

BAB I

PENDAHULUAN

1.1 Latar Belakang

Deteksi objek secara *real-time* merupakan bidang penelitian yang terus berkembang dan semakin banyak penggunaannya dalam berbagai bidang mulai dari kendaraan otonom, *monitoring* industri, hingga pengawasan lingkungan. Salah satu algoritma yang paling umum digunakan untuk deteksi objek adalah model You Only Look Once (YOLO). Model YOLO pertama kali diperkenalkan oleh Joseph Redmon pada tahun 2016 (Redmon dkk., 2016). YOLO terkenal karena kemampuannya untuk melakukan deteksi dengan nilai mean Average Precision (mAP) dan Frame Per-Second (FPS) yang lebih baik dibandingkan algoritma deteksi objek tradisional seperti R-CNN atau Fast R-CNN (Liunanda dkk., 2020). Model YOLO terus berkembang hingga versi terbaru yaitu YOLOv11. YOLOv11 mampu meningkatkan akurasi dan kecepatan yang lebih tinggi dibandingkan model YOLO sebelumnya dengan penggunaan daya komputasi yang lebih rendah (Derrenger dkk., 2024). Namun, penerapan YOLO pada perangkat dengan daya komputasi terbatas seperti ponsel atau perangkat IoT masih menghadapi tantangan yang signifikan.

Kebutuhan untuk mengurangi beban komputasi model YOLO pada perangkat *low-end* mendorong berbagai penelitian yang mengeksplorasi metode optimasi YOLO. Penggunaan model YOLO untuk deteksi objek *real-time* seperti dalam aplikasi pengawasan (*home surveillance system*) atau *object counter* pada produk

di atas *conveyor* membutuhkan waktu respon deteksi atau *inference time* yang tinggi (Liberatori dkk., 2022). Waktu inferensi yang lambat dapat menyebabkan penurunan kinerja yang mengakibatkan keterbatasan dalam aplikasi dunia nyata, khususnya dalam mendeteksi objek yang bergerak atau dalam kondisi lingkungan yang kompleks (Dalal dkk., 2024; Javed dkk., 2021; Wai dkk., 2019). Sistem menjadi tidak reliabel dan juga tidak mampu melakukan tugasnya dengan baik karena waktu yang dibutuhkan oleh model untuk mendeteksi objek dan mengklasifikasikannya terlalu lama (Imanullah & Reswan, 2022; Liberatori dkk., 2022). Dengan demikian, terdapat kebutuhan yang mendesak untuk mengoptimalkan model YOLO agar mampu mencapai kinerja yang lebih optimal ketika model digunakan pada perangkat dengan daya komputasi rendah (Dalal dkk., 2024; Javed dkk., 2021).

Penelitian sebelumnya telah menggunakan berbagai pendekatan untuk meningkatkan efisiensi YOLO pada perangkat berdaya komputasi rendah, termasuk *hyperparameter tuning* dan *quantization* (Dettmers dkk., 2021; Liao dkk., 2022). *Hyperparameter tuning* adalah proses penyesuaian parameter pelatihan model seperti *learning rate*, ukuran *batch*, dan jenis *optimizer* untuk mencapai keseimbangan antara akurasi deteksi dan kecepatan inferensi (Feurer & Hutter, 2019; Isa dkk., 2022; R. Anita Jasmine dkk., 2022). Penelitian yang dilakukan oleh (Ramos dkk., 2024a, 2024b) berhasil meningkatkan performa YOLOv8 melalui penggabungan metode *One Factor at A Time* (OFAT) dan *Random Search* (RS) dalam proses optimasi *hyperparameter*. Pada tahap awal, metode OFAT digunakan untuk mengevaluasi pengaruh masing-masing *hyperparameter* secara terpisah

terhadap performa sistem deteksi objek. Evaluasi ini bertujuan untuk mengidentifikasi nilai-nilai *hyperparameter* yang paling optimal pada *dataset* yang digunakan. Nilai-nilai optimal hasil OFAT kemudian di gunakan untuk mempersempit ruang pencarian, sehingga proses *Random Search* dapat difokuskan pada area lokal yang lebih relevan. Meskipun demikian, *Random Search* memiliki keterbatasan dalam hal eksplorasi ruang pencarian yang lebih luas dan komprehensif, sehingga belum mampu secara optimal menemukan kombinasi *hyperparameter* terbaik secara global. Sebagai alternatif yang lebih menjanjikan, penggunaan Algoritma Genetika (GA) telah terbukti efektif untuk mengeksplorasi ruang pencarian yang besar dan kompleks, serta menemukan konfigurasi *hyperparameter* optimal, seperti yang ditunjukkan dalam penelitian (Naik & Rudra, 2024; Soltanikazemi dkk., 2023; Suhail & Brindha, 2024) pada berbagai varian bobot model YOLOv5. Hanya saja, Algoritma Genetika memerlukan sumber daya komputasi yang sangat tinggi apabila search space yang diberikan terlalu luas dan tidak di fokuskan pada nilai optimal yang ingin dicari (Safarik dkk., 2018; Santoso dkk., 2023).

Integer quantization adalah teknik untuk mengonversi parameter model ke dalam format integer 8-bit untuk mengurangi ukuran model dan meningkatkan waktu inferensi sehingga model menjadi lebih sederhana dan dapat dijalankan pada perangkat *low-end* (Kluska & Zięba, 2020; Krishnamoorthi, 2018; Liberatori dkk., 2022; Zmora dkk., 2021). Hanya saja, model yang sudah di kuantisasi umumnya akan mengalami penurunan akurasi deteksi dan klasifikasi objek karena penurunan

nilai presisi model dari rentang 32-bit menjadi rentang bilangan bulat 8-bit (Kluska & Zięba, 2020; Liberatori dkk., 2022; Zmora dkk., 2021).

Namun, penelitian mengenai optimasi YOLO dengan kombinasi *hyperparameter tuning* dan *integer quantization* secara mendalam masih terbatas. Tantangan utama yang belum banyak dibahas adalah bagaimana menyeimbangkan performa model dan waktu inferensi model setelah proses *post-training quantization* agar didapat performa yang optimal pada perangkat berdaya komputasi rendah (Dalal et al., 2024; Javed et al., 2021). Mengingat terbatasnya penelitian yang mendalami kombinasi dua teknik optimasi ini terlebih pada versi YOLO yang lebih baru, penelitian ini bertujuan untuk mengisi kesenjangan pada penelitian sebelumnya dengan mengimplementasikan *hyperparameter tuning* pada YOLOv11 dengan kombinasi metode OFAT untuk mengevaluasi setiap *hyperparameter* secara terpisah dan menentukan nilai *hyperparameter* yang memberikan performa optimal sebagai *hyperparameter parent* metode GA dan mempersempit ruang pencarian metode GA. *Hyperparameter tuning* berperan untuk mengompensasi penurunan akurasi yang disebabkan oleh proses *integer quantization* sehingga dapat menghasilkan model YOLOv11 yang ringan tetapi memiliki nilai akurasi yang sama dengan model YOLOv11 dasar atau bahkan bisa lebih baik. Evaluasi terhadap model akhir yang telah di-*deploy* akan menggunakan nilai *fitness* sebagai metrik gabungan untuk menilai performa sistem secara menyeluruh. Nilai *fitness* ini tidak hanya merepresentasikan tingkat keberhasilan deteksi dan klasifikasi objek melalui mAP50 dan mAP50-95, tetapi juga menyeimbangkannya dengan nilai F1-Score yang merepresentasikan

keseimbangan antara *precision* dan *recall*. Dengan demikian, evaluasi tidak hanya berfokus pada akurasi deteksi secara spasial, tetapi juga pada konsistensi dan keandalan deteksi dalam kondisi nyata.

Penelitian ini membuka peluang untuk mengembangkan model deteksi objek yang lebih efisien dan ringan. Model baru yang optimal juga dapat dijadikan landasan dalam pengembangan model lebih lanjut khususnya untuk *deployment* model pada perangkat dengan daya komputasi terbatas. Dengan optimasi model YOLOv11 melalui kombinasi *hyperparameter tuning* dan *integer quantization*, diharapkan model ini dapat diterapkan dalam berbagai aplikasi deteksi *real-time* seperti sistem pengawasan tempat, kendaraan otonom, aplikasi *mobile* penghitung/deteksi/klasifikasi objek dan perangkat *wearable* yang memerlukan deteksi cepat, akurat serta ringan. Pengembangan model ini diharapkan dapat meningkatkan kualitas sistem deteksi objek sehingga performanya lebih baik dibandingkan model YOLOv11 dasar atau bahkan model deteksi objek lain dan mampu memenuhi kebutuhan aplikasi dunia nyata yang semakin kompleks.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah dijabarkan sebelumnya maka dalam penelitian ini permasalahan yang akan dirumuskan adalah sebagai berikut:

1. Bagaimana pengaruh metode OFAT dalam proses *hyperparameter tuning* terhadap nilai *fitness* model YOLOv11?
2. Apakah kombinasi metode OFAT dan GA dalam proses *hyperparameter tuning* dapat meningkatkan nilai *fitness* model dibandingkan dengan hanya menggunakan metode OFAT saja?

3. Bagaimana pengaruh *integer quantization* terhadap waktu inferensi dan FPS deteksi model YOLOv11?
4. Apakah kombinasi *hyperparameter tuning* dan *integer quantization* dapat meningkatkan waktu inferensi dan akurasi model YOLOv11 untuk deteksi objek secara *real-time* pada perangkat *low-end*?

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Menganalisis pengaruh *hyperparameter tuning* metode OFAT terhadap nilai *fitness* model YOLOv11.
2. Menganalisis efektivitas kombinasi metode OFAT dan GA dalam meningkatkan nilai *fitness* model YOLOv11.
3. Menganalisis pengaruh *integer quantization* terhadap waktu inferensi dan FPS deteksi model YOLOv11.
4. Menganalisis efektivitas kombinasi teknik *hyperparameter tuning* dan *integer quantization* dalam mengoptimalkan model YOLOv11 untuk deteksi objek *real-time* pada perangkat *low-end*.

1.4 Manfaat Penelitian

Manfaat dari penelitian ini adalah:

1. Menganalisis tingkat kenaikan performa dari penerapan *Hyperparameter Tuning* dan *Integer Quantization* pada model YOLOv11 dalam aplikasi android berbasis Flutter pada deteksi objek secara *real-time*.

2. Mengetahui tingkat keakuratan deteksi objek secara *real-time* dari model YOLOv11 yang sudah di optimalisasi dengan *Hyperparameter Tuning* dan *Integer Quantization* serta diterapkan ke dalam aplikasi android.
3. Memberikan solusi praktis dalam mengimplementasikan model YOLOv11 yang efisien pada perangkat dengan daya komputasi terbatas, sehingga model dapat digunakan pada berbagai aplikasi komersial seperti sistem *monitoring*, otomatisasi industri, dan keamanan.
4. Memperluas penggunaan teknologi deteksi objek *real-time* untuk meningkatkan pengalaman pengguna dalam bidang seperti navigasi kendaraan otonom, atau aplikasi yang memerlukan deteksi objek secara cepat dan akurat dengan sumber daya komputasi yang terbatas.

1.5 Batasan Masalah

Batasan masalah dalam penelitian ini adalah:

1. Objek yang dipilih dalam penelitian ini adalah kendaraan dengan anggota kelas Bus, Mobil, Motor dan Truk.
2. Metode yang digunakan dalam penelitian ini adalah metode *Hyperparameter Tuning* dan *Integer Quantization*.
3. Data yang digunakan sebagai data *training*, *validation* dan *test* berupa video lalu lintas kendaraan dalam format .mp4 yang kemudian di unggah ke dalam *website* Roboflow. Video tersebut kemudian di konversi menjadi *dataset* Gambar dengan rentang pengambilan 5 Gambar per detik.
4. Pembagian data berupa 70% data latih, 20% data validasi dan 10% data uji.
5. Model dasar YOLOv11 yang digunakan adalah YOLOv11n (Nano)

6. *Software* yang dipakai untuk pelatihan model YOLOv11 adalah Google Colab dan Kaggle dengan bahasa pemrograman Python.
7. *Graphics Processing Unit* (GPU) yang dipakai untuk pelatihan model YOLOv11 adalah GPU yang tersedia gratis di platform Google Colab dan Kaggle yaitu NVIDIA P100 VRAM 16GB satu unit.
8. *Software* yang dipakai untuk membuat aplikasi android berbasis Flutter menggunakan Android Studio dengan bahasa pemrograman Dart.
9. Pengujian dan evaluasi performa deteksi model dilakukan pada platform Android berbasis Flutter.
10. Perangkat keras yang digunakan adalah gawai Android OPPO Reno 12f dengan spesifikasi Android 14, kamera 50MP, RAM 8 GB, Octacore CPU 2 GHz, dan Mali-G57 MC2 GPU.

1.6 Sistematika Penulisan

Sistematika penulisan yang digunakan dalam laporan ilmiah tugas akhir ini dapat diuraikan sebagai berikut:

BAB I PENDAHULUAN

BAB ini menjelaskan tentang Gambaran umum dari penelitian yang dilakukan. Terdiri dari latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

BAB II LANDASAN TEORI

BAB ini berisi tentang teori-teori pendukung beserta studi-studi penelitian terkait yang berhubungan dengan permasalahan yang akan diteliti sehingga dihasilkan

sebuah *state-of-the art* dan menjadi acuan konseptual terhadap penelitian yang dilakukan

BAB III METODOLOGI

BAB ini memaparkan metode dalam penyelesaian masalah selama melakukan penelitian. Langkah-langkah tersebut memuat studi literatur, pengumpulan data, praproses data, pemodelan YOLOV11 dasar, *hyperparameter tuning*, *export* dan *Quantization*, integrasi model dengan aplikasi dan evaluasi beserta penjelasan lain yang berkaitan dengan penelitian.

BAB IV HASIL DAN PEMBAHASAN

BAB ini berisi hasil dan pembahasan dari penelitian, dimulai dengan pengumpulan data, praproses data, pemodelan YOLOV11 dasar, *hyperparameter tuning*, *export* dan *Quantization*, integrasi model dengan aplikasi dan evaluasi dari penelitian yang sudah dilakukan.

BAB V KESIMPULAN DAN SARAN

BAB ini berisikan kesimpulan dari hasil penelitian yang berupa rangkuman dari pembahasan masalah pada penelitian dan analisis data, serta saran yang dapat diterapkan untuk penelitian selanjutnya berdasarkan keterbatasan yang ada dari penelitian ini.