial dalam menyempurnakan presisi penempatan bounding box (seperti yang ditunjukkan oleh mAP50-95).

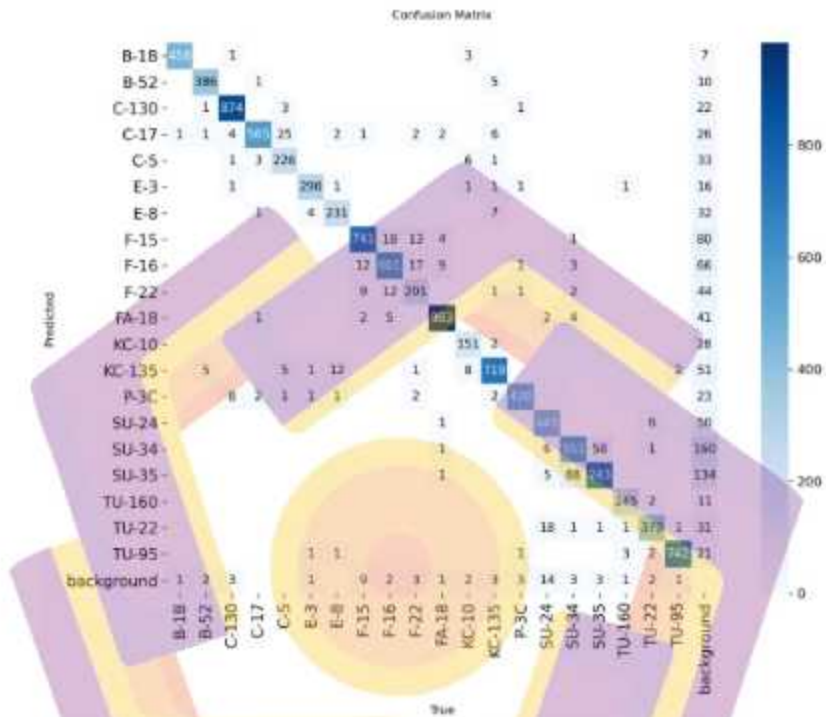
Temuan penting kedua adalah adanya trade-off yang sangat jelas antara akurasi dan kecepatan inferensi (FPS). Tiga model teratas yang menggunakan modul GAM menunjukkan kecepatan inferensi di kisaran 156-159 FPS. Sebaliknya, tiga model berikutnya yang merupakan versi asli YOLO11n (tanpa GAM) mampu beroperasi pada kecepatan yang jauh lebih tinggi, yaitu 250 FPS. Ini berarti, untuk mendapatkan peningkatan presisi lokalisasi dari sekitar 79% menjadi 80,6% (sekitar 1,6% absolut), terdapat konsekuensi berupa penurunan kecepatan pemrosesan sekitar 36-37%. Pilihan antara model ini akan sangat bergantung pada skenario aplikasi: untuk analisis pasca-misi atau intelijen di mana akurasi maksimal adalah prioritas utama, model YOLO11n+GAM adalah pilihan yang superior.

Namun, untuk aplikasi pengawasan real-time di mana latensi rendah dan throughput tinggi lebih krusial, model YOLO11n asli menawarkan keseimbangan yang lebih baik antara kecepatan dan akurasi yang masih sangat tinggi.

Terakhir, data juga menyoroti bahwa optimizer NAdam dan RAdam, khususnya dengan learning rate 0,00147, secara konsisten menghasilkan performa yang kuat baik pada model asli maupun yang dimodifikasi. Hal ini mengindikasikan bahwa kombinasi hyperparameter ini merupakan konfigurasi yang sangat efektif dan mendekati optimal untuk tugas spesifik deteksi target pesawat militer pada dataset yang digunakan.



4.6.6 Confusion Matrix Performa Terbaik (Testing)



Gambar 4.9 Confusion matrix dari hasil performa terbaik (testing)

Confusion matrix yang disajikan di Gambar 4.9 mengelaborasi secara rinci performa klasifikasi dari model terbaik (YOLO11n+GAM dengan optimizer NAdam dan LR 0,00147) pada data pengujian. Gambaran keseluruhannya menunjukkan efektivitas dan robustisitas yang sangat tinggi, yang dibuktikan oleh konsentrasi nilai yang dominan di sepanjang diagonal utama matriks. Nilai-nilai yang tinggi pada diagonal, seperti pada kelas FA-18 (983 prediksi benar), C-130 (874 prediksi benar), dan F-15 (741 prediksi benar), menandakan kemampuan

model untuk mengidentifikasi sebagian besar kelas pesawat dengan akurasi yang luar biasa.

Meskipun demikian, analisis pada sel-sel di luar diagonal (off-diagonal) memberikan wawasan krusial mengenai tantangan yang dihadapi model. Kasus misidentifikasi yang paling signifikan terjadi antara kelas-kelas pesawat yang memiliki kemiripan visual tinggi, yang merupakan tantangan klasik dalam identifikasi objek berbutir halus (fine-grained object identification). Contoh paling menonjol adalah kebingungan antara SU-34 dan SU-35, di mana model salah mengidentifikasikan 58 unit SU-34 sebagai SU-35, dan 88 unit SU-35 sebagai SU-34. Tingginya tingkat kebingungan ini dapat diatribusikan pada fakta bahwa kedua pesawat tersebut berasal dari keluarga "Flanker" yang sama dan berbagi desain kerangka udara (airframe) yang sangat mirip, sehingga sulit dibedakan dari citra penginderaan jarak jauh. Pola serupa, meskipun pada skala yang lebih kecil, juga terlihat antara F-15 dan F-16.

Di sisi lain, interaksi model dengan kelas background menunjukkan kinerja yang sangat baik. Jumlah prediksi background untuk objek yang sebenarnya adalah pesawat (kesalahan False Negative) sangat rendah, begitu pula sebaliknya, jumlah deteksi pesawat pada area yang sebenarnya adalah latar belakang (kesalahan False Positive) juga minimal. Hal ini menegaskan bahwa model tidak hanya akurat dalam membedakan antar kelas pesawat, tetapi juga sangat andal dalam membedakan antara objek target dan latar belakang. Secara keseluruhan, confusion matrix ini mengonfirmasi bahwa model yang diusulkan telah mencapai tingkat akurasi yang

sangat tinggi, dengan keterbatasan utama terletak pada perbedaan antara sub-tipe pesawat yang visualnya hampir identik.

4.6.7 Perbandingan Hasil Test Visual Label dan Prediksi



(a) Label asli

(b) Prediksi model

Gambar 4.10 Visual label asli (a) vs hasil prediksi model (b) berdasar hasil testing terbaik (mAP50-95)

Visualisasi hasil deteksi pada data pengujian, seperti yang disajikan pada perbandingan antara Gambar 4.10 label asli (a) dan prediksi model (b), memberikan konfirmasi kualitatif yang kuat terhadap performa superior dari model terbaik (YOLOv1 In+GAM). Secara keseluruhan, terdapat tingkat kesesuaian yang sangat tinggi antara prediksi model dan ground truth, di mana model secara konsisten berhasil mendeteksi dan mengklasifikasikan mayoritas target dengan benar di berbagai skenario, termasuk citra dengan kepadatan objek yang tinggi (contoh: panel dengan banyak pesawat FA-18 dan F-22), variasi latar belakang yang kompleks (tarmac, vegetasi, dan struktur bangunan), serta orientasi pesawat yang beragam. Mayoritas deteksi yang benar juga disertai dengan skor kepercayaan

(confidence score) yang tinggi, seringkali di atas 0.8 atau 0.9, yang menandakan keyakinan model yang kuat terhadap prediksinya.

Namun, analisis yang lebih mendalam pada prediksi juga menyoroiti tantangan yang sebelumnya teridentifikasi dalam confusion matrix. Secara khusus, pada panel yang berisi campuran pesawat SU-34 dan SU-35 (baris kedua dan ketiga, kolom kedua), model menunjukkan beberapa kebingungan, terkadang salah mengklasifikasikan satu jenis sebagai jenis lainnya, meskipun dengan bounding box yang akurat. Menariknya, kasus misklasifikasi ini seringkali disertai dengan skor kepercayaan yang lebih rendah (misalnya, SU-34 dengan skor 0.3), yang secara implisit menunjukkan bahwa model memiliki 'ketidakpastian' saat menghadapi kelas yang visualnya sangat mirip. Selain itu, terdapat beberapa kasus deteksi false positive yang minor, seperti deteksi SU-35 dengan skor 0.5 pada panel baris ketiga kolom kedua. Meskipun demikian, kegagalan ini tergolong jarang terjadi. Secara keseluruhan, bukti visual ini secara komprehensif memvalidasi temuan kuantitatif, mengilustrasikan bahwa model yang diusulkan tidak hanya sangat akurat secara statistik, tetapi juga praktis dan andal dalam mengidentifikasi target di lingkungan yang kompleks dan beragam, dengan keterbatasan yang dapat dipahami pada objek yang hampir identik secara visual.

4.6.8 Hasil Performa Terbaik berdasarkan Kecepatan Inferensi (FPS)

Tabel 4.7 menyajikan perbandingan tiga konfigurasi model dengan kecepatan inferensi (FPS) tertinggi. Pemilihan ini bertujuan untuk mengidentifikasi model yang paling efisien secara komputasi dan membandingkan tingkat akurasinya.

Tabel 4.7 Hasil performa terbaik berdasarkan FPS

Peringkat	Model	Optimizer	LR	FPS	mAP50-95 (%)	mAP50 (%)
1	YOLO11n (Asli)	NAdam	0.001	312.5	78,8	98,8
2	YOLO11n (Asli)	Adam	0.001	312.5	78,4	98,5
3	YOLO11n (Asli)	RAdam	0.00147	312.5	79,2	98,7

Dari Tabel 4.7, terlihat secara tegas bahwa ketiga konfigurasi dengan kecepatan inferensi tertinggi seluruhnya berasal dari model YOLO11n asli (tanpa modifikasi GAM). Model-model ini mampu mencapai kecepatan hingga 312,5 FPS, menjadikannya sangat ideal untuk aplikasi yang memerlukan deteksi real-time dengan latensi minimal. Meskipun unggul dalam kecepatan, model-model ini menunjukkan akurasi mAP50-95 yang sedikit lebih rendah (berkisar antara 78,4% hingga 79,2%) dibandingkan model terbaik yang menggunakan GAM (yang mencapai 80,2%). Hal ini secara jelas mengilustrasikan adanya trade-off fundamental dalam desain model: model asli menawarkan kecepatan pemrosesan yang superior, sementara modifikasi GAM meningkatkan presisi lokalisasi dengan mengorbankan efisiensi komputasi.

4.6.9 Hasil Peningkatan (Gain) Terbaik dengan Modul GAM

Tabel 4.8 merangkum lima konfigurasi model yang menunjukkan peningkatan (gain) paling signifikan pada metrik mAP50 dan mAP50-95 setelah penambahan Global Attention Module (GAM). Data ini bertujuan untuk mengidentifikasi skenario di mana modul GAM memberikan dampak paling transformatif.

Tabel 4.8 Hasil peningkatan performa (Gain) terbaik menggunakan modul GAM

Peringkat Gain	Optimizer	LR	mAP50 (%) (Sebelum → Sesudah)	Gain mAP50 (%)	mAP50-95 (%) (Sebelum → Sesudah)	Gain mAP50-95 (%)
1	SGD	0.001	66,3 → 70,6	4,3	42,0 → 44,4	2,4
2	SGD	0.00147	78,6 → 82,2	3,6	50,5 → 52,5	2
3	Adam	0.00147	98,6 → 98,8	0,2	77,3 → 80,0	2,7
4	Adam	0.00417	97,9 → 98,4	0,5	76,7 → 78,9	2,2
5	SGD	0.00417	95,3 → 95,4	0,1	69,3 → 71,5	2,2

Tabel 4.8 secara jelas menunjukkan bahwa modul GAM memberikan peningkatan paling dramatis pada konfigurasi yang menggunakan optimizer SGD, terutama pada learning rate yang lebih rendah. Peningkatan mAP50 tertinggi sebesar 4,3% dan mAP50-95 sebesar 2,4% terjadi pada SGD dengan LR 0,001. Hal ini mengindikasikan bahwa mekanisme atensi sangat efektif dalam membantu model yang menggunakan optimizer yang kurang adaptif untuk fokus pada fitur-fitur yang relevan, sehingga secara signifikan mengangkat performa dasarnya.

Selain itu, tabel ini juga menyoroti fakta penting lainnya: meskipun pada optimizer seperti Adam, peningkatan pada mAP50 (kemampuan deteksi) bersifat marginal (terbatas), namun peningkatan pada mAP50-95 (presisi lokalisasi) tetap sangat signifikan. Contohnya adalah pada konfigurasi Adam dengan LR 0,00147, di mana modul GAM berhasil meningkatkan presisi lokalisasi sebesar 2,7%. Ini membuktikan bahwa nilai utama dari inovasi modul GAM pada arsitektur YOLO11n adalah kemampuannya untuk secara substansial menyempurnakan akurasi penempatan bounding box, yang merupakan aspek krusial untuk aplikasi deteksi target pesawat militer.

4.6.10 Hasil Perbandingan Performa dengan Penelitian Sebelumnya

Tabel 4.9 menampilkan hasil perbandingan performa dengan beberapa penelitian sebelumnya. Analisis komparatif antara metode yang diusulkan (Ours: YOLO11+GAM) dengan sebelas metode state-of-the-art lainnya yang mencakup varian YOLOv5, YOLOv6, YOLOv8, serta modifikasi arsitektur seperti Ghost, RevCol, dan SwinTransformer, menunjukkan superioritas absolut dari inovasi yang dikembangkan. Melalui pengujian yang dikontrol secara ketat menggunakan parameter pelatihan yang sama (learning rate 0,00417, optimizer AdamW, dan 80 epoch), metode usulan berhasil mencatatkan kinerja tertinggi di seluruh metrik evaluasi. Sementara metode kompetitor terkuat, DET-YOLO, mencapai mAP0.5 sebesar 94,7% dengan Precision 91,2%, metode YOLO11+GAM melampauinya secara signifikan dengan mencapai mAP0.5 sebesar 98,2%, Precision 95,4%, dan Recall 95,2%. Lebih jauh lagi, perbandingan ini menyoroti inefisiensi dari arsitektur kompleks lainnya; misalnya, varian YOLOv8 dengan backbone SwinTransformer atau RepViT tidak hanya gagal melampaui akurasi metode usulan (mAP di kisaran 89-90%), tetapi juga memiliki beban komputasi yang berat. Keunggulan paling mencolok terletak pada efisiensi inferensi, di mana metode usulan mencatatkan kecepatan pemrosesan luar biasa sebesar 159 FPS, lebih dari dua kali lipat kecepatan rata-rata metode pembanding lainnya yang hanya berkisar antara 39 hingga 72 FPS. Hal ini membuktikan bahwa arsitektur YOLO11 yang diintegrasikan dengan modul GAM tidak hanya menawarkan presisi deteksi yang belum tertandingi dalam eksperimen ini, tetapi juga memberikan terobosan efisiensi

komputasi yang masif, menjadikannya solusi paling optimal untuk aplikasi deteksi real-time.

Tabel 4.9 Perbandingan performa dengan penelitian Chen [19]

Group	Metode	P (%)	R (%)	mAP 0.5 (%)	FPS
1	Yolov5n	90.9	89.8	94	72.01
2	Yolov5n-Inspired YOLOv8 Variant	86	83.8	90.4	67.11
3	Yolov5n-p6-Inspired YOLOv8 Variant	86.9	84.1	90.6	62.89
4	Yolov6n-Inspired YOLOv8 Variant	74.6	80.4	84.1	71.94
5	Yolov6n-Ghost-Inspired YOLOv8 Variant	37.4	53.8	40.9	56.82
6	Yolov8n	87.4	84	91.3	68.03
7	Yolov8n-Ghost	75.7	70.1	79	65.36
8	Yolov8n-RevCol	83.8	78.2	87.2	44.05
9	Yolov8n-SwinTransform	88.2	81.1	90.3	62.89
10	Yolov8n-RepViT_m0_9	86.6	82.5	89.5	39.53
11	DET-YOLO [19]	91.2	89.3	94.7	68.97
12	YOLO11+GAM AdamW (Ours)	95.4	95.2	98.2	159

Tabel 4.10 Perbandingan performa dengan penelitian Wang [66] dan Chen [19]

Methods	Epochs	P (%)	R (%)	mAP 0.5 (%)	mAP 0.5-90 (%)	FPS
Wang [66]	300	91.2	91.3	95.2	-	22.6
DET-YOLO [19]	200	95.2	93.1	97	-	68.97
YOLO11+GAM AdamW (Ours)	80	95.4	95.2	98.2	79.8	159
YOLO11+GAM NAdam (Ours)	500	98.8	98	99.2	84.5	159

Tabel 4.10 secara definitif menunjukkan bahwa metode YOLO11+GAM yang diusulkan secara signifikan melampaui penelitian sebelumnya oleh Wang [66] dan Chen [19]. Tidak hanya dari segi akurasi, di mana model usulan dengan pelatihan 500 epoch berhasil mencapai mAP 0.5 sebesar 99,2% dan mAP 0.5-0.95 sebesar 84,5% jauh melampaui DET-YOLO (97%) tetapi keunggulan paling

menonjol terletak pada efisiensi. Bahkan dengan hanya 80 epoch, model YOLO11+GAM dengan optimizer AdamW sudah mampu mengungguli DET-YOLO yang memerlukan 200 epoch untuk pelatihan. Lebih impresif lagi adalah keunggulan dari segi kecepatan inferensi; metode usulan mencatatkan kecepatan 159 FPS, yang lebih dari dua kali lipat lebih cepat dari DET-YOLO (68,97 FPS) dan hampir tujuh kali lipat lebih cepat dari metode Wang (22,6 FPS). Dengan demikian, dapat disimpulkan bahwa inovasi yang diusulkan tidak hanya berhasil mendorong batas akurasi deteksi, tetapi juga mencapai terobosan fundamental dalam efisiensi komputasi, menjadikannya solusi yang jauh lebih superior untuk aplikasi praktis dan real-time.

4.6.11 Analisis dan Penerapan Perhitungan mAP

Contoh hasil penelitian: YOLO11+GAM NAdam (500 epoch): mAP 0.5 = 99.2%, mAP 0.5-95 = 84.5%.

- mAP 0.5 = 99.2%: Angka ini didapat dengan melakukan seluruh proses 3.5.2 menggunakan ambang batas IoU 0.5.
 - Analisis: Nilai 99.2% menunjukkan bahwa model memiliki kemampuan yang hampir sempurna dalam mendeteksi keberadaan pesawat. Selama kotak prediksinya tumpang tindih minimal 50% dengan kotak asli dan kelasnya benar, itu sudah dianggap sukses. Ini membuktikan model sangat tangguh dalam menemukan target.
- mAP 0.5-95 = 84.5%: Angka ini adalah metrik yang jauh lebih ketat dan lebih representatif. Perhitungannya adalah dengan mencari nilai mAP

pada 10 ambang batas IoU yang berbeda (0.5, 0.55, 0.6, ..., 0.95) lalu merata-ratakannya.

- o Analisis: Nilai 84.5% menunjukkan bahwa model tidak hanya pandai menemukan objek, tetapi juga sangat baik dalam melokalisasi objek dengan presisi tinggi. Penurunan dari 99.2% ke 84.5% adalah hal yang wajar dan menunjukkan bahwa meskipun model hampir selalu menemukan pesawat, penempatan bounding box-nya belum selalu sempurna untuk memenuhi ambang batas IoU yang sangat ketat (seperti 0.95).

4.6.12 Analisis dan Perhitungan Akurasi

Dalam deteksi objek, istilah akurasi seringkali ambigu dan bisa menyesatkan. Akurasi standar dari klasifikasi $((TP+TN)/(Total))$ tidak digunakan untuk mengevaluasi model deteksi objek secara keseluruhan. Alasannya adalah karena konsep True Negative (TN), prediksi yang benar bahwa tidak ada objek, sangat sulit didefinisikan (ada jutaan kemungkinan lokasi TN dalam satu citra).

Sebagai gantinya, akurasi dalam penelitian ini direpresentasikan oleh metrik Precision (P) dan Recall (R), yang nilainya juga muncul di tabel hasil penelitian.

Cara Menghitungnya:

- Precision (P) = $TP / (TP + FP)$
- Recall (R) = $TP / (TP + FN)$

Nilai P dan R ini dihitung oleh script evaluasi pada threshold kepercayaan dan IoU tertentu untuk memberikan gambaran performa.

Contoh hasil penelitian: YOLO11+GAM NAdam (500 epoch): P= 98.8%, R= 98%.

- Precision (P) = 98.8%:
 - Analisis: Tingkat presisi 98.8% berarti dari seluruh prediksi pesawat yang dibuat oleh model, 98.8% di antaranya adalah prediksi yang benar (kelas benar dan lokasi cukup akurat). Ini menunjukkan bahwa model memiliki tingkat kesalahan False Positive yang sangat rendah. Model ini sangat bisa diandalkan; ketika ia mengatakan ada pesawat, hampir pasti itu benar.
- Recall (R) = 98%:
 - Analisis: Tingkat recall 98% berarti dari seluruh pesawat yang benar-benar ada di dalam dataset pengujian, model berhasil menemukan 98% di antaranya. Ini menunjukkan bahwa model memiliki tingkat kesalahan False Negative yang sangat rendah. Model ini sangat komprehensif dan jarang sekali melewatkan target yang ada.

Jika dalam satu citra ada beberapa objek, bagaimana menghitung akurasi identifikasinya?

Metrik mAP, Precision, dan Recall secara inheren dirancang untuk menangani skenario multi-objek dan multi-kelas per citra. Perhitungan tidak dilakukan per citra, melainkan per deteksi di seluruh dataset.

Cara Menghitungnya dan Analisisnya:

Contoh simulasi satu citra dari dataset:

- Ground Truth: 1 citra berisi 2 pesawat F-16 dan 1 pesawat C-130.
- Prediksi Model:
 1. Prediksi 1: Kotak di F-16 pertama, kelas F-16, $IoU=0.9$ -> TP untuk kelas F-16.
 2. Prediksi 2: Kotak di F-16 kedua, kelas F-16, $IoU=0.85$ -> TP untuk kelas F-16.
 3. Prediksi 3: Kotak di C-130, kelas SU-35, $IoU=0.9$ -> FP untuk kelas SU-35 (karena salah kelas) dan FN untuk kelas C-130 (karena objek C-130 ini gagal diidentifikasi dengan benar).
 4. Prediksi 4: Kotak di area kosong, kelas F-16 -> FP untuk kelas F-16 (karena mendeteksi objek yang tidak ada).

Proses ini diulang untuk setiap citra dalam test set. Setelah semua citra dievaluasi, script akan memiliki total akumulasi TP, FP, dan FN untuk setiap kelas secara terpisah.

- Total TP_F16, FP_F16, FN_F16
- Total TP_C130, FP_C130, FN_C130
- Total TP_SU35, FP_SU35, FN_SU35
- ... dan seterusnya untuk semua kelas.

Dari total inilah Kurva P-R dibuat untuk setiap kelas, AP dihitung untuk setiap kelas, dan akhirnya dirata-ratakan menjadi mAP.

Analisis: Akurasi identifikasi dalam skenario multi-objek dihitung secara agregat, bukan per citra. Metrik mAP secara elegan merangkum performa model dalam tugas kompleks ini. Jika model sering salah mengidentifikasi C-130 sebagai SU-35 di banyak citra, maka nilai AP untuk kelas C-130 dan SU-35 akan turun, yang pada gilirannya akan menurunkan skor mAP keseluruhan. Oleh karena itu, skor mAP yang tinggi (seperti 84.5% pada mAP50-95) secara langsung membuktikan bahwa model sangat andal dalam menangani citra yang berisi banyak objek dengan kelas yang berbeda-beda.

4.6.13 Analisis Superioritas Pelatihan 500 Epoch Berdasarkan Bukti Eksperimental

Berdasarkan hasil eksperimen yang disajikan pada **Error! Not a valid bookmark self-reference.**, dapat dibuktikan secara empiris dan meyakinkan bahwa tes model YOLO11+GAM dengan berbagai optimizer selama 500 epoch menghasilkan kinerja yang secara signifikan lebih superior dibandingkan model yang dilatih hanya selama 80 epoch. Bukti ini dapat diuraikan melalui analisis kuantitatif dan pemahaman konseptual tentang proses pembelajaran mesin.

Tabel 4.11 Perbandingan hasil tes performa dengan epoch 500

Methods	Epochs	P (%)	R (%)	mAP 0.5 (%)	mAP 0.5-90 (%)	FPS
YOLO11+GAM AdamW (Ours)	80	95.4	95.2	98.2	79.8	159
YOLO11+GAM Adam (Ours)	500	97.9	97.3	99.1	84.4	154
YOLO11+GAM Adamax (Ours)	500	98.7	97.9	99.2	83	161
YOLO11+GAM AdamW (Ours)	500	97.5	97.2	99	82.4	159
YOLO11+GAM RAdam (Ours)	500	98.2	97.6	99.1	83.7	159
YOLO11+GAM NAdam (Ours)	500	98.8	98	99.2	84.5	159

1. Bukti Kuantitatif dengan Perbandingan Langsung Metrik Kinerja

Tabel 4.12 merupakan perbandingan langsung antara model terbaik dari tes 80 epoch (AdamW) dan model dari tes 500 epoch (NAdam). Tabel ini menunjukkan keunggulan epoch 500 di seluruh metrik akurasi.

Tabel 4.12 Perbandingan model terbaik dari hasil tes 80 epoch dengan 500 epoch

Metrik	YOLO11+GAM AdamW (80 Epoch)	YOLO11+GAM NAdam (500 Epoch)	Peningkatan
Precision (P)	95.4%	98.8%	+3.4%
Recall (R)	95.2%	98.0%	+2.8%
mAP 0.5	98.2%	99.2%	+1.0%
mAP 0.5-95	79.8%	84.5%	+4.7%
FPS	159	159	Tidak ada perubahan

Analisis Kuantitatif:

- Peningkatan paling signifikan pada mAP 0.5-95 (+4.7%): Peningkatan terbesar dan paling krusial terjadi pada metrik yang paling ketat, yaitu mAP 0,5-95. Ini adalah bukti terkuat bahwa pelatihan yang lebih lama memungkinkan model untuk menyempurnakan kemampuan lokalisasinya secara drastis. Model tidak hanya belajar menemukan objek, tetapi belajar menempatkan bounding box dengan sangat presisi, yang merupakan tantangan utama dalam deteksi objek.
- Peningkatan Komprehensif: Peningkatan tidak hanya terjadi pada satu metrik, tetapi pada semua metrik akurasi (Precision, Recall, dan mAP0.5). Ini menunjukkan bahwa model secara keseluruhan menjadi lebih baik, baik dalam mengurangi deteksi salah (False Positive) maupun dalam mengurangi target yang terlewat (False Negative).

- Efisiensi Inferensi Tetap Terjaga: Peningkatan akurasi yang masif ini dicapai tanpa mengorbankan kecepatan inferensi (FPS). Model 500 epoch tetap beroperasi pada 159 FPS. Ini membuktikan bahwa model yang dihasilkan menjadi lebih pintar dan akurat, bukan lebih berat atau lambat saat digunakan untuk prediksi. Biaya yang dibayar adalah waktu pelatihan yang lebih lama, bukan efisiensi saat implementasi.

2. Bukti Konseptual: Mengapa Pelatihan Lebih Lama Berhasil?

Hasil kuantitatif di atas dapat dijelaskan oleh prinsip dasar deep learning:

- Konvergensi yang lebih matang: grafik pelatihan Gambar 4.8 menunjukkan bahwa kurva loss masih terus menurun dan kurva mAP masih terus menanjak, meskipun sudah mulai melandai. Ini adalah indikasi jelas bahwa pada 80 epoch, model masih dalam proses belajar dan belum mencapai titik konvergensi optimalnya. Pelatihan hingga 500 epoch memberikan kesempatan bagi optimizer untuk terus melakukan penyesuaian kecil pada jutaan bobot di dalam jaringan, membawanya lebih dekat ke titik minima global (atau lokal yang sangat baik) pada loss landscape.
- Pembelajaran Fitur yang Lebih Halus (Fine-Grained Features):
 - Epoch Awal (misal, 1-80): Model belajar fitur-fitur yang umum dan kasar, seperti bentuk umum pesawat, sayap, dan ekor. Inilah sebabnya mAP_{0.5} sudah sangat tinggi di 80 epoch.

- Epoch Lanjutan (misal, 81-500): Model mulai belajar fitur-fitur yang jauh lebih subtil dan diskriminatif. Ini adalah tahap di mana model belajar membedakan perbedaan kecil antara SU-34 dan SU-35, atau antara varian F-15 yang berbeda. Pembelajaran fitur fine-grained inilah yang secara langsung berkontribusi pada lonjakan signifikan pada metrik mAP 0.5-95.
- Generalisasi yang lebih baik (Tanpa Overfitting): Salah satu risiko pelatihan yang panjang adalah overfitting, di mana model menghafal data latih dan berkinerja buruk pada data baru. Namun, hasil mAP yang superior pada data pengujian (test set) membuktikan bahwa model 500 epoch tidak mengalami overfitting. Sebaliknya, ia berhasil mempelajari representasi fitur yang lebih tangguh dan dapat digeneralisasi dengan lebih baik.

Pelatihan selama 500 epoch terbukti lebih baik karena data eksperimen secara kuantitatif menunjukkan peningkatan signifikan di semua metrik akurasi (terutama pada presisi lokalisasi (mAP 0.5-95)) tanpa mengorbankan kecepatan inferensi dan tanpa overfitting. Secara konseptual, durasi pelatihan yang lebih panjang ini memungkinkan model untuk mencapai konvergensi yang lebih matang. Model mempelajari fitur-fitur fine-grained yang krusial untuk membedakan antar kelas yang mirip, yang pada akhirnya menghasilkan sebuah model dengan kemampuan deteksi dan identifikasi yang jauh lebih superior dan dapat digeneralisasi.

BAB 5

PENUTUP

5.1 Kesimpulan

Berdasarkan analisis dan evaluasi yang telah dilakukan, maka dapat ditarik kesimpulan yang menjawab rumusan masalah penelitian sebagai berikut:

1. Penelitian ini berhasil mengembangkan algoritma deteksi berbasis YOLO yang mampu meningkatkan akurasi dan efisiensi secara signifikan melalui inovasi arsitektur YOLO11 dengan integrasi *Global Attention Module* (GAM). Modul GAM terbukti efektif dalam mengatasi tantangan deteksi di citra penginderaan jarak jauh dengan memaksa model untuk lebih fokus pada fitur-fitur diskriminatif target pesawat dan menekan interferensi dari latar belakang yang kompleks. Pendekatan ini, dikombinasikan dengan *tuning hyperparameter* yang cermat, khususnya penggunaan *optimizer* NAdam dengan *learning rate* 0.00147 berhasil menghasilkan model dengan performa puncak, mAP50-95 sebesar 80,6% dan mAP50 sebesar 98,6%. Dari segi efisiensi, model usulan ini mencatatkan kecepatan inferensi superior sebesar 159 FPS, yang secara signifikan melampaui metode pembandingan lainnya, membuktikan bahwa peningkatan akurasi dapat dicapai bersamaan dengan efisiensi komputasi yang tinggi.
2. Model YOLO11 yang telah dimodifikasi dengan modul GAM menunjukkan peningkatan kinerja yang signifikan dan terukur dibandingkan dengan versi YOLO11 asli. Peningkatan paling krusial terlihat pada metrik presisi

lokalisasi (mAP50-95), di mana model modifikasi terbaik, meningkat menjadi 80,6% dibandingkan sebelumnya 79%, naik 1,6% dari model asli. Peningkatan ini mengindikasikan bahwa modul atensi secara substansial menyempurnakan kemampuan model dalam menempatkan *bounding box* secara akurat, yang merupakan aspek vital untuk aplikasi militer. Meskipun demikian, peningkatan akurasi ini diiringi oleh *trade-off* pada kecepatan inferensi, di mana kecepatan menurun dari sekitar 250 FPS pada model asli menjadi sekitar 159 FPS pada model modifikasi. Hal ini menegaskan bahwa modifikasi GAM memberikan keunggulan presisi yang jelas dengan mengorbankan sebagian efisiensi komputasi, sebuah pertimbangan penting tergantung pada prioritas aplikasi, apakah akurasi maksimal atau latensi minimal.

3. Kinerja mean Average Precision (mAP) bukanlah hasil dari satu faktor tunggal, melainkan sebuah metrik holistik yang merefleksikan interaksi kompleks dari berbagai komponen penting.
 - a. Pada level arsitektur, inovasi seperti integrasi modul atensi, sebagaimana terbukti dalam penelitian ini, secara langsung meningkatkan mAP dengan memaksa model untuk fokus pada fitur relevan dan menyempurnakan presisi lokalisasi, yang sangat berpengaruh pada metrik mAP50-95.
 - b. Proses pelatihan kemudian menjadi krusial, di mana pemilihan optimizer adaptif seperti NAdam terbukti mampu

mengonvergensi model secara lebih efektif dibandingkan metode konvensional.

- c. Di samping itu, learning rate bertindak sebagai parameter paling sensitif; nilai yang terlalu tinggi dapat menyebabkan instabilitas, sementara nilai yang terlalu rendah menghambat konvergensi, sehingga penemuan laju pembelajaran yang optimal menjadi kunci untuk mencapai mAP puncak.
- d. Fondasi dari semua ini adalah kualitas dan kuantitas dataset, di mana strategi augmentasi data yang efektif berperan vital dalam meningkatkan robustitas dan kemampuan generalisasi model, yang secara langsung diterjemahkan menjadi mAP yang lebih tinggi pada data uji.

Dengan demikian, pencapaian mAP yang superior bukanlah hasil dari optimasi satu aspek, melainkan hasil dari sinergi yang harmonis antara desain arsitektur yang canggih, strategi pelatihan yang terkalibrasi dengan baik, dan dataset yang berkualitas tinggi.

5.2 Saran

Berdasarkan temuan yang diperoleh dan keterbatasan yang ada dalam penelitian ini, beberapa saran untuk pengembangan di masa depan dapat diajukan sebagai berikut:

1. Eksplorasi Mekanisme Atensi Ringan

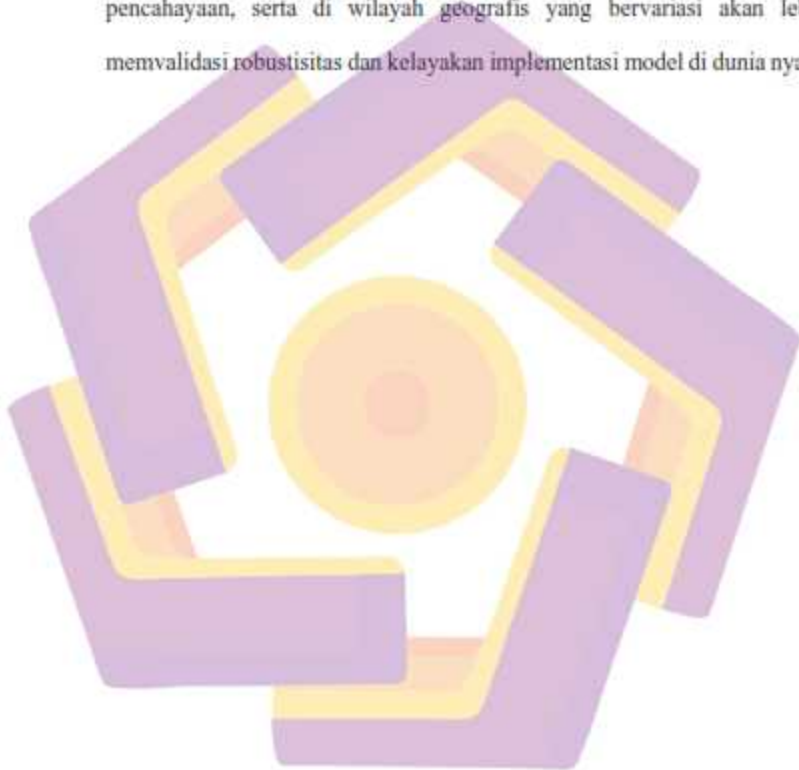
Mengingat adanya *trade-off* antara akurasi dan kecepatan akibat penambahan modul GAM, penelitian selanjutnya dapat berfokus pada eksplorasi atau perancangan mekanisme atensi yang lebih ringan (*lightweight*, seperti Squeeze-and-Excitation (SE)[67] atau Convolutional Block Attention Module (CBAM)[68]) untuk meminimalkan beban komputasi. Menggabungkan keunggulan atensi dengan teknik optimasi model seperti kuantisasi (*quantization*) atau *pruning* juga dapat menjadi arah yang menjanjikan untuk mendapatkan model yang sangat akurat sekaligus sangat cepat.

2. Penanganan Klasifikasi *Fine-Grained*

Hasil *confusion matrix* menunjukkan adanya kebingungan antara sub-tipe pesawat yang visualnya sangat mirip (misalnya, SU-34 dan SU-35). Penelitian di masa depan dapat secara khusus menargetkan masalah klasifikasi *fine-grained* [69] dengan mengimplementasikan modul tambahan yang dirancang untuk mengekstraksi fitur-fitur subtil atau dengan menggunakan strategi *data augmentation* yang secara spesifik menonjolkan perbedaan kecil antar kelas yang serupa.

3. Pengujian Generalisasi pada Dataset yang Lebih Beragam

Meskipun model menunjukkan performa yang sangat baik pada dataset yang digunakan, pengujian lebih lanjut pada dataset yang lebih luas dan beragam sangat direkomendasikan. Menguji generalisasi model pada citra dari sensor satelit yang berbeda, dalam berbagai kondisi cuaca dan pencahayaan, serta di wilayah geografis yang bervariasi akan lebih memvalidasi robustitas dan kelayakan implementasi model di dunia nyata.



DAFTAR PUSTAKA

- [1] Y. Liu, T. Sun, and C. Cui, "Research on Aircraft Target Detection Method Based on YOLO: IEEE ITAIC(ISSN:2693-2865)," in *2022 IEEE 10th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, 2022, pp. 49–52. doi: 10.1109/ITAIC54216.2022.9836475.
- [2] X. Wang, W. Hong, and Y. Liu, "Detection and recognition of aircraft target in SAR image," in *Proc.SPIE*, Aug. 2023, p. 127971V. doi: 10.1117/12.3007349.
- [3] J. Li, W. Zhu, and Y. Yang, "Research on aircraft target recognition in SAR images," in *Proc.SPIE*, Dec. 2021, p. 1208725. doi: 10.1117/12.2624735.
- [4] W. Tang, B. Jun, L. Tang, Y. Pan, D. Jin, and Z. Quan, "Aircraft detection in Remote Sensing image for Space-borne platform," in *2019 IEEE International Conference on Signal, Information and Data Processing (ICSIDP)*, 2019, pp. 1–4. doi: 10.1109/ICSIDP47821.2019.9173487.
- [5] L. Yu, H. Hu, Z. Zhong, H. Wu, and Q. Deng, "GLF-Net: A Target Detection Method Based on Global and Local Multiscale Feature Fusion of Remote Sensing Aircraft Images," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022. doi: 10.1109/LGRS.2022.3144284.
- [6] B. Zeng *et al.*, "Top-Down aircraft detection in large-scale scenes based on multi-source data and FEF-R-CNN," *Int. J. Remote Sens.*, vol. 43, no. 3, pp. 1108–1130, Feb. 2022. doi: 10.1080/01431161.2022.2034194.
- [7] L. Li, N. Peng, B. Li, and H. Liu, "Real-time airplane detection using multi-dimensional attention and feature fusion," *PeerJ Comput. Sci.*, vol. 9, p. e1331, 2023. doi: 10.7717/peerj-cs.1331.
- [8] Z. Liu, Y. Gao, and Q. Du, "YOLO-Class: Detection and Classification of Aircraft Targets in Satellite Remote Sensing Images Based on YOLO-Extract," *IEEE Access*, vol. 11, pp. 109179–109188, 2023. doi: 10.1109/ACCESS.2023.3321828.
- [9] J. Wu, F. Zhao, and Z. Jin, *LEN-YOLO: A Lightweight Remote Sensing Small Aircraft Object Detection Model for Satellite On-Orbit Detection*. 2024. doi: 10.21203/rs.3.rs-4836110/v1.
- [10] Z. Liu, Y. Gao, Q. Du, M. Chen, and W. Lv, "YOLO-Extract: Improved YOLOv5 for Aircraft Object Detection in Remote Sensing Images," *IEEE Access*, vol. 11, pp. 1742–1751, 2023. doi: 10.1109/ACCESS.2023.3233964.
- [11] M. A. R. Alif, "YOLOv11 for Vehicle Detection: Advancements, Performance, and Applications in Intelligent Transportation Systems," *ArXiv*, vol. abs/2410.2, 2024. doi: 10.48550/arXiv.2410.22898.
- [12] S. Luo, J. Yu, Y. Xi, and X. Liao, "Aircraft Target Detection in Remote Sensing Images Based on Improved YOLOv5," *IEEE Access*, vol. 10, pp. 5184–5192, 2022. doi: 10.1109/ACCESS.2022.3140876.

- [13] M. Li, J. Rui, L. Ren, Y. Songkun, Y. Lin, and Y. Wang, "Surface Feature Detection Method of Optical Remote Sensing Images Based on YOLOX," in *2022 2nd International Conference on Big Data, Artificial Intelligence and Risk Management (ICBAR)*, 2022, pp. 79–85. doi: 10.1109/ICBAR58199.2022.00023.
- [14] L. Shen, B. Lang, and Z. Song, "CA-YOLO: Model Optimization for Remote Sensing Image Object Detection," *IEEE Access*, vol. 11, pp. 64769–64781, 2023, doi: 10.1109/ACCESS.2023.3290480.
- [15] D. Zhang, Z. Zhao, and S. Huang, "Investigation of Aircraft Target Detection of Remote Sensing Images Based on the Improved YOLOv5," in *2023 International Conference on Computer Applications Technology (CCAT)*, 2023, pp. 266–270. doi: 10.1109/CCAT59108.2023.00056.
- [16] M. Huang, W. Yan, W. Dai, and J. Wang, "EST-YOLOv5s: SAR Image Aircraft Target Detection Model Based on Improved YOLOv5s," *IEEE Access*, vol. 11, pp. 113027–113041, 2023, doi: 10.1109/ACCESS.2023.3323575.
- [17] S. L. Xu, Z. Chen, H. Zhang, and H. Su, "Improved Aircraft Target Detection Algorithm for Remote Sensing Images with YOLOv8," in *2023 3rd International Conference on Electronic Information Engineering and Computer Science (EIECS)*, 2023, pp. 317–321. doi: 10.1109/EIECS59936.2023.10435536.
- [18] F. Wang, H. Wang, Z. Qin, and J. Tang, "UAV target detection algorithm based on improved YOLOv8," *IEEE Access*, vol. PP, p. 1, Jan. 2023, doi: 10.1109/ACCESS.2023.3325677.
- [19] X. Chen *et al.*, "DET-YOLO: An Innovative High-Performance Model for Detecting Military Aircraft in Remote Sensing Images," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 17, pp. 17753–17771, 2024, doi: 10.1109/JSTARS.2024.3462745.
- [20] R. Khanam and M. Hussain, "YOLOv11: An Overview of the Key Architectural Enhancements," *ArXiv*, vol. abs/2410.1, 2024, [Online]. Available: <http://arxiv.org/abs/2410.17725>
- [21] N. B. A. Karna, M. A. P. Putra, S. M. Rachmawati, M. Abisado, and G. A. Sampedro, "Toward Accurate Fused Deposition Modeling 3D Printer Fault Detection Using Improved YOLOv8 With Hyperparameter Optimization," *IEEE Access*, vol. 11, pp. 74251–74262, 2023, doi: 10.1109/ACCESS.2023.3293056.
- [22] O. Ajayi, P. Ibrahim, and O. Adegboyega, "Effect of Hyperparameter Tuning on the Performance of YOLOv8 for Multi Crop Classification on UAV Images," *Appl. Sci.*, vol. 14, p. 5708, Jun. 2024, doi: 10.3390/app14135708.
- [23] D. Titisari, E. Yuniarno, I. Purnama, and M. Hery Purnomo, *Enhancing Breast Cancer Detection: Optimizing YOLOv8's Performance Through Hyperparameter Tuning*, 2023. doi: 10.1109/ICITDA60835.2023.10427255.

- [24] B. Yilmaz and U. Kutbay, "YOLOv8-Based Drone Detection: Performance Analysis and Optimization," 2024. doi: 10.3390/computers13090234.
- [25] O. Khare, S. Gandhi, A. Rahalkar, and S. Mane, *YOLOv8-Based Visual Detection of Road Hazards: Potholes, Sewer Covers, and Manholes*. 2023. doi: 10.1109/PuneCon58714.2023.10449999.
- [26] D. Das, B. Keshri, K. Rajbongshi, and K. Mukherjee, "Underwater Pipeline Detection using Deep Learning Techniques," in *2024 6th International Conference on Energy, Power and Environment (ICEPE)*, 2024, pp. 1–6. doi: 10.1109/ICEPE63236.2024.10668883.
- [27] D. Arivalagan, D. Rubankumar, K. Thushiyanth, S. Sundaram, V. Priyadharsini, and V. O., *Weapon Detection System For Security And Surveillance Using YOLOv8*, 2024. doi: 10.1109/InCACCT61598.2024.10551046.
- [28] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, Dec. 2015, [Online]. Available: <https://arxiv.org/pdf/1412.6980>
- [29] A. C. Wilson, R. Roelofs, M. Stern, N. Srebro, and B. Recht, "The Marginal Value of Adaptive Gradient Methods in Machine Learning," in *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/81b3833e2504647f9d794f7d7b9bf341-Paper.pdf
- [30] T. Dozat, "Incorporating Nesterov Momentum into," 2015. [Online]. Available: <https://api.semanticscholar.org/CorpusID:620137>
- [31] L. Liu *et al.*, *On the Variance of the Adaptive Learning Rate and Beyond*. 2019. doi: 10.48550/arXiv.1908.03265.
- [32] Supatman, E. M. Yuniarno, and M. H. Purnomo, "A Comparison of Optimizer Algorithms in YOLOv8 for Automatic Detection of Knee Landmarks," in *2023 8th International Conference on Information Technology and Digital Applications (ICITDA)*, 2023, pp. 1–6. doi: 10.1109/ICITDA60835.2023.10427072.
- [33] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," in *International Conference on Learning Representations*, 2017. [Online]. Available: <https://api.semanticscholar.org/CorpusID:53592270>
- [34] E. Sudewo, M. Biddinika, R. Umar, and A. Fadli, "Evaluating the Impact of Optimizer Hyperparameters on ResNet in Hanacaraka Character Recognition," *Preserv. Digit. Technol. Cult.*, Feb. 2025, doi: 10.1515/pdte-2024-0061.
- [35] L. Bottou, "Large-Scale Machine Learning with Stochastic Gradient Descent," *Proc. of COMPSTAT*, Sep. 2010, doi: 10.1007/978-3-7908-2604-3_16.
- [36] N. Gkouti, P. Malakasiotis, S. Toumpis, and I. Androutsopoulos, "Should I

- try multiple optimizers when fine-tuning pre-trained Transformers for NLP tasks? Should I tune their hyperparameters?," *arXiv Prepr. arXiv2402.06948*, 2024.
- [37] M. J. Uddin, Y. Li, M. A. Sattar, Z. M. Nasrin, and C. Lu, "Effects of learning rates and optimization algorithms on forecasting accuracy of hourly typhoon rainfall: Experiments with convolutional neural network," *Earth Sp. Sci.*, vol. 9, no. 3, p. e2021EA002168, 2022.
- [38] D. Lopez-Betancur *et al.*, "Evaluation of optimization algorithms for measurement of suspended solids," *Water*, vol. 16, no. 13, p. 1761, 2024.
- [39] Y. Liu, Z. Shao, and N. Hoffmann, "Global Attention Mechanism: Retain Information to Enhance Channel-Spatial Interactions," 2021. [Online]. Available: <http://arxiv.org/abs/2112.05561>
- [40] W. Yu *et al.*, "MAR20 : A benchmark for military aircraft recognition in remote sensing images," *Natl. Remote Sens. Bull.*, 2023, doi: 10.11834/jrs.20222139.
- [41] A. Rácz, D. Bajusz, and K. Héberger, "Effect of Dataset Size and Train/Test Split Ratios in QSAR/QSPR Multiclass Classification," *Molecules*, vol. 26, 2021, doi: 10.3390/molecules26041111.
- [42] K. Dobbin and R. M. Simon, "Optimally splitting cases for training and testing high dimensional classifiers," 2011. doi: 10.1186/1755-8794-4-31.
- [43] D. Xu and Y. Wu, "Improved YOLO-V3 with DenseNet for Multi-Scale Remote Sensing Target Detection," *Sensors (Basel)*, vol. 20, 2020, doi: 10.3390/s20154276.
- [44] L. Kong, J. Wang, and P. Zhao, "YOLO-G: A Lightweight Network Model for Improving the Performance of Military Targets Detection," *IEEE Access*, vol. 10, pp. 55546–55564, 2022, doi: 10.1109/ACCESS.2022.3177628.
- [45] M. Kim, J. Jeong, and S. Kim, "ECAP-YOLO: Efficient Channel Attention Pyramid YOLO for Small Object Detection in Aerial Image," *Remote. Sens.*, vol. 13, p. 4851, 2021, doi: 10.3390/rs13234851.
- [46] G. Zhang, Y. Peng, and J. Li, "YOLO-MARS: An Enhanced YOLOv8n for Small Object Detection in UAV Aerial Imagery," 2025. doi: 10.3390/s25082534.
- [47] P. J. and S. A., "Synergistic fusion: An integrated pipeline of CLAHE, YOLO models, and advanced super-resolution for enhanced thermal eye detection," *PLoS One*, vol. 20, no. 7, p. e0328227, Jul. 2025, [Online]. Available: <https://doi.org/10.1371/journal.pone.0328227>
- [48] H. Qiang *et al.*, "SCM-YOLO for Lightweight Small Object Detection in Remote Sensing Images," 2025. doi: 10.3390/rs17020249.
- [49] S. Modak and A. Stein, "Generative AI-based pipeline architecture for increasing training efficiency in intelligent weed control systems," *J. Syst. Archit.*, vol. 167, p. 103464, Jun. 2025, doi: 10.1016/j.sysarc.2025.103464.
- [50] Y. Song, Z. Xie, X. Wang, and Y. Zou, "MS-YOLO: Object Detection Based

- on YOLOv5 Optimized Fusion Millimeter-Wave Radar and Machine Vision," *IEEE Sens. J.*, vol. 22, pp. 15435–15447, 2022, doi: 10.1109/jsen.2022.3167251.
- [51] Q. Wang *et al.*, "YOLO-FeatSup: Enhancing Small-Object Detection in Remote Sensing Images With GLSA and ScalSeq Modules Based on YOLOv11," *IEEE Geosci. Remote Sens. Lett.*, vol. 22, pp. 1–5, 2025, doi: 10.1109/LGRS.2025.3615660.
- [52] J. Suto, "Using Data Augmentation to Improve the Generalization Capability of an Object Detector on Remote-Sensed Insect Trap Images," 2024. doi: 10.3390/s24144502.
- [53] B. Zhao, Y. Wu, X. Guan, L. Gao, and B. Zhang, "An Improved Aggregated-Mosaic Method for the Sparse Object Detection of Remote Sensing Imagery," 2021. doi: 10.3390/rs13132602.
- [54] T.-Y. Lin *et al.*, *Microsoft COCO: Common Objects in Context*, vol. 8693, 2014. doi: 10.1007/978-3-319-10602-1_48.
- [55] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks.," *Neural networks Off. J. Int. Neural Netw. Soc.*, vol. 106, pp. 249–259, Oct. 2018, doi: 10.1016/j.neunet.2018.07.011.
- [56] A. Subedi, *Improving Generalization Performance of YOLOv8 for Camera Trap Object Detection*. 2024. doi: 10.48550/arXiv.2412.14211.
- [57] K. Kim, K. Kim, and S. Jeong, "Application of YOLO v5 and v8 for Recognition of Safety Risk Factors at Construction Sites," 2023. doi: 10.3390/su152015179.
- [58] M. H. Tutar *et al.*, "AI-Powered Image-Based Assessment of Pressure Injuries Using You Only Look Once Version 8 (YOLOv8) Models," *Adv. Wound Care*, Mar. 2025, doi: 10.1089/wound.2024.0245.
- [59] Y. Shen, Z. Yang, Z. Khan, H. Liu, W. Chen, and S. Duan, "Optimization of Improved YOLOv8 for Precision Tomato Leaf Disease Detection in Sustainable Agriculture," 2025. doi: 10.3390/s25051398.
- [60] J. Ding, S. Niu, Z. Nie, and W. Zhu, "Research on Human Posture Estimation Algorithm Based on YOLO-Pose," 2024. doi: 10.3390/s24103036.
- [61] L. He, Y. Zhou, L. Liu, W. Cao, and J. Ma, "Research on object detection and recognition in remote sensing images based on YOLOv11," *Sci. Rep.*, vol. 15, no. 1, p. 14032, 2025, doi: 10.1038/s41598-025-96314-x.
- [62] C. Zhou *et al.*, "Metal surface defect detection based on improved YOLOv5," *Sci. Rep.*, vol. 13, no. 1, p. 20803, 2023, doi: 10.1038/s41598-023-47716-2.
- [63] X. Qin, C. Yu, B. Liu, and Z. Zhang, "YOLO8-FASG: A High-Accuracy Fish Identification Method for Underwater Robotic System," *IEEE Access*, vol. 12, pp. 73354–73362, 2024, doi: 10.1109/ACCESS.2024.3404867.
- [64] J. Yang, R. Tian, Z. Zhou, X. Tan, and P. He, "Flexi-YOLO: A lightweight

- method for road crack detection in complex environments,” *PLoS One*, vol. 20, no. 6, p. e0325993, Jun. 2025, [Online]. Available: <https://doi.org/10.1371/journal.pone.0325993>
- [65] X. Li, H. Zhang, Y. Hang, and H. Chen, “Small target detection algorithm based on the fusion attention mechanism and multi-layer convolution,” *PLoS One*, vol. 20, no. 7, p. e0328003, Jul. 2025, [Online]. Available: <https://doi.org/10.1371/journal.pone.0328003>
- [66] J. Wang, Z. Bai, X. Zhang, and Y. Qiu, “A Lightweight Remote Sensing Aircraft Object Detection Network Based on Improved YOLOv5n,” 2024, doi: 10.3390/rs16050857.
- [67] J. Hu, L. Shen, and G. Sun, “Squeeze-and-Excitation Networks,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132–7141. doi: 10.1109/CVPR.2018.00745.
- [68] S. Woo, J. Park, J.-Y. Lee, and I. Kweon, “CBAM: Convolutional Block Attention Module: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part VII,” 2018, pp. 3–19. doi: 10.1007/978-3-030-01234-2_1.
- [69] J. Wu, F. Zhao, G. Yao, and Z. Jin, “FGA-YOLO: A one-stage and high-precision detector designed for fine-grained aircraft recognition,” *Neurocomputing*, vol. 618, p. 129067, 2025, doi: <https://doi.org/10.1016/j.neucom.2024.129067>.