Syntax Admiration: p-ISSN 2722-7782 | e-ISSN 2722-5356 Vol. 5, No. 11, November 2024

Evaluasi Trade-off Akurasi dan Kecepatan YOLOv5 dalam Deteksi Kebakaran pada Edge Devices

Rahmad Arif Setiawan[1]*, Arief Setyanto[2]
[1,2] Universitas Amikom Yogyakarta, Indonesia
Email: rahmad.arif@students.amikom.ac.id, arief_s@amikom.ac.id

Abstrak

Deteksi objek real-time menggunakan algoritma YOLO (*You Only Look Once*) telah menunjukkan kinerja yang menjanjikan dalam berbagai aplikasi computer vision. Namun, penerapannya pada perangkat dengan sumber daya terbatas masih menjadi tantangan karena kebutuhan komputasi yang tinggi. Penelitian ini bertujuan untuk mengoptimalkan model YOLOv5 untuk deteksi api dan asap pada perangkat Orange Pi Zero 3 menggunakan teknik kuantisasi. Menggunakan dataset 2247 gambar api dan asap, penelitian ini menerapkan teknik kuantisasi statis untuk meningkatkan efisiensi model. Metodologi meliputi pelatihan model YOLOv5 standar, konversi ke format ONNX, dan penerapan kuantisasi statis. Hasil menunjukkan peningkatan signifikan dalam efisiensi komputasi, dengan pengurangan ukuran model sebesar 42.2% dan peningkatan kecepatan inferensi hingga 65.21%. Meskipun terjadi penurunan nilai mAP sebesar 25.6%, model yang dioptimalkan tetap mampu melakukan deteksi objek dengan kecepatan yang signifikan lebih tinggi. Kesimpulannya, teknik kuantisasi efektif dalam mengoptimalkan model YOLOv5 untuk penerapan pada perangkat edge computing, meskipun terdapat trade-off antara kecepatan dan akurasi.

Kata Kunci: YOLOV5, deteksi objek, kuantisasi, edge computing, Orange Pi

Abstract

Real-time object detection using the YOLO (You Only Look Once) algorithm has shown promising performance in various computer vision applications. However, its application on devices with limited resources is still a challenge due to its high computational requirements. This study aims to optimize the YOLOv5 model for fire and smoke detection on Orange Pi Zero 3 devices using quantization techniques. Using a dataset of 2247 fire and smoke images, this study applies static quantization techniques to improve model efficiency. The methodology includes training of standard YOLOv5 models, conversion to ONNX format, and application of static quantization. Results show a significant improvement in computational efficiency, with a 42.2% reduction in model size and a 65.21% increase in inference speed. Despite a decrease in the mAP value by 25.6%, the optimized model was still able to perform object detection at a significantly higher speed. In conclusion, the quantization technique is effective in optimizing the YOLOv5 model for deployment on edge computing devices, despite the trade-off between speed and accuracy.

Keywords: *YOLOv5, object detection, quantization, edge computing, Orange Pi*

Pendahuluan

Informasi visual telah menjadi komponen penting dalam kehidupan manusia modern, memengaruhi segala aspek mulai dari persepsi hingga pengambilan Keputusan (Nalbant & Uyanık, 2021). Kemajuan dalam bidang komputer telah memungkinkan kita untuk mengembangkan sistem yang mampu memahami dan menginterpretasi informasi visual. Salah satu bidang yang paling menonjol adalah *computer vision*.

Computer vision bertujuan untuk meniru kemampuan manusia dalam melihat dan memahami dunia visual (Afni et al., 2021). Salah satu tugas utama dalam computer vision adalah deteksi objek, yakni kemampuan untuk mengidentifikasi dan menentukan lokasi objek tertentu dalam sebuah gambar atau video. Deteksi objek memiliki aplikasi yang luas, mulai dari diagnosis medis hingga kendaraan otonom (Fang et al., 2019).

Deteksi objek, sebuah tantangan klasik dalam computer vision, telah mengalami transformasi signifikan berkat kemajuan deep learning. Dengan dukungan perangkat keras yang mumpuni seperti GPU, algoritma berbasis deep learning seperti YOLO telah berhasil mencapai kinerja deteksi objek yang real-time dengan akurasi tinggi (Nguyen et al., 2019);(Yusup et al., 2024). Namun, ketergantungan YOLO pada GPU yang berdaya tinggi masih menjadi kendala dalam penerapannya pada perangkat dengan sumber daya terbatas

Perangkat dengan keterbatasan CPU dan GPU seperti Raspberry Pi menjadi kendala utama dalam menjalankan algoritma YOLO secara efisien, mengingat kebutuhannya yang tinggi akan memori dan daya komputasi (T. Li et al., 2020). Dalam upaya penerapan algoritma YOLO pada perangkat dengan sumber daya terbatas, berbagai teknik kompresi telah dikembangkan (Putri et al., 2024). Teknik-teknik seperti pruning, yang bertujuan untuk menghilangkan neuron yang tidak relevan, *quantization*, yang mengurangi presisi representasi numerik, dan knowledge distillation, yang memungkinkan transfer pengetahuan dari model yang lebih besar, dapat secara efektif mengurangi kompleksitas komputasi model (Z. Li et al., 2022);(Wahyudi et al., 2024). Selain itu, optimasi desain arsitektur dapat menghasilkan model yang lebih efisien dan disesuaikan dengan kebutuhan perangkat yang berbeda-beda (Jani et al., 2023).

Penelitian ini bertujuan untuk mengembangkan model YOLO yang lebih ringan dan efisien, dengan fokus pada penerapannya pada perangkat berdaya rendah seperti Orange Pi. Melalui penerapan teknik quantization, diharapkan dapat meningkatkan kecepatan deteksi objek dengan tetap mempertahankan akurasi yang memadai (Sholahuddin et al., 2023). Hasil penelitian ini memiliki potensi untuk memperluas penerapan algoritma YOLO pada berbagai aplikasi yang membutuhkan *real-time processing*.

Penelitian ini memberikan manfaat baik secara teoretis maupun praktis. Secara teoretis, penelitian ini memperkaya wawasan terkait optimasi algoritma YOLOv5 dalam konteks edge computing, khususnya melalui teknik kuantisasi untuk

meningkatkan efisiensi model. Secara praktis, hasil penelitian dapat menjadi panduan dalam pengembangan sistem deteksi kebakaran yang efisien dan terjangkau, dengan penerapan pada perangkat berdaya rendah seperti Orange Pi Zero 3. Hal ini diharapkan dapat mendukung upaya deteksi dini kebakaran, yang sangat penting untuk mitigasi risiko dan pengurangan kerugian akibat bencana kebakaran, khususnya di wilayah dengan akses terbatas terhadap infrastruktur komputasi canggih.

Metode Penelitian

Penelitian ini menggunakan pendekatan kualitatif deskriptif untuk menyoroti tantangan praktis dan solusi potensial dalam penerapan algoritma deteksi objek pada perangkat berdaya rendah. Penelitian ini dilakukan dalam jangka waktu tertentu, menggunakan dataset yang secara khusus dikurasi untuk skenario deteksi api dan asap. Penelitian ini difokuskan pada upaya untuk menyeimbangkan dua elemen utama: akurasi deteksi objek dan efisiensi komputasi yang diperlukan untuk aplikasi real-time. Untuk mencapai tujuan ini, penelitian menggunakan perangkat edge, yaitu Orange Pi Zero 3, yang dipilih karena kemampuan komputasinya yang terbatas, menjadikannya kandidat ideal untuk menguji kelayakan penerapan algoritma deteksi canggih dalam lingkungan dengan sumber daya yang terbatas.

Populasi penelitian diwakili oleh dataset yang luas, yang mencakup berbagai skenario kehadiran api dan asap di lingkungan dalam dan luar ruangan. Sampel dipilih secara cermat untuk memastikan dataset ini menyediakan dasar yang komprehensif untuk melatih dan memvalidasi model deteksi objek. Dataset ini dibagi menjadi set pelatihan, validasi, dan pengujian, memastikan bahwa model terekspos pada berbagai kondisi input untuk meningkatkan generalisasi.

Selain itu, penelitian ini didasarkan pada hipotesis bahwa dengan memanfaatkan teknik optimalisasi model seperti kuantisasi, memungkinkan untuk menerapkan model pembelajaran mesin yang canggih pada perangkat berdaya rendah tanpa kehilangan kinerja yang signifikan. Temuan, berdasarkan pengujian di dunia nyata, menunjukkan bahwa hipotesis ini benar, terutama ketika mempertimbangkan aplikasi di mana kecepatan lebih diutamakan daripada akurasi yang sempurna.

Hasil dan Pembahasan

Strategi untuk meningkatkan efisiensi komputasi tanpa mengorbankan akurasi deteksi berpusat pada teknik kuantisasi model. Proses kuantisasi ini melibatkan konversi model yang telah dilatih menjadi format yang lebih ringan, cocok untuk diterapkan pada perangkat dengan daya pemrosesan terbatas. Kuantisasi, dalam konteks ini, menjadi strategi utama untuk mengurangi beban komputasi dan kebutuhan memori, sambil mempertahankan tingkat kinerja deteksi yang memadai. Untuk mengevaluasi keberhasilan strategi ini, serangkaian uji kinerja dilakukan. Uji ini mengukur kecepatan inferensi (seberapa cepat perangkat dapat mengidentifikasi dan mengklasifikasikan objek) serta akurasi model, yang tercermin dalam metrik seperti mean Average Precision (mAP).

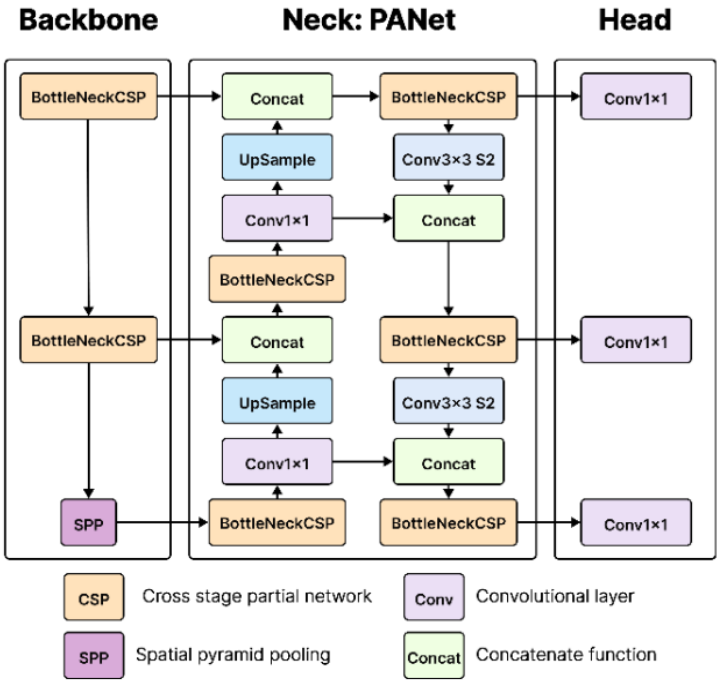
Proses pengujian juga mengeksplorasi trade-off antara ukuran model dan kecepatan, dengan menyadari bahwa pengurangan ukuran model dapat menyebabkan penurunan akurasi deteksi, sebuah fenomena yang didokumentasikan secara menyeluruh dalam hasil. Dengan mengintegrasikan teknik pembelajaran mesin lanjutan dengan keterbatasan dunia nyata, penelitian ini memberikan wawasan berharga tentang penerapan sistem deteksi objek berbasis AI pada perangkat edge. Meskipun pendekatan tradisional sering mengasumsikan ketersediaan perangkat keras yang kuat, penelitian ini berusaha menunjukkan kelayakan teknologi tersebut dalam lingkungan yang lebih terbatas, yang berpotensi memperluas cakupan aplikasi AI di bidang seperti deteksi bencana, keamanan, dan pemantauan lingkungan.

Penelitian ini akan mengambil kasus deteksi dini api dan asap, sehingga dataset yang digunakan merupakan sekumpulan gambar yang berisi api dan asap. Dataset yang digunakan merupakan dataset publik yang dapat diakses di URL : https://universe.roboflow.com/smoke-fire/smokefire-wzrdi. Total gambar yang digunakan adalah 2247 gambar dengan pembagian 70:20:10 (Johnson et al., 2018), yaitu 1573 gambar untuk training, 449 gambar untuk validasi, dan 225 gambar untuk testing (Johnson et al., 2018).

Gambar 1. Sampel Dataset

Algoritma deteksi objek yang digunakan dalam penelitian ini adalah YOLOv5. Versi YOLO ini dipilih karena menawarkan kinerja yang lebih baik pada perangkat dengan spesifikasi rendah dibandingkan dengan versi YOLO yang lebih baru. Meskipun versi terbaru YOLO umumnya memiliki akurasi yang lebih tinggi, YOLOv5 tetap menjadi pilihan yang menarik karena kemampuannya dalam menjalankan deteksi objek secara real-time pada perangkat dengan sumber daya terbatas (Sary et al., 2023).

Arsitektur YOLOv5 terdiri dari tiga komponen utama seperti yang ditunjukkan pada Gambar 2. Backbone yang menggunakan Darknet53 berfungsi mengekstrak fitur-fitur dasar dari gambar. Neck berperan sebagai penghubung antara backbone dan head, menggabungkan dan menyempurnakan fitur-fitur yang telah diekstraksi. Terakhir, head terdiri dari tiga cabang yang masing-masing memprediksi kotak bounding box (bounding boxes) dan kelas objek pada skala yang berbeda (Casas et al., 2024).



Gambar 2. Arsitektur YOLOv5[12]

Proses training atau pelatihan dilakukan menggunakan computasi cloud Paperspace. Sedangkan pengujian akan dilakukan pada perangkat Orange Pi Zero 3 dengan spesifikasi sebagai berikut: Board Type ARM, CPU Allwinnner H618 Quad-Core Cortex-A53 1,5Ghz, RAM LPDDR4 1GB, OS Ubuntu Server, Power via USB Type C 5V 3A.

Gambar 3. Perangkat Orange Pi Zero 3

Pelatihan model YOLOv5 dilakukan dengan 100 epoch menggunakan parameter default (Sary et al., 2023). Model yang telah dilatih kemudian divalidasi untuk menghitung mAP. Selanjutnya, model diuji pada perangkat Orange Pi untuk mengevaluasi kinerja deteksi objek, termasuk waktu inferensi. Tujuannya adalah untuk menilai kemampuan perangkat Orange Pi dalam menjalankan model YOLOv5 secara real-time.

Selanjutnya, untuk meningkatkan efisiensi komputasi model, akan dilakukan teknik quantization. Model yang telah terkompresi kemudian akan dievaluasi ulang untuk membandingkan performanya dengan model YOLOv5 awal, baik dari segi akurasi

deteksi yang diukur menggunakan mAP, waktu inferensi dan ukuran model. Hasil analisis model dilakukan untuk mengetahui nilai mAP, kecepatan inferensi dan ukuran model. Pada proses training awal didapatkan hasil yang cukup tinggi sebagai berikut:

Tabel 1. Hasil pelatihan awal

Nilai mAP@50			Ukuran Model
All	Api	Asap	
0.841	0.914	0.768	3823 KB

Hasil pelatihan awal menghasilkan file best.pt dengan ukuran 3823KB selanjutkan akan digunakan untuk melakukan deteksi objek di perangkat Orange Pi. Dengan menggunakan gambar dari dataset testing didapatkan hasil sebagai berikut :

Tabel 2. Hasil deteksi objek file best.pt

Durasi (ms)				Ukuran Model
Pre-process	Inference	NMS	Total	(KB)
7.46	745.16	17.8	770.43	3823

Quantization merupakan metode untuk mengurangi ketepatan bobot dan aktivasi model deep learning untuk mengurangi kebutuhan memori dan komputasi. Quantization terdiri dari 2 tipe yaitu dynamic quantization dan static quantization. Karena model YOLO sebagian besar berisi lapisan CNN maka penelitian ini menggunakan static quantization. Proses static quantization dilakukan dengan ONNX Runtime, sehingga file best.pt harus di convert terlebih dahulu menjadi file dengan ekstensi ONNX. Setelah dilakukan proses konvert didapatkan file baru dengan nama best.onnx, sebagai perbandingan file tersebut juga dilakukan validasi dan pengujian deteksi dengan hasil sebagai berikut:

Tabel 3. Hasil validasi dan deteksi objek file tipe onnx

Nama File	mAP@50			Durasi (ms)				Ukuran Model
	All	Api	Asap	Pre-Process	Inference	NMS	Total	
best.onnx	0.849	0.918	0.78	8.67	430.20	3.73	442.60	7320

Untuk melakukan kuantisasi statis, perlu menggunakan bagian dari dataset pelatihan sebagai data kalibrasi. Data kalibrasi digunakan untuk menentukan cara terbaik mengubah nilai-nilai dalam model (bobot dan aktivasi) menjadi format 8-bit integer (INT8) (Wang et al., 2023). Setelah proses kuantisasi selesai, dilakukan pengujian model yang telah disederhanakan menggunakan dataset validasi dan pengujian untuk melihat seberapa baik model tersebut masih dapat mendeteksi objek.

Tabel 4. Hasil validasi dan deteksi objek model setelah kuantisasi

Nama File	mAP@50			Durasi (ms)				Ukuran Model
	All	Api	Asap	Pre-Process	Inference	NMS	Total	
quant.onnx	0.625	0.708	0.542	8.67	256.33	3	268	2219

Gambar 4. Hasil Interpretasi model YOLO pada gambar

Ketiga eksperimen yang dilakukan pada dataset api dan asap mengkonfirmasi bahwa penerapan teknik kuantisasi pada model YOLOv5 berhasil mengurangi ukuran model dan waktu inferensi. Meskipun demikian, terdapat penurunan nilai mAP yang signifikan. Detail perbandingan kinerja ketiga model, termasuk nilai mAP, waktu inferensi, dan ukuran model.

Tabel 5. Perbandingan performa model YOLO

Nama File	mAP@50			Durasi (ms)				Ukuran Model
	All	Api	Asap	Pre-Process	Inference	NMS	Total	
best.pt	0.841	0.914	0.768	7.46	745.16	17.8	770.43	3823
best.onnx	0.849	0.918	0.78	8.67	430.20	3.73	442.60	7320
quant.onnx	0.625	0.708	0.542	8.67	256.33	3	268	2219

Berdasarkan analisis menunjukkan adanya trade-off antara akurasi dan kecepatan pada model YOLOv5n. Model standar (best.pt) memiliki akurasi deteksi (mAP) yang lebih baik sebesar 25.6%. Namun, model yang telah dikuantisasi (quant.onnx) menawarkan waktu inferensi yang jauh lebih cepat, yaitu 65.21% lebih singkat, dan ukuran model yang lebih kecil sebesar 42.2%. Hal ini mengindikasikan bahwa model hasil kuantisasi lebih cocok untuk aplikasi yang membutuhkan waktu respon yang cepat dan ukuran model yang kompak, meskipun dengan sedikit penurunan akurasi.

Kesimpulan

Penelitian ini menunjukkan bahwa penerapan teknik kuantisasi pada model YOLOv5 untuk deteksi api dan asap berhasil meningkatkan efisiensi komputasi pada perangkat dengan sumber daya terbatas (edge computing) seperti Orange Pi Zero 3. Beberapa kesimpulan penting yan dapat ditarik:

Implementasi teknik kuantisasi statis berhasil mengurangi ukuran model YOLOv5n standar sebesar 42.2% dan meningkatkan kecepatan inferensi hingga 65.21%. Terdapat trade-off antara efisiensi komputasi serta akurasi deteksi dengan penurutan nilai mAP sebesar 25.6% pada model yang dikuantisasi. Model YOLOv5n yang dioptimalkan

menunjukan potensi yang signifikan untuk aplikasi deteksi api dan asap secara real-time pada perangkat dengan sumber daya terbatas (edge computing).

Konversi model ke format ONNX dan penerapan kuantisasi terbukti efektif dalam mengoptimalkan knierja model pada perangkat edge. Hasil penelitian ini membuka peluang untuk pengembangan sistem deteksi dini kebakaran yang lebih efisien dan terjangkau menggunakan perangkat edge compuing. Namun, diperlukan penelitian lebih lanjut untuk meningkatkan akuras model yang telah dikuantisasi tanpa mengorbankan efisiensi komputasi.

BIBLIOGRAFI

Afni, S. V. N., Silmina, E. P., & Pangestu, I. B. (2021). Computer Vision Used to Monitor The Youth during The Pandemic Covid-19. *Procedia of Engineering and Life Science*, *1*(2).

Casas, E., Ramos, L., Bendek, E., & Rivas-Echeverria, F. (2024). YOLOv5 vs. YOLOv8: Performance Benchmarking in Wildfire and Smoke Detection Scenarios. *Journal of Image and Graphics*, *12*(2). https://doi.org/10.18178/joig.12.2.127-136.

Fang, W., Wang, L., & Ren, P. (2019). Tinier-YOLO: A real-time object detection method for constrained environments. *Ieee Access*, *8*, 1935–1944. https://doi.org/10.1109/ACCESS.2019.2961959.

Jani, M., Fayyad, J., Al-Younes, Y., & Najjaran, H. (2023). Model compression methods for YOLOv5: A review. *ArXiv Preprint ArXiv:2307.11904*.

Johnson, S. J., Blackman, D. A., & Buick, F. (2018). The 70: 20: 10 framework and the transfer of learning. *Human Resource Development Quarterly*, *29*(4), 383–402.

Li, T., Ma, Y., & Endoh, T. (2020). A systematic study of tiny YOLO3 inference: Toward compact brainware processor with less memory and logic gate. *IEEE Access*, *8*, 142931–142955. https://doi.org/10.1109/ACCESS.2020.3013934.

Li, Z., Wang, Y., Chen, K., & Yu, Z. (2022). Channel Pruned YOLOv5-based Deep Learning Approach for Rapid and Accurate Outdoor Obstacles Detection. *ArXiv Preprint ArXiv:2204.13699*.

Nalbant, K. G., & Uyanık, Ş. (2021). Computer vision in the metaverse. *Journal of Metaverse*, *1*(1), 9–12.

Nguyen, D. T., Nguyen, T. N., Kim, H., & Lee, H.-J. (2019). A high-throughput and power-efficient FPGA implementation of YOLO CNN for object detection. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, *27*(8), 1861–1873. https://doi.org/10.1109/TVLSI.2019.2905242.

Putri, A. R., Dewi, R., & Ramiati, R. (2024). Penerapan Metode Yolov5 dan Teknologi Text-To-Speech dalam Aplikasi Pengenalan Abjad dan Objek Sekitar untuk Anak Usia Dini. *Elektron: Jurnal Ilmiah*, 94–101.

Sary, I. P., Andromeda, S., & Armin, E. U. (2023). Performance Comparison of YOLOv5 and YOLOv8 Architectures in Human Detection using Aerial Images. *Ultima Computing: Jurnal Sistem Komputer*, *15*(1), 8–13. https://doi.org/10.1109/CVPR.2016.91.

Sholahuddin, M. R., Harika, M., Awaludin, I., Dewi, Y. C., Fauzan, F. D., Sudimulya, B. P., & Widarta, V. P. (2023). Optimizing YOLOv8 for Real-Time CCTV Surveillance: A Trade-off Between Speed and Accuracy. *Jurnal Online Informatika*, *8*(2), 261–270. https://doi.org/10.15575/join.v8i2.1196.

Wahyudi, A. A., Khumaidi, A., Rahmat, M. B., Riananda, D. P., Syai'in, M., &

Endrasmono, J. (2024). Implementasi Robot Operating System (ROS) Untuk Meningkatkan Akurasi Deteksi Bola Menggunakan YOLO V5 Pada KRSBI-Beroda. *Jurnal Elektronika Dan Otomasi Industri*, *11*(2), 590–603.

Wang, M., Sun, H., Shi, J., Liu, X., Cao, X., Zhang, L., & Zhang, B. (2023). Q-YOLO: Efficient inference for real-time object detection. *Asian Conference on Pattern Recognition*, 307–321.

Yusup, R. M., Anugrah, A. F., Muslimah, D. D., Permana, S. M. W. N., & Yuliani, S. (2024). PENDETEKSIAN OBJEK MENGGUNAKAN OPENCV DAN METODE YOLOv4-TINY UNTUK MEMBANTU TUNANETRA. *Journal of Computer Science and Information Technology*, *1*(2), 59–68.