

Deteksi Objek Pohon Secara *Real-Time* dari Udara Menggunakan YOLOv8 dengan Pendekatan *Fine-Tuning* pada *Unmanned Aerial Vehicle* (UAV) untuk Aplikasi Penentuan Kerapatan Vegetasi Hutan

Adiyasa Pratama Putra, Maria Evita^{*}

Departemen Fisika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Institut Teknologi Bandung
Jl. Ganesha No. 10, Bandung, Jawa Barat 40132, Indonesia

Info Artikel

Histori Artikel:

Diajukan: 23 Januari 2026
Direvisi: 18 April 2026
Diterima: 30 Juli 2026

Kata kunci:

Deteksi Objek
Metode *Fine-Tuning*
Pemantauan Hutan
UAV
YOLOv8

Keywords:

forest monitoring
fine-tuning method
object detection
UAV
YOLOv8

Penulis Korespondensi:

Maria Evita
Email: mariaevida19@itb.ac.id

ABSTRAK

Penelitian memperkenalkan berbagai arsitektur YOLOv8 untuk melakukan pendeteksian objek pohon sebagai langkah menentukan kerapatan vegetasi menggunakan UAV. Tujuan penelitian adalah mengevaluasi arsitektur YOLO paling optimal yang dapat digunakan untuk mendeteksi objek pohon dalam kerapatan vegetasi hutan secara *real-time* dengan meninjau performa model serta waktu inferensi yang diperlukan model terkait. Pelatihan dilakukan dengan menggunakan sebanyak 2273 gambar dengan detail sebagai berikut: untuk pelatihan berjumlah 2073 gambar, untuk pengujian 67 gambar, dan untuk validasi sebanyak 133 gambar. Pada model terbaik, dilakukan pelatihan lanjutan menggunakan metode *fine-tuning* untuk meminimalkan kesalahan pembacaan yang dilakukan oleh model. Berdasarkan pelatihan dan pengujian, didapatkan model paling optimal digunakan untuk pemantauan hutan secara *real-time* adalah model pada YOLOv8n dengan nilai *F1-score* mencapai 0,961 pada pengujian dengan ketinggian 70 m. Ketinggian penerbangan arsitektur YOLO paling optimal untuk melakukan pemantauan adalah pada ketinggian 70 m, ditunjukkan dengan nilai *F1-score* paling tinggi pada arsitektur YOLOv8l sebesar 0,973. Penggunaan metode *fine-tuning* terbukti dapat mengurangi kesalahan pembacaan objek oleh arsitektur YOLO secara signifikan.

This study introduces various YOLOv8 architectures for detecting tree objects as a step in determining vegetation density using UAVs. The objective of this study is to investigate the optimal YOLO architecture that can be used to detect tree objects in dense forest vegetation in real-time by reviewing model performance and the inference time required by the related model. Training was conducted using 2273 images with the following details: 2073 training images, 67 testing images, and 133 validation images. For the best model, further training was carried out using a fine-tuning method to minimize reading errors made by the model. Based on training and testing, the most optimal model used for real-time forest monitoring was the YOLOv8n model with F1-score of 0.961 in testing at a height of 70 m. The most optimal flight altitude of the YOLO architecture for monitoring is at a height of 70 m, indicated by the highest F1-score in the YOLOv8l architecture of 0.973. The use of the fine-tuning method has been proven to significantly reduce object readings by the YOLO architecture.

Copyright © 2026 Author(s).

I. PENDAHULUAN

Hutan berperan penting menjaga stabilitas biosfer melalui penyediaan oksigen (O_2), penyerapan karbon dioksida (CO_2), serta dalam pengaturan sistem hidrologi (Bruijnzeel, 2004; Pan et al., 2011). Peran tersebut berkaitan dengan kerapatan vegetasi sebagai salah satu indikator dalam menentukan kualitas kesehatan hutan (De Frenne et al., 2021). Namun, deforestasi dan bencana alam menyebabkan luas hutan semakin menyusut, sehingga diperlukan sistem pemantauan yang akurat, berkelanjutan, serta efisien untuk mendukung kerja manusia di lapangan.

Integrasi kamera pada *Unmanned Aerial Vehicle* (UAV) menjadi solusi efektif karena mobilitas tinggi dalam menjangkau area yang sulit diakses. Dalam bidang pencitraan komputer, deteksi objek memungkinkan digunakan untuk mengidentifikasi keberadaan objek dalam suatu bingkai gambar secara simultan, serta mampu memberikan informasi berharga untuk pemahaman semantik dari gambar atau video (Jia et al., 2014; Krizhevsky et al., 2017). Untuk memenuhi kebutuhan pemrosesan secara *real-time*, penelitian menggunakan basis arsitektur YOLO (You Only Look Once), yaitu suatu algoritma yang digunakan untuk memprediksi keberadaan dan posisi dari suatu objek dalam suatu bingkai gambar (Redmon et al., 2016). YOLOv8 dipilih karena menunjukkan performa serta stabilitas yang lebih baik jika dibandingkan dengan versi YOLOv11 (Sengun et al., 2025). Penelitian menggunakan berbagai variasi skala model yang memberikan kompromi kecepatan dan akurasi (Yaseen, 2024).

Keberhasilan implementasi model pada arsitektur YOLO sangat ditentukan oleh kualitas dan keberagaman gambar pelatihan. Oleh karena itu, penerapan teknik augmentasi gambar menjadi krusial untuk mencegah model mengalami *overfitting* dan generalisasi model arsitektur YOLO (Shorten & Khoshgoftaar, 2019). Efektivitas didapatkan melalui evaluasi kemampuan model arsitektur YOLO yang dimiliki melalui metrik performa model tersebut, diantaranya adalah *precision*, *recall*, *average precision* (AP), dan *mean average precision* (mAP) (Padilla et al., 2020).

II. METODE

2.1 Pelatihan Model pada Arsitektur YOLO

Data yang digunakan dalam penelitian ini menggunakan dataset publik Roboflow Universe (Drone Project, 2025). Total data untuk pelatihan yang didapatkan mencakup 891 gambar dengan beberapa kondisi pohon, yaitu pohon tanpa daun, pohon rimbun tampak atas, serta pohon rimbun tampak samping. Contoh citra yang digunakan dalam penelitian ini dapat dilihat pada Gambar 1.



Gambar 1 Contoh citra dataset pelatihan yang diambil dari Roboflow.

Pada proses pelabelan, setiap objek pohon gambar pada dataset diberi anotasi *bounding box* secara manual untuk menunjukkan posisi pohon sebenarnya pada gambar tersebut. Anotasi diterapkan pada seluruh gambar yang digunakan untuk penelitian, termasuk gambar pelatihan (*train*), pengujian (*test*), serta validasi (*valid*). Setelah proses anotasi selesai, dilakukan *pre-processing* berupa augmentasi pada dataset untuk meningkatkan variansi dataset melalui parameter augmentasi seperti pada Tabel 1.

Tabel 1 Pengaturan augmentasi gambar latihan.

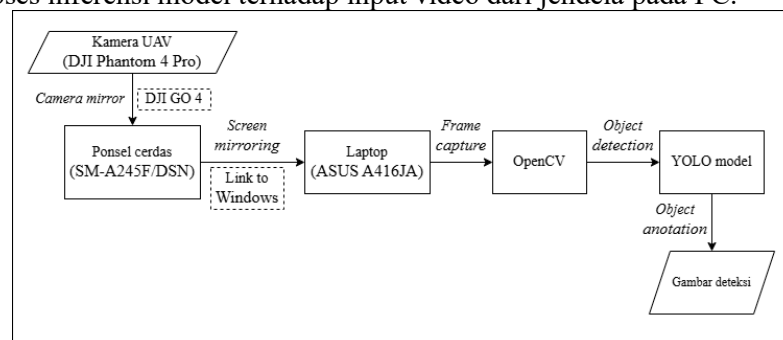
Jenis augmentasi	Nilai
<i>Rotation</i> (Rotasi)	-21° dan $+21^{\circ}$
<i>Shear</i> (Geseran)	$\pm 15^{\circ}$ Horizontal, $\pm 15^{\circ}$ vertikal
<i>Crop</i> (Potong)	0% minimum <i>Zoom</i> , 20% Maksimal <i>Zoom</i>
<i>Brightness</i> (Kecerahan)	-25% dan $+25\%$
<i>Exposure</i> (Eksposur)	-10% dan $+10\%$
<i>Blur</i> (Blur)	Hingga 3.5 px
<i>Outputs per training example</i> (Keluaran gambar untuk setiap contoh)	3

Setelah tahap augmentasi, total *dataset* untuk proses pelatihan berjumlah 2273 gambar yang terdiri atas 2073 gambar pelatihan, 67 gambar pengujian, dan 133 gambar validasi. Pembagian gambar menitikberatkan pada data pelatihan sehingga model dapat menangkap sebanyak mungkin variasi fitur objek (Ng, 2017). Proses pelatihan dilakukan menggunakan Google Colab karena kesederhanaan yang ditawarkan, serta akses konfigurasi GPU yang dapat diakses dalam waktu terbatas (Carneiro et al., 2018). Hasil pelatihan berupa bobot (*weight*) YOLO yang telah teroptimasi untuk objek pohon. Implementasi pada PC menggunakan bobot terbaik dari hasil pelatihan.

Proses pelatihan model dilakukan terhadap beberapa variasi model, yaitu YOLOv8n, YOLOv8m, serta YOLOv8l. Pemilihan berbagai variasi model YOLOv8 dilakukan untuk menentukan arsitektur model dengan kemampuan paling baik dan efisien untuk implementasi pada kasus yang dihadapi. Untuk masing-masing model diterapkan sejumlah 100 *epoch* (satu siklus pelatihan menggunakan seluruh data) dikarenakan jumlah data yang relatif kecil. Selain itu, target objek deteksi merupakan pohon yang memiliki karakteristik relatif seragam dalam berbagai sisi.

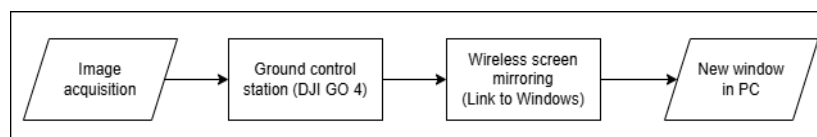
2.2 Implementasi Model

Proses implementasi model dalam penelitian ini mengikuti diagram alir sistem yang direpresentasikan pada Gambar 2. Implementasi model dilakukan melalui IDE Visual Studio Code dengan mengintegrasikan pustaka OpenCV untuk mendukung pemrosesan gambar secara *real-time*. Algoritma eksekusi model menggunakan bahasa pemrograman Python yang dirancang untuk memfasilitasi proses inferensi model terhadap input video dari jendela pada PC.



Gambar 2 Alur kerja sistem.

Sistem transmisi data dimulai dari pengambilan citra oleh kamera UAV DJI Phantom 4 Pro kemudian ditransmisikan melalui aplikasi DJI GO 4 sebagai *ground control station*. Tampilan layar perangkat seluler diproyeksikan secara *real-time* melalui protokol *wireless screen mirroring* melalui perangkat lunak Link to Windows. Setelah tampilan kamera UAV terdeteksi sebagai jendela pada PC, algoritma YOLO melakukan inferensi terhadap setiap bingkai gambar untuk mendeteksi keberadaan pohon yang ditunjukkan dalam bentuk *bounding box* dan *confidence score*. Alur transmisi data dari UAV hingga muncul pada jendela PC ditunjukkan pada Gambar 3.



Gambar 3 Diagram alir transmisi data.

2.3 Evaluasi Performa Model YOLO

Evaluasi performa model menggunakan parameter *precision*, *recall*, *F1-score*, dan *mean average precision* (mAP) (Padilla et al., 2020). *Precision* dan *recall* digunakan untuk mengukur keakuratan serta sensitivitas deteksi pohon, sedangkan *F1-score* merepresentasikan nilai keduanya. Nilai mAP@50 dan mAP@50-95 digunakan pada tahap pelatihan untuk menilai akurasi *bounding box*, sedangkan pengujian lapangan difokuskan pada nilai *precision*, *recall*, serta *F1-score*.

III. HASIL DAN DISKUSI

3.1 Hasil Pelatihan Model

Berdasarkan model yang telah dilatih, kemudian ditampilkan metrik performa untuk masing-masing model pada Tabel 2. Pada hasil terlihat bahwa model YOLOv8n memiliki keunggulan pada metrik *precision* dan mAP@50.

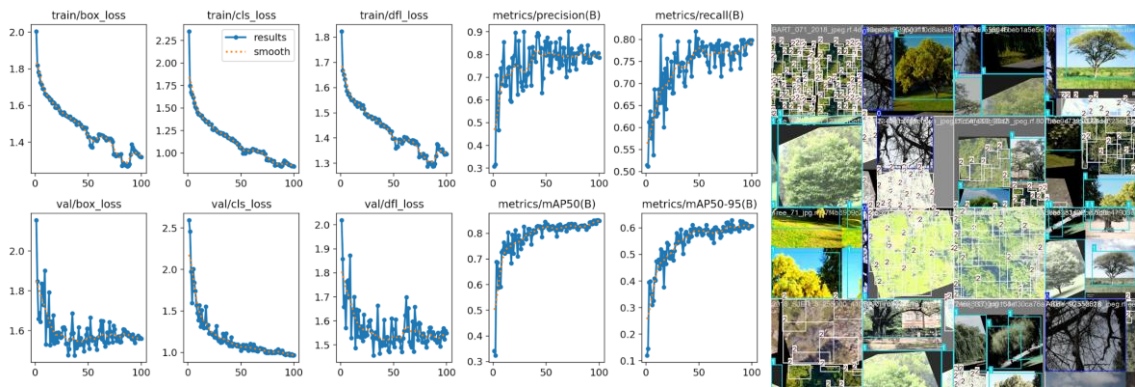
Tabel 2 Metrik performa untuk masing-masing model hasil pelatihan.

No.	Arsitektur	Ukuran gambar	Epoch	Precision	Recall	mAP@50	mAP@50:95
1	YOLOv8n	640×640	100	0,8278	0,75842	0,82349	0,62885
2	YOLOv8n-FT	640×640	25	0,72572	0,79282	0,82435	0,59047
3	YOLOv8m	640×640	100	0,82562	0,73758	0,80766	0,58809
4	YOLOv8l	640×640	100	0,79936	0,78486	0,81996	0,60807

Hasil pelatihan menunjukkan performa yang cukup baik meskipun menggunakan *dataset* terbatas dan jumlah *epoch* relatif kecil. Pada pelatihan *fine-tuning*, diperoleh peningkatan nilai *recall* meskipun menggunakan *epoch* lebih kecil. Ini menunjukkan bahwa pada proses *transfer learning*, penggunaan *epoch* sebesar 25 sudah cukup baik.

Dibandingkan penelitian lain dengan arsitektur model YOLOv5 (Evita et al., 2022), pelatihan yang dilakukan menunjukkan stabilitas yang lebih baik. Kemampuan kedua arsitektur model berbeda, YOLOv8 cenderung lebih stabil digunakan untuk deteksi objek (Soumia et al., 2023). Meskipun begitu, pada penelitian tersebut digunakan 4 kelas, *epoch* hingga 500, serta perbedaan jumlah *dataset* yang digunakan. Maka dari itu, nilai kuantitatif performa model tidak dapat dibandingkan secara langsung karena memiliki parameter pelatihan yang berbeda. Meskipun demikian, pelatihan yang dilakukan memiliki hasil yang mendekati penelitian lain yang menggunakan YOLOv8 serta jumlah *epoch* 100 (Kundu et al., 2025). Hal ini ditunjukkan dengan tren dan performa yang serupa. Ini berarti pelatihan yang dilakukan konsisten memiliki hasil pelatihan yang efektif. Berdasarkan studi komparatif terhadap kedua penelitian tersebut, didapatkan bahwa pelatihan yang dilakukan berjalan secara efektif hingga penggunaan *epoch* 100. Untuk dapat mengurangi nilai *loss* yang didapatkan, perlu ditambahkan jumlah *epoch*. Hal tersebut dilakukan sehingga model lebih andal dalam mengenali objek pohon dengan struktur rumit.

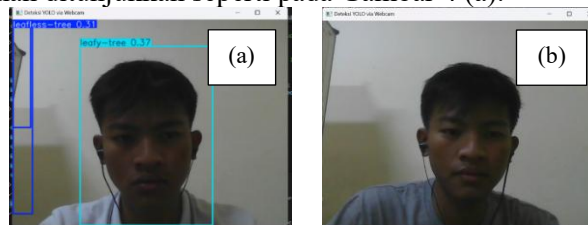
Gambar 4 menunjukkan kurva dan hasil pembacaan pelatihan YOLOv8n yang memiliki kecepatan inferensi pendeteksian. Pada kurva tersebut terlihat bahwa nilai *loss* menurun secara konsisten, menunjukkan kesalahan yang dilakukan model semakin menurun selama proses pelatihan. Sementara nilai mAP, *precision*, serta *recall* mengalami peningkatan seiring bertambahnya *epoch*, ini menunjukkan bahwa seiring berjalannya pelatihan model mengalami peningkatan akurasi serta sensitivitas dalam mendeteksi pohon. Secara keseluruhan, hal tersebut menunjukkan bahwa model mengalami peningkatan pemahaman terhadap objek pohon selama proses pelatihan berlangsung.



Gambar 4 Grafik dan gambar proses pelatihan YOLOv8n.

3.2 Evaluasi Menggunakan Metode *Fine-Tuning*

Pelatihan menggunakan metode *fine-tuning* dilakukan karena terdapat kesalahan pembacaan objek yang dilakukan oleh model YOLOv8n. Hal ini ditunjukkan dengan adanya pembacaan objek pohon pada gambar yang bukan merupakan pohon. Kesalahan ini dapat diklasifikasikan menjadi *FN* (*False Negative*). Kesalahan ditunjukkan seperti pada Gambar 4 (a).



Gambar 5 (a) Kesalahan pembacaan oleh model YOLOv8n dan (b) pembacaan hasil evaluasi.

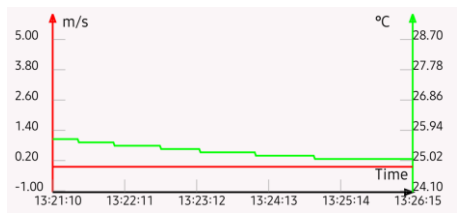
Untuk mengurangi kesalahan dalam deteksi objek pohon, dilakukan *fine-tuning* pada model yang telah dilatih (Li et al., 2021). Pada proses ini, bobot terbaik dari pelatihan awal digunakan sebagai titik awal pembelajaran. Dengan demikian, proses *fine-tuning* tidak membuat model belajar objek pohon dari awal, melainkan menyempurnakan parameter yang telah terbentuk (Parthasarathy et al., 2024).

Pada proses ini, dataset ditambahkan dengan citra tanpa pohon, seperti citra bangunan, citra manusia, serta citra yang sebelumnya salah dibaca sebagai pohon. Penambahan citra tanpa pohon bertujuan memberikan contoh negatif yang lebih beragam, sehingga model belajar untuk membedakan objek pohon lebih akurat. Melalui penerapan metode *fine-tuning* ini, tingkat kesalahan prediksi dapat ditekan secara signifikan, tanpa perlu melakukan pelatihan ulang dari awal (Parthasarathy et al., 2024).

Total jumlah citra yang digunakan dalam pelatihan berjumlah 1136 gambar. Setelah dilakukan augmentasi, didapatkan jumlah gambar sebanyak 2868 dengan pembagian 2598 gambar pelatihan, 180 gambar validasi, dan 90 gambar pengujian. Proses pelatihan menggunakan *epoch* sebanyak 25 karena model telah memiliki pengetahuan awal, serta untuk menghindari terjadinya *overfitting* pada model (Howard & Ruder, 2018). Berdasarkan hasil pelatihan metode *fine-tuning*, kesalahan pembacaan sebelumnya sudah tidak muncul, seperti ditunjukkan pada Gambar 5 (b).

3.3 Pengujian Lapangan

Pengujian lapangan dilaksanakan di kawasan kampus Institut Teknologi Bandung, Jl. Ganesha 10, Bandung, Kota Bandung, Jawa Barat, pada hari Kamis, 27 November 2025, pukul 13.00 WIB hingga 13.10 WIB. Kondisi lingkungan diamati melalui instrumen kecepatan angin serta suhu lingkungan, yaitu UNI-T UT363BT. Kondisi lingkungan terutama kecepatan angin dapat mempengaruhi stabilitas UAV ketika terbang, akibatnya dapat mempengaruhi stabilitas tangkapan gambar oleh kamera UAV (Wang et al., 2019). Selama pengambilan data, kondisi lingkungan cenderung berkabut dengan data angin ditampilkan pada Gambar 6.



Gambar 6 Kondisi angin ketika percobaan lapangan.

Pengujian lapangan menggunakan dua ketinggian berbeda, yaitu 39 m dan 70 m di atas tanah. Perbedaan ketinggian dilakukan untuk menentukan ketinggian berapa model lebih efektif untuk digunakan (Lanča et al., 2025). Berdasarkan hasil pengujian untuk masing-masing model, didapatkan hasil metrik performa seperti ditunjukkan pada Tabel 3.

Tabel 2 Metrik uji coba lapangan untuk setiap model pada dua ketinggian berbeda.

Arsitektur	Ketinggian (m)	Precision	Recall	F1-Score
YOLOv8n	39	0.96878	0.68889	0.805195
	70	1.0000	0.92592	0.961538
YOLOv8n-FT	39	1.0000	0.86364	0.926829
	70	1.0000	0.83529	0.910256
YOLOv8m	39	0.8000	0.82143	0.836364
	70	0.86667	0.72222	0.787879
YOLOv8l	39	1.0000	0.83871	0.912281
	70	1.0000	0.94737	0.972973

Berdasarkan Tabel 4, ketinggian paling efektif untuk menjalankan model adalah pada ketinggian 70 m, ditunjukkan pada nilai *precision* dan *recall* yang cenderung lebih tinggi. Sebagai contoh, pada model YOLOv8n, terjadi peningkatan nilai *F1-score* dari 0,805 (pada 39 m) menjadi 0,961 (pada 70 m). Hal yang sama terjadi pada penelitian lain yang menunjukkan pembacaan objek lebih efektif dilakukan pada ketinggian cukup tinggi dari tanah (Baselly-Villanueva et al., 2026). Efektivitas disebabkan sudut pandang yang lebih luas dibandingkan pengujian rendah (Torres-Sánchez et al., 2015). Selain itu, tingkat keburaman gambar mempengaruhi performa deteksi. Saat UAV cenderung bergerak pada ketinggian rendah terjadi, distorsi *frame* yang lebih besar jika dibandingkan dengan pengujian pada ketinggian tinggi (Sieberth et al., 2015). Maka dari itu, citra terlihat lebih buram, seperti ditunjukkan pada Gambar 6.



Gambar 7 Pembacaan gambar saat terbang pada ketinggian (a) 26 m dan (b) 27 m.

Model YOLOv8n hasil *fine-tuning* mengalami peningkatan performa pada pengujian dengan ketinggian rendah (39 m). Peningkatan nilai *F1-score* yang didapatkan cukup besar dibandingkan dengan model standar (0,805 menjadi 0,927). Hal tersebut menunjukkan bahwa metode *fine-tuning* membantu model mengenali objek dengan jarak lebih dekat karena memiliki resolusi citra yang lebih baik (Burmeister et al., 2025).

Pada pengujian dengan ketinggian 70 m, nilai *F1-score* pengujian menunjukkan bahwa model standar lebih unggul dibandingkan hasil *fine-tuning* (0,961 menjadi 0,901). Hal ini sesuai dengan hasil pelatihan yang menunjukkan bahwa model dengan pelatihan lanjutan melalui metode *fine-tuning* akan memberikan pembacaan yang lebih hati-hati dalam memutuskan objek yang dilihatnya sebagai pohon.



Gambar 8 Perbandingan pembacaan model *fine-tuning* (a) dan standar (b) pada ketinggian terbang 70 m.

IV. KESIMPULAN

Penelitian yang dilakukan telah berhasil melakukan evaluasi terhadap berbagai versi arsitektur YOLOv8 untuk pembacaan objek pohon melalui citra udara menggunakan UAV. Berdasarkan penelitian, model YOLOv8n diidentifikasi sebagai model paling optimal dalam penggunaan pemantauan secara *real-time* karena memiliki waktu inferensi paling kecil (5.8 ms). Penggunaan metode *fine-tuning* terbukti dapat meningkatkan nilai *F1-score* pada kondisi terbang rendah (0,805 menjadi 0,927). Di samping meningkatkan nilai *F1-score*, penggunaan metode *fine-tuning* secara signifikan berhasil meminimalkan kesalahan pembacaan yang dilakukan oleh arsitektur YOLO. Hasil pengujian lapangan menunjukkan bahwa arsitektur YOLOv8n paling optimal digunakan dengan ketinggian terbang UAV sebesar 70 m, ditunjukkan dengan nilai *F1-score* mencapai 0,961. Berdasarkan hasil pengujian, model pelatihan menggunakan implementasi *fine-tuning* memberikan kemampuan yang baik dalam implementasi pengukuran kerapatan pohon. Model terpilih dapat digunakan sebagai dasar pengembangan sistem pemantauan kerapatan vegetasi hutan, khususnya pada tahap identifikasi dan perhitungan objek pohon dalam skala ruang tertentu.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada semua pihak yang telah terlibat selama penelitian ini berlangsung. Ucapan terima kasih khusus disampaikan kepada Institut Teknologi Bandung yang telah memberikan dana riset sehingga penelitian ini dapat diselesaikan.

KONFLIK KEPENTINGAN

Penulis menyatakan tidak terdapat konflik kepentingan.

PERNYATAAN PENDANAAN

Penelitian ini tidak memperoleh pendanaan khusus.

PERNYATAAN PENGGUNAAN AI GENERATIF DAN TEKNOLOGI BERBASIS AI

GPT hanya digunakan untuk mengoreksi kesalahan tata bahasa dan tipografi. Penulis bertanggung jawab penuh atas isi dan integritas publikasi.

DAFTAR PUSTAKA

- Baselly-Villanueva, J. R., Fernández-Sandoval, A., Pinedo Freyre, S. F., Salazar-Hinostroza, E. J., Cárdenas-Rengifo, G. P., Puerta, R., Huanca Diaz, J. R., Tuesta Cometivos, G. A., Vallejos-Torres, G., Casas, G. G., Álvarez-Álvarez, P., & Ismail, Z. H. (2026). UAV Flight Orientation and Height Influence on Tree Crown Segmentation in Agroforestry Systems. *Forests*, 17(1), 87. <https://doi.org/10.3390/f17010087>
- Bruijnzeel, L. A. (2004). Hydrological functions of tropical forests: Not seeing the soil for the trees? *Agriculture, Ecosystems and Environment*, 104(1), 185–228. <https://doi.org/10.1016/j.agee.2004.01.015>

- Burmeister, J. M., Zabbarov, J., Reder, S., Richter, R., Mund, J. P., & Döllner, J. (2025). Fine-Tuning DeepForest for Forest Tree Detection in High-Resolution UAV Imagery. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 48(4/W15-2025), 39–46. <https://doi.org/10.5194/isprs-archives-XLVIII-4-W15-2025-39-2025>
- Carneiro, T., Da Nobrega, R. V. M., Nepomuceno, T., Bian, G. Bin, De Albuquerque, V. H. C., & Filho, P. P. R. (2018). Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications. *IEEE Access*, 6, 61677–61685. <https://doi.org/10.1109/ACCESS.2018.2874767>
- De Frenne, P., Lenoir, J., Luoto, M., Scheffers, B. R., Zellweger, F., Aalto, J., Ashcroft, M. B., Christiansen, D. M., Decocq, G., De Pauw, K., Govaert, S., Greiser, C., Gril, E., Hampe, A., Jucker, T., Klimes, D. H., Koelemeijer, I. A., Lembrechts, J. J., Marrec, R., ... Hylander, K. (2021). Forest microclimates and climate change: Importance, drivers and future research agenda. In *Global Change Biology* (Vol. 27, Issue 11, pp. 2279–2297). Blackwell Publishing Ltd. <https://doi.org/10.1111/gcb.15569>
- Drone Project. (2025). *tree_top_view Object Detection Model by Drone Project*. https://Universe.Roboflow.Com/Drone-Project-F1ea2/Tree_top_view-Fzybd
- Evita, M., Mustikawati, S. T., & Djamal, M. (2022). Design of Real-Time Object Detection in Mobile Robot for Volcano Monitoring Application. *Journal of Physics: Conference Series*, 2243(1). <https://doi.org/10.1088/1742-6596/2243/1/012038>
- Howard, J., & Ruder, S. (2018). *Universal Language Model Fine-tuning for Text Classification*. <http://nlp.fast.ai/ulmfit>
- Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., & Darrell, T. (2014). *Caffe: Convolutional Architecture for Fast Feature Embedding*. <http://arxiv.org/abs/1408.5093>
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). *ImageNet Classification with Deep Convolutional Neural Networks*. <http://code.google.com/p/cuda-convnet/>
- Kundu, S., Ninoria, S. Z., Chaturvedi, R. P., Mishra, A., Agrawal, A., Batra, R., Dubale, M., & Hashmi, A. (2025). Real-time deforestation anomaly detection using YOLO and LangChain agents for sustainable environmental monitoring. *Scientific Reports*, 15(1). <https://doi.org/10.1038/s41598-025-23617-4>
- Lanča, L., Mališa, M., Jakac, K., & Ivić, S. (2025). Optimal Flight Speed and Height Parameters for Computer Vision Detection in UAV Search. *Drones*, 9(9). <https://doi.org/10.3390/drones9090595>
- Li, Y., Zhang, H., & Zhang, Y. (2021). *Rethinking Training from Scratch for Object Detection*. <http://arxiv.org/abs/2106.03112>
- Ng, A. (2017, August 25). *Structuring ML Projects*.
- Padilla, R., Netto, S. L., & Da Silva, E. A. B. (2020). *A Survey on Performance Metrics for Object-Detection Algorithms*.
- Pan, Y., Birdsey, R., Fang, J., Houghton, R., Kauppi, P., Kurz, W., Phillips, O., Shvidenko, A., Lewis, S., Canadell, J., Ciais, P., Jackson, R., Pacala, S., McGuire, A., Piao, S., Rautiainen, A., Sitch, S., & Hayes, D. (2011). A Large and Persistent Carbon Sink in the World's Forests. *Science*, 333(6045), 988–993.
- Parthasarathy, V. B., Zafar, A., Khan, A., & Shahid, A. (2024). *The Ultimate Guide to Fine-Tuning LLMs from Basics to Breakthroughs: An Exhaustive Review of Technologies, Research, Best Practices, Applied Research Challenges and Opportunities*. <http://arxiv.org/abs/2408.13296>
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). *You Only Look Once: Unified, Real-Time Object Detection*. <http://arxiv.org/abs/1506.02640>
- Sengun, E., Aksoy, S., Sertel, E., & Fransson, J. E. S. (2025). Comparative Analysis of YOLOv8 and YOLOv11 on Tree Detection Using UAV RGB and Laser Scanning Data. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 10(2/W2-2025), 173–179. <https://doi.org/10.5194/isprs-annals-X-2-W2-2025-173-2025>
- Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1). <https://doi.org/10.1186/s40537-019-0197-0>

- Sieberth, T., Wackrow, R., & Chandler, J. H. (2015). UAV image blur-its influence and ways to correct it. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 40(1W4), 33–39. <https://doi.org/10.5194/isprsarchives-XL-1-W4-33-2015>
- Soumia, S. A., Asma, B., & Khaoula, N. (2023). *Comparative Evaluation of YOLOv5 and YOLOv8 Across Diverse Datasets*.
- Torres-Sánchez, J., López-Granados, F., & Peña, J. M. (2015). An automatic object-based method for optimal thresholding in UAV images: Application for vegetation detection in herbaceous crops. *Computers and Electronics in Agriculture*, 114, 43–52. <https://doi.org/10.1016/j.compag.2015.03.019>
- Wang, B. H., Wang, D. B., Ali, Z. A., Ting Ting, B., & Wang, H. (2019). An overview of various kinds of wind effects on unmanned aerial vehicle. *Measurement and Control (United Kingdom)*, 52(7–8), 731–739. <https://doi.org/10.1177/0020294019847688>
- Yaseen, M. (2024). *What is YOLOv8: An In-Depth Exploration of the Internal Features of the Next-Generation Object Detector*. <http://arxiv.org/abs/2408.15857>