

Pengembangan Sistem Deteksi *On-Shelf Availability* Produk Menggunakan Algoritma YOLOv8 pada Aplikasi Bergerak

Gabriel Imam Andaru ^{1*}, Dhomas Hatta Fudholi ²

^{1,2} Program Studi Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia, Kabupaten Sleman, Daerah Istimewa Yogyakarta, Indonesia.

Email: 20523239@students.uui.ac.id ^{1*}, hatta.fudholi@uui.ac.id ²

Histori Artikel:

Dikirim 18 April 2024; Diterima dalam bentuk revisi 24 April 2024; Diterima 5 Mei 2024; Diterbitkan 20 Mei 2024. Semua hak dilindungi oleh Lembaga Penelitian dan Pengabdian Masyarakat (LPPM) STMIK Indonesia Banda Aceh.

Abstrak

Penelitian terkait operasi ritel telah menjadi sorotan utama dalam beberapa tahun terakhir, didorong oleh dinamika pasar yang cepat berubah dan pentingnya ketersediaan produk di rak toko untuk memenuhi kepuasan pelanggan. Konsep OSA (On Shelf Availability) menjadi kunci dalam memastikan produk tersedia ketika dibutuhkan. Namun, tantangan dalam manajemen ritel adalah memantau ribuan produk yang berbeda, yang memakan waktu dan sumber daya. Untuk mengatasi masalah ini, diperlukan solusi deteksi objek yang efisien. Penelitian mengimplementasikan algoritma YOLOv8 dalam mendeteksi kehabisan stok, khususnya dalam perangkat bergerak yang memiliki keterbatasan sumber daya. Dalam rangka mencapai tujuan ini, penelitian ini mengadopsi metodologi yang komprehensif, dimulai dari pengumpulan data langsung dari supermarket, pemrosesan data, pelabelan, hingga pelatihan model menggunakan teknik transfer learning. Metode transfer learning dipilih dengan pertimbangan untuk mengatasi keterbatasan data dan mempercepat proses pelatihan model, sehingga memungkinkan adaptasi yang lebih cepat terhadap kondisi deteksi objek pada lokasi spesifik. Hasil pengujian menunjukkan bahwa YOLOv8s memberikan performa terbaik dengan akurasi mencapai 94,7%, memungkinkan deteksi objek secara real-time. Pengujian dilakukan pada berbagai perangkat bergerak, termasuk Samsung A54 dan Samsung A6, di mana YOLOv8n mampu memberikan performa yang konsisten dengan waktu inferensi 41.46 ms pada Samsung A54 dan 257.73 ms pada Samsung A6. Kontribusi utama penelitian ini adalah meningkatkan kemampuan deteksi objek pada perangkat dengan daya komputasi rendah, seperti perangkat bergerak, serta memberikan solusi untuk masalah ketersediaan produk di rak toko secara efektif. Dengan demikian, penelitian ini tidak hanya membawa dampak positif bagi manajemen ritel tetapi juga mendorong perkembangan teknologi deteksi objek dalam perangkat dengan sumber daya terbatas dan koneksi internet yang tidak stabil.

Kata Kunci: Operasi Ritel; Ketersediaan Produk; Deteksi Objek; YOLOv8; Perangkat Bergerak.

Abstract

Research related to retail operations has been a major focus in recent years, driven by rapidly changing market dynamics and the importance of product availability on store shelves to meet customer satisfaction. The concept of On Shelf Availability (OSA) has become key in ensuring products are available when needed. However, the challenge in retail management lies in monitoring thousands of different products, which is time-consuming and resource-intensive. To address this issue, an efficient object detection solution is needed. The research implements the YOLOv8 algorithm in detecting out-of-stock items, particularly in the context of mobile devices with resource limitations. In order to achieve this goal, the research adopts a comprehensive methodology, starting from direct data collection from supermarkets, data processing, labeling, to model training using transfer learning techniques. Transfer learning method is chosen to overcome data limitations and accelerate the model training process, enabling faster adaptation to object detection conditions at specific locations. Test results show that YOLOv8s delivers the best performance with an accuracy of up to 94.7%, allowing real-time object detection. Testing is conducted on various mobile devices, including Samsung A54 and Samsung A6, where YOLOv8n consistently performs with an inference time of 41.46 ms on Samsung A54 and 257.73 ms on Samsung A6. The main contribution of this research is to enhance object detection capabilities on devices with low computational power, such as mobile devices, and provide an effective solution to the problem of product availability on store shelves. Thus, this research not only brings positive impact to retail management but also drives the development of object detection technology in the context of resource-limited devices and unstable internet connections.

Keyword: Retail Operations; Product Availability; Object Detection; YOLOv8; Mobile Devices.

1. Pendahuluan

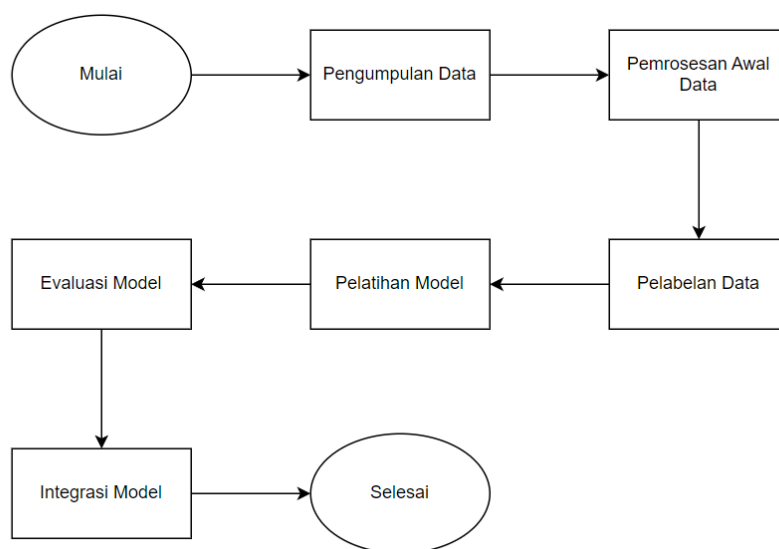
Penelitian mengenai operasi ritel telah menjadi fokus yang signifikan dalam beberapa tahun terakhir, didorong oleh berbagai faktor salah satunya karena sektor ekonomi vital di mana perubahan pasar, perusahaan, proses, dan produk terjadi dengan cepat (Hübner *et al.*, 2018). Dalam operasi ritel, penting bagi pelanggan untuk dapat dengan mudah menemukan barang belanjaan yang mereka inginkan di rak toko. Manajemen ritel sendiri menjadi tantangan karena berbagai hal yang harus dikelola, seperti ketersediaan barang dan penataan rak. Konsep OSA (*On Shelf Availability*) digunakan untuk mengukur seberapa mungkin suatu produk tersedia dalam kondisi yang siap dijual ketika pelanggan mencarinya di rak (Chopra & Meindl, 2002). Menjamin "*On Shelf Availability*" (OSA) atau menghindari kejadian "*Out of Stock*" (OOS) di toko merupakan faktor kunci bagi kepuasan pelanggan (García-Arca *et al.*, 2020). Untuk memastikan tingkat ketersediaan barang yang tinggi di rak, staf toko harus aktif berkeliling di sekitar toko untuk mencari produk yang perlu diisi ulang serta memeriksa produk yang mungkin ditempatkan dengan tidak benar (Afram, 2020). Namun, tugas ini cukup memakan waktu karena biasanya ada banyak sekali produk yang harus dipantau di dalam toko. Sebuah supermarket biasanya memiliki lebih dari ribuan produk yang berbeda (Goldman *et al.*, 2019). Oleh karena itu, penerapan sistem deteksi kehabisan stok yang berkinerja tinggi tidak hanya akan meningkatkan volume penjualan, tetapi juga meringankan pekerja dari tugas pemeriksaan visual dan memastikan kemungkinan untuk mempekerjakan mereka dalam tugas manajemen toko lainnya (Šikić *et al.*, 2024).

Kemajuan *artificial intelligence* (AI) telah sangat pesat dalam beberapa tahun terakhir, didorong oleh potensi yang besar dalam berbagai aplikasi. Terdapat tren baru dalam menerapkan pembelajaran mesin pada sistem yang ringan dan tertanam, seperti gawai, dengan tujuan mobilitas tinggi, biaya rendah, implementasi cepat, dan manfaat lainnya. Dengan meningkatnya minat dalam sistem operasi Android, beberapa kerangka kerja pembelajaran mendalam yang populer telah dipindahkan ke platform ini, termasuk *TensorFlow Mobile* (TFM), *TensorFlow Lite* (TFL), *OpenCV*, dan *Qualcomm Snapdragon*. Platform-platform ini dirancang untuk menangani tugas inferensi di ponsel cerdas, ideal untuk situasi di mana konektivitas buruk atau tidak tersedia (Martinez-Alpiste *et al.*, 2022). Deteksi objek adalah salah satu tugas kunci dalam bidang visi komputer yang bertujuan untuk menemukan posisi objek tertentu. Hal ini sangat berguna dalam ritel, seperti mengidentifikasi produk di rak untuk memberikan informasi tentang ulasan atau harga, dan dalam navigasi di supermarket untuk meningkatkan penjualan. Sebuah penelitian (K. Li *et al.*, 2021) membahas penggunaan deteksi objek dalam memantau jumlah produk di rak, mengisi produk yang hilang, dan menjaga konsistensi dengan planogram menggunakan YOLOv2. Penelitian lain (Sinha *et al.*, 2022) menggabungkan pendekatan Faster R-CNN untuk deteksi objek dan Resnet-18 untuk klasifikasi objek, dengan memanfaatkan dataset GroZi 120k (George & Floerkemeier, 2014) dan GP-180 (Sinha *et al.*, 2022; Tonioni & Di Stefano, 2017). Dari seluruh penelitian tersebut, mereka berhasil mengimplementasikan berbagai algoritma deteksi objek dan mendapatkan hasil yang cukup baik. Namun, terdapat beberapa kekurangan dari penelitian tersebut seperti jumlah kelas yang relatif cukup sedikit dan penggunaan algoritma tidak dapat berjalan pada gawai atau perangkat bergerak. Oleh karena itu, penelitian ini berfokus kepada pengenalan produk objek dengan jumlah kelas yang cukup banyak dengan memanfaatkan algoritma YOLOv8 dan penggunaan algoritma tersebut dalam perangkat bergerak.

2. Metode Penelitian

Penelitian ini menggunakan algoritma YOLOv8 yang saat ini merupakan versi terbaru dari keluarga YOLO (You Look Only Once). Model YOLOv8, yang didasarkan pada arsitektur YOLOv5, melanjutkan kemajuan ini dengan menggabungkan inovasi tambahan dalam teknologi deteksi objek. YOLOv8 mengadopsi beberapa peningkatan dari model YOLOv5, seperti waktu pelatihan yang lebih cepat, akurasi yang lebih tinggi, dan kinerja yang lebih baik pada perangkat keras yang lebih kecil

(Joher *et al.*, 2023). YOLOv8 menggunakan pendekatan di mana tidak menggunakan *anchor* dalam modelnya. Model ini memiliki cabang-cabang terpisah untuk melakukan deteksi objek, klasifikasi, dan regresi secara terpisah. Dengan pendekatan ini, setiap cabang dapat fokus pada tugasnya sendiri, yang dapat meningkatkan akurasi total modelnya (Wu & Dong, 2023). CIoU (Zheng *et al.*, 2020) dan fungsi *loss* DFL (X. Li *et al.*, 2020) digunakan oleh YOLOv8 untuk *box loss*, dan *binary cross-entropy* digunakan untuk *cls loss*. Fungsi-fungsi *loss* ini telah menunjukkan peningkatan kinerja deteksi objek, terutama dalam skenario yang melibatkan objek yang lebih kecil. Kemudian, untuk alur penelitian dapat dilihat pada Gambar 1.



Gambar 1. Alur Penelitian

2.1 Pengumpulan Data

Dataset yang dipakai terdiri dari serangkaian gambar yang menampilkan beberapa kelas produk di rak toko. Terdapat 135 kelas yang diwakili dalam dataset ini. Untuk membangun model yang akurat, penting memiliki dataset yang serupa dengan data uji. Oleh karena itu, pada penelitian ini gambar diambil langsung di supermarket menggunakan perangkat bergerak sebagai bagian dari proses pengumpulan data.

2.2 Pemrosesan Awal Data

Pada tahap ini, penelitian menggunakan data gambar yang telah terkumpul, kemudian melakukan proses awal pengolahan agar gambar bisa dimasukkan ke dalam model. Proses ini mencakup mengubah ukuran gambar menjadi 640x640 piksel.

2.3 Pelabelan Data

Data gambar yang telah melewati tahap pemrosesan awal akan diberi label untuk memberikan kategori atau kelas objek sebelum dilatih menggunakan model. Proses pelabelan dilakukan secara manual pada setiap gambar menggunakan alat "*LabelImg*". Anotasi dilakukan dengan menarik kotak di sekitar setiap objek dan memberikan nama kelas. Setiap objek akan memiliki koordinat dua titik (x, y) yang menunjukkan sudut kiri atas dan sudut kanan bawah dari kotak tersebut, memberikan informasi tentang lokasi dan ukuran objek dalam format anotasi YOLOv8 dibawah ini.

< class > < x_center > < y_center > < width > < height >

2.4 Pelatihan Model

Pada langkah ini, gambar yang telah diberi label atau anotasi akan disiapkan untuk pelatihan menggunakan arsitektur model YOLOv8. Tujuannya adalah agar model dapat mempelajari pola atau karakteristik dari data gambar yang telah disiapkan, dengan harapan menghasilkan model yang mampu mendeteksi objek sesuai kelasnya dengan tingkat akurasi yang tinggi dan dapat dijalankan pada perangkat bergerak. Dalam hal ini, penulis memilih menggunakan versi YOLOv8n dan YOLOv8s karena jarak waktu inferensinya lebih sedikit dibandingkan YOLOv8m. Model dilatih dengan ukuran *batch* sebesar 32 karena hal tersebut dapat mengurangi jumlah pembaruan yang diperlukan untuk mencapai konvergensi (Bengio, 2012). Sebaliknya, jumlah *epoch* sebanyak 300 ditetapkan untuk mengidentifikasi potensi *overfitting* atau *underfitting* pada model yang dikembangkan. Pada penelitian ini model dilatih dengan menggunakan teknik *transfer learning*. Metode ini dapat mengurangi waktu dan sumber daya komputasi yang dibutuhkan, karena model yang sudah dilatih sebelumnya telah memperoleh kemampuan untuk mengekstraksi fitur dari gambar. Selain itu, *transfer learning* dalam deteksi objek juga dapat membantu mengatasi masalah data terbatas yang telah diberi label (Vasconcelos *et al.*, 2022). Saat menggunakan YOLOv8 untuk *transfer learning* dalam deteksi objek, penting untuk memperhatikan penggunaan bobot dalam proses tersebut. Model YOLOv8 yang telah dilatih sebelumnya memiliki bobot pretrain yang berasal dari dataset Microsoft COCO (Lin Tsung-Yi and Maire, 2014). Bobot ini sangat berharga dalam mengekstrak fitur dari gambar dan dapat menjadi titik awal yang kuat untuk penyesuaian kembali pada dataset baru yang spesifik (Ganesh *et al.*, 2021).

2.5 Evaluasi Model

Setelah model dilatih, evaluasi dilakukan menggunakan berbagai metrik seperti mAP (*mean Average Precision*) dan metrik lainnya. Metrik evaluasi ini berguna untuk mengevaluasi kinerja model deteksi objek pada dataset yang telah diatur ulang. Metrik mAP memberikan gambaran menyeluruh tentang presisi dan recall model dalam berbagai kategori objek. Selain itu, metrik seperti presisi, recall, dan skor F1 juga dapat digunakan untuk menilai kinerja model dalam mendeteksi dan memposisikan objek dalam gambar. Proses evaluasi sangat penting untuk memahami seberapa baik model YOLOv8 yang telah diatur ulang berkinerja pada dataset baru dan mengidentifikasi area-area yang perlu ditingkatkan.

2.6 Integrasi Model

Dalam penelitian ini, digunakan format NCNN yang dikembangkan oleh Tencent karena performanya yang unggul dalam komputasi *neural network* (Lawal *et al.*, 2023). NCNN telah dikembangkan dan diunggah ke *ncnn-android-yolov8* (FeiGeChuanShu & Q-Engineering, 2023), dan digunakan pada perangkat Huawei Nova 10 Pro. Integrasi model YOLOv8 dilakukan dengan mengubah format model yang telah dikembangkan dari *PyTorch* (pt) ke *Open Neural Network Exchange* (ONNX), dan kemudian ke NCNN. Sementara ONNX dirancang sebagai format perantara antara model, NCNN dioptimalkan agar dapat berjalan lebih cepat daripada kerangka kerja yang umumnya digunakan pada CPU ponsel. Integrasi dilakukan pada perangkat mobile Samsung A6 dengan prosesor Exynos 7870 octa-core 1,6 GHz dan RAM 3GB, serta Samsung A54 dengan RAM 6GB dan prosesor Octa-core (4x2,4 GHz Cortex-A78 & 4x2,0 GHz Cortex-A55) untuk deteksi objek.

3. Hasil dan Pembahasan

3.1 Pengolahan Dataset

Data dikumpulkan dengan mempertimbangkan kriteria seleksi toko ritel dan metode pengambilan foto langsung di lapangan, menghasilkan total 11.216 foto. Foto-foto dari berbagai produk diambil melalui kunjungan langsung ke toko ritel yang dipilih, memastikan kondisi sesuai dengan aturan pengambilan foto. Manajemen data dilakukan dengan menyimpan foto-foto dalam

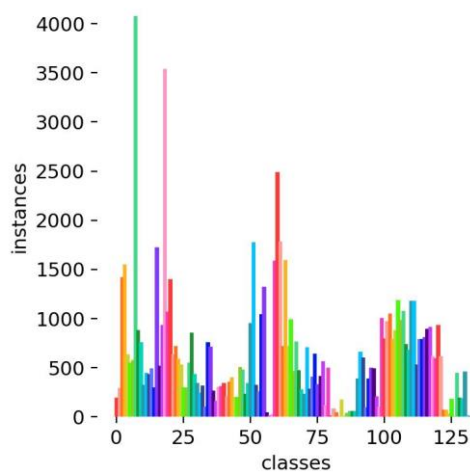
struktur folder terorganisir dan menjaga privasi serta kerahasiaan informasi dari toko ritel tersebut. Contoh gambar dapat dilihat pada gambar 2.



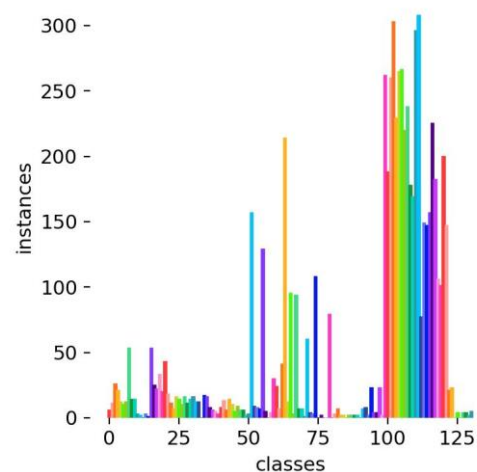
Gambar 2. Contoh Pengambilan Foto Rak

3.2 Pemrosesan Awal Data

Proses awal pemrosesan data melibatkan *resizing* setiap gambar dalam dataset. Sebelumnya, gambar-gambar memiliki dimensi yang bervariasi, seperti 4080x3060, 1024x768, dan 2040x1530. Untuk mengoptimalkan ukuran gambar sesuai dengan kebutuhan komputasi pada pengembangan model deteksi objek, dilakukan *resizing* menjadi dimensi 640x640. Pilihan dimensi tersebut didasarkan pada upaya mencapai keseimbangan antara akurasi deteksi dan efisiensi komputasi. Selain itu, *resizing* ke dimensi 640x640 dipilih untuk menyederhanakan representasi visual objek tanpa kehilangan informasi kritis, memungkinkan model beroperasi secara efisien pada sumber daya komputasi yang terbatas. Selanjutnya, data dipilih secara acak dan dibagi menjadi dua kelompok dengan mempertahankan proporsi 90% untuk data pelatihan dan 10% untuk data validasi. Dari pembagian dataset tersebut, didapatkan 10.601 foto untuk data pelatihan dan 615 foto untuk data validasi. Data pada gambar pelatihan terdiri dari total 59.235 objek dengan distribusi yang sesuai dengan gambar 3, sedangkan pada gambar validasi terdapat 6.619 objek dengan distribusi seperti yang terlihat pada gambar 4.



Gambar 3. Distribusi Data Latih



Gambar 4. Distribusi Data Evaluasi

3.3 Hasil Pelatihan

Hasil pelatihan YOLOv8n menunjukkan peningkatan performa model selama proses pelatihan, terbukti dari grafik yang menunjukkan penurunan *loss* pelatihan dan validasi. Nilai *box loss* pelatihan turun dari 0.75 menjadi 0.45, sementara *box loss* validasi turun dari 0.5 menjadi kurang dari 0.35. Penurunan ini menunjukkan peningkatan kemampuan model dalam memprediksi *bounding box* yang akurat untuk objek dalam gambar. Selain itu, nilai *classification loss* pelatihan turun dari 4.0 menjadi kurang dari 1, dan pada validasi turun dari 2.0 menjadi kurang dari 0.5, menunjukkan peningkatan kemampuan model dalam memprediksi label kelas. Nilai *dfl loss* pelatihan turun dari 0.98 menjadi 0.9, dan pada validasi turun dari 0.92 menjadi 0.84, menunjukkan peningkatan kemampuan model dalam menangani ketidakseimbangan kelas.

Selain itu, terdapat peningkatan nilai mAP50, *precision*, dan *recall* model. mAP50 naik dari 0.2 menjadi 0.8, *precision* naik dari 0.4 menjadi 0.9, dan *recall* naik dari 0.3 menjadi 0.9. Peningkatan ini menunjukkan peningkatan akurasi model dalam mendeteksi objek dan mengurangi kemungkinan prediksi yang salah. Model dilatih selama 300 *epoch* tetapi dihentikan pada epoch ke-135 karena tidak ada penurunan *loss* lebih lanjut. Setelah pelatihan, YOLOv8n menghasilkan file *best.pt* dengan ukuran 6.93 MB, memiliki 168 *layer*, 3.351.329 parameter, waktu inferensi 19 ms, dan 9.7 GFLOPs. Proses pelatihan YOLOv8s juga menghasilkan peningkatan performa model. *Box loss* pelatihan turun dari 0.7 menjadi 0.45, sedangkan *box loss* validasi turun dari 0.425 menjadi 0.3. *Classification loss* pelatihan turun dari 2.5 menjadi 0.5, dan pada validasi turun dari 1.0 menjadi kurang dari 0.2. Selain itu, *dfl loss* pelatihan turun dari 1 menjadi 0.875, dan pada validasi turun dari 0.86 menjadi 0.81. Peningkatan juga terjadi pada nilai mAP50, *precision*, dan *recall*. mAP50 naik dari 0.6 menjadi 0.9, *precision* naik dari 0.65 menjadi 0.95, dan *recall* naik dari 0.6 menjadi 0.9. Model dihentikan pada epoch ke-110 setelah dilatih selama 300 *epoch*. Setelah pelatihan, YOLOv8s menghasilkan file *best.pt* dengan ukuran 22.6 MB, memiliki 168 *layer*, 11.177.829 parameter, waktu inferensi 19.9 ms, dan 28.7 GFLOPs.

3.4 Evaluasi Model

Model YOLOv8 telah dievaluasi menggunakan data validasi yang telah dipisahkan sebelumnya, terdiri dari 615 gambar dan 6619 objek, dengan menggunakan GPU NVIDIA Tesla V100. Hasil evaluasi ini telah dirangkum dalam Tabel 1. Selain itu, terdapat hasil inferensi antara model YOLOv8n dan YOLOv8s yang ditampilkan pada gambar 5.

Tabel 1. Hasil Evaluasi Model YOLOv8

Model	Waktu Inferensi	mAP50	mAP50:95	Presisi	Recall
YOLOv8n	19 ms	0.961	0.895	0.922	0.918
YOLOv8s	19.9 ms	0.969	0.905	0.92	0.927



Gambar 5. Hasil Deteksi YOLOv8n (kanan) dan YOLOv8s (kiri)

Hasil dari penelitian menunjukkan peningkatan performa model YOLOv8, khususnya YOLOv8n dan YOLOv8s, dengan adanya penurunan *loss* dan peningkatan *mean Average Precision* (mAP), *precision*, dan *recall*. Evaluasi model dilakukan dengan memperhitungkan berbagai metrik, termasuk mAP50 yang digunakan sebagai ukuran utama dalam mengevaluasi akurasi deteksi objek. Dari hasil evaluasi, YOLOv8s menunjukkan kinerja yang lebih baik dalam mendeteksi objek, dengan mAP50 yang lebih tinggi dibandingkan dengan YOLOv8n.

3.5 Integrasi Model

Selain itu, penelitian ini juga memperhitungkan integrasi model ke dalam perangkat bergerak dan GPU untuk mengevaluasi waktu inferensi yang berbeda-beda tergantung pada spesifikasi perangkat yang digunakan. Hasilnya menunjukkan bahwa performa deteksi objek dapat bervariasi, tergantung pada kemampuan perangkat keras yang digunakan, dengan waktu inferensi yang lebih baik pada perangkat GPU dibandingkan dengan perangkat bergerak yang dapat dilihat pada Tabel 2.

Tabel 2. Hasil Inferensi Model

Model	Waktu Inferensi NVIDIA Tesla V100	Waktu Inferensi Samsung A54	Waktu Inferensi Samsung A6	mAP50 PyTorch Model	mAP50 NCNN Model
YOLOv8n	19 ms	41.46 ms	257.73 ms	0.961	0.947
YOLOv8s	19.9 ms	75.19 ms	341.3 ms	0.969	0.962

Berdasarkan hasil tersebut, terlihat bahwa mAP pada model YOLOv8s lebih baik daripada YOLOv8n, meskipun perbedaannya tidak signifikan dalam perbandingan antara model PyTorch dan NCNN. Namun, saat membandingkan waktu inferensi pada berbagai perangkat, terlihat bahwa NVIDIA Tesla V100 dapat menjalankan inferensi dalam waktu 19 ms, sementara perangkat bergerak seperti Samsung A54 jauh lebih cepat daripada Samsung A6. Waktu inferensi untuk YOLOv8n dan YOLOv8s pada Samsung A54 adalah 41.46 ms dan 75.19 ms, sedangkan pada Samsung A6 adalah 257.73 ms dan 341.3 ms. Perbedaan ini disebabkan oleh kemampuan CPU yang lebih baik pada Samsung A54 daripada Samsung A6.

4. Kesimpulan

Dari hasil penelitian, beberapa kesimpulan dapat diambil sebagai berikut, penelitian berhasil mengimplementasikan YOLOv8 untuk deteksi ketersediaan produk, baik pada perangkat bergerak maupun GPU. Model yang diuji meliputi YOLOv8n dan YOLOv8s, dengan YOLOv8s menunjukkan hasil deteksi terbaik, memiliki mAP50 0.969 untuk perangkat GPU dan 0.962 untuk perangkat bergerak. YOLOv8n memiliki waktu inferensi yang sedikit lebih cepat daripada YOLOv8s, dengan perbedaan hanya 0.09 ms. Namun, waktu inferensi pada perangkat bergerak lebih lama daripada perangkat GPU. Misalnya, Samsung A54 membutuhkan waktu 41.46 ms untuk YOLOv8n dan 75.19 ms untuk YOLOv8s, sedangkan Samsung A6 memerlukan waktu 257.73 ms untuk YOLOv8n dan 341.3 ms untuk YOLOv8s. Terbukti bahwa kemampuan komputasi pada Samsung A54 lebih unggul dibandingkan Samsung A6, terutama dalam melakukan inferensi model YOLOv8n dan YOLOv8s. Pengujian dilakukan menggunakan perangkat keras Linux dengan GPU NVIDIA Tesla V100, serta perangkat bergerak Samsung A54 dan Samsung A6. Penelitian ini memberikan kontribusi penting dalam meningkatkan kemampuan deteksi objek pada perangkat bergerak, yang sering kali memiliki keterbatasan dalam hal daya pemrosesan dan sumber daya lainnya. Dengan demikian, penelitian ini dapat menjadi landasan yang kuat untuk pengembangan lebih lanjut dalam bidang deteksi objek, terutama dalam pengelolaan stok dan inventarisasi produk pada perangkat bergerak.

5. Ucapan Terima Kasih

Terima kasih penulis ucapkan kepada teman-teman penulis yang berada di kampus yang telah membantu penulis untuk mengumpulkan dataset dan juga memberikan dorongan moral dan mental dalam menyelesaikan artikel ini.

6. Daftar Pustaka

- Afram, G. (2020). A study on how to improve On-shelf availability: With deep learning.
- Bengio, Y. (2012). Practical recommendations for gradient-based training of deep architectures. In *Neural networks: Tricks of the trade: Second edition* (pp. 437-478). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Chopra, S., & Meindl, P. (2007). *Supply chain management. Strategy, planning & operation* (pp. 265-275). Gabler.
- Ganesh, P., Chen, Y., Yang, Y., Chen, D., & Winslett, M. (2022). YOLO-ReT: Towards high accuracy real-time object detection on edge GPUs. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (pp. 3267-3277).
- García-Arca, J., Prado-Prado, J. C., & Garrido, A. T. G. P. (2020). On-shelf availability and logistics rationalization. A participative methodology for supply chain improvement. *Journal of Retailing and Consumer Services*, 52, 101889. DOI: <https://doi.org/10.1016/j.jretconser.2019.101889>.
- George, M., & Floerkemeier, C. (2014). Recognizing products: A per-exemplar multi-label image classification approach. In *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part II 13* (pp. 440-455). Springer International Publishing.
- Goldman, E., Herzig, R., Eisenschtat, A., Goldberger, J., & Hassner, T. (2019). Precise detection in densely packed scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 5227-5236).
- Hübner, A., Amorim, P., Kuhn, H., Minner, S., & Van Woensel, T. (2018). Retail operations. *OR Spectrum*, 40, 831-835.
- Lawal, O. M., Zhu, S., Cheng, K., & Liu, C. (2023). A simplified network topology for fruit detection, counting and mobile-phone deployment. *Plos one*, 18(10), e0292600. DOI: <https://doi.org/10.1371/journal.pone.0292600>.
- Li, K., Zhu, J., & Li, N. (2021, May). Insect detection and counting based on YOLOv3 model. In *2021 IEEE 4th International Conference on Electronics Technology (ICET)* (pp. 1229-1233). IEEE. DOI: <https://doi.org/10.1109/ICET51757.2021.9450898>.
- Li, X., Wang, W., Wu, L., Chen, S., Hu, X., Li, J., ... & Yang, J. (2020). Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. *Advances in Neural Information Processing Systems*, 33, 21002-21012.

- Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13* (pp. 740-755). Springer International Publishing.
- Martinez-Alpiste, I., Golcarenenji, G., Wang, Q., & Alcaraz-Calero, J. M. (2022). Smartphone-based real-time object recognition architecture for portable and constrained systems. *Journal of Real-Time Image Processing*, 19(1), 103-115.
- Šikić, F., Kalafatić, Z., Subašić, M., & Lončarić, S. (2024). Enhanced Out-of-Stock Detection in Retail Shelf Images Based on Deep Learning. *Sensors*, 24(2), 693. DOI: <https://doi.org/10.3390/s24020693>.
- Sinha, A., Banerjee, S., & Chattopadhyay, P. (2022). An improved deep learning approach for product recognition on racks in retail stores. *arXiv preprint arXiv:2202.13081*.
- Tonioni, A., & Di Stefano, L. (2017). Product recognition in store shelves as a sub-graph isomorphism problem. In *Image Analysis and Processing-ICLAP 2017: 19th International Conference, Catania, Italy, September 11-15, 2017, Proceedings, Part I 19* (pp. 682-693). Springer International Publishing.
- Vasconcelos, C., Birodkar, V., & Dumoulin, V. (2022). Proper reuse of image classification features improves object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 13628-13637).
- Wu, T., & Dong, Y. (2023). YOLO-SE: Improved YOLOv8 for remote sensing object detection and recognition. *Applied Sciences*, 13(24), 12977. DOI: <https://doi.org/10.3390/app132412977>.
- Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., & Ren, D. (2020, April). Distance-IoU loss: Faster and better learning for bounding box regression. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 34, No. 07, pp. 12993-13000). DOI: <https://doi.org/10.1609/aaai.v34i07.6999>.