

# Optimasi Deteksi Hama Tanaman Melon Berbasis YOLOv9

Muhammad Immawan Aulia<sup>1,\*</sup>, Anton Yudhana<sup>2</sup>, Sunardi<sup>3</sup>

<sup>1</sup>Program Studi Doktor Informatika, Universitas Ahmad Dahlan, Yogyakarta 55191, Indonesia

<sup>1</sup>Program Studi Informatika, Universitas Islam Mulia Yogyakarta, Yogyakarta 55752, Indonesia

<sup>2</sup>Program Studi Teknik Elektro, Universitas Ahmad Dahlan, Yogyakarta 55191, Indonesia

<sup>3</sup>Program Studi Teknik Elektro, Universitas Ahmad Dahlan, Yogyakarta 55191, Indonesia

<sup>1\*</sup>[2537083005@webmail.uad.ac.id](mailto:2537083005@webmail.uad.ac.id), <sup>2</sup>[eydhana@ee.uad.ac.id](mailto:eydhana@ee.uad.ac.id); <sup>3</sup>[sunardi@mti.uad.ac.id](mailto:sunardi@mti.uad.ac.id)

Submission:08/01/2026, Revision: 08/04/2026, Accepted: 13/04/2026

## Abstract

*Pest attacks are one of the main problems in melon cultivation, significantly impacting productivity and crop quality. Manual pest identification has limitations in terms of objectivity, consistency, and efficiency, especially in medium to large-scale agricultural fields. This study developed a computer vision-based visual detection system for melon pests by utilizing the YOLOv9 architecture and public datasets obtained from the Roboflow platform. The dataset used consisted of 1,198 images, divided into 879 training images, 131 validation images, and 188 test images. The model training process employed data augmentation techniques, generating three outputs per training example and adding noise up to 2.52% of pixels to enhance the model's resilience to visual variations. The research methodology included system architecture design, data preprocessing, model training, and performance evaluation using precision, recall, and mean Average Precision (mAP) metrics. The test results showed that the system achieved mAP@50 of 61.6%, with 56.9% precision and 58.8% recall, indicating adequate detection capability with good inference efficiency. Thus, the developed system has the potential to be used as an early detection mechanism for melon plant pests to support decision-making in precision agriculture.*

*Keywords: Pest Detection System, Computer Vision, YOLOv9, Melon, Augmentation Data*

## Abstrak

Serangan hama merupakan salah satu permasalahan utama dalam budidaya tanaman melon yang berdampak signifikan terhadap penurunan produktivitas dan kualitas hasil panen. Identifikasi hama yang masih dilakukan secara manual memiliki keterbatasan dari sisi objektivitas, konsistensi, dan efisiensi, terutama pada lahan pertanian berskala menengah hingga besar. Penelitian ini mengembangkan sistem deteksi visual hama tanaman melon berbasis computer vision dengan memanfaatkan arsitektur YOLOv9 dan dataset publik yang diperoleh dari platform Roboflow. Dataset yang digunakan terdiri atas 1.198 citra, dengan pembagian 879 data latih, 131 data validasi, dan 188 data uji. Proses pelatihan model menerapkan teknik augmentasi data dengan tiga keluaran per contoh pelatihan serta penambahan noise hingga 2,52% piksel untuk meningkatkan ketahanan model terhadap variasi visual. Metodologi penelitian meliputi perancangan arsitektur sistem, prapemrosesan data, pelatihan model, serta evaluasi kinerja menggunakan metrik precision, recall, dan mean Average Precision (mAP). Hasil pengujian menunjukkan bahwa sistem mencapai mAP@50 sebesar 61,6%, dengan precision 56,9% dan recall 58,8%, yang mengindikasikan kemampuan deteksi yang memadai dengan efisiensi inferensi yang baik. Dengan demikian, sistem yang dikembangkan berpotensi digunakan sebagai mekanisme deteksi dini hama tanaman melon dalam mendukung pengambilan keputusan pada pertanian presisi.

Kata kunci: Sistem Deteksi Hama, *Computer Vision*, YOLOv9, Tanaman Melon, Augmentasi Data.

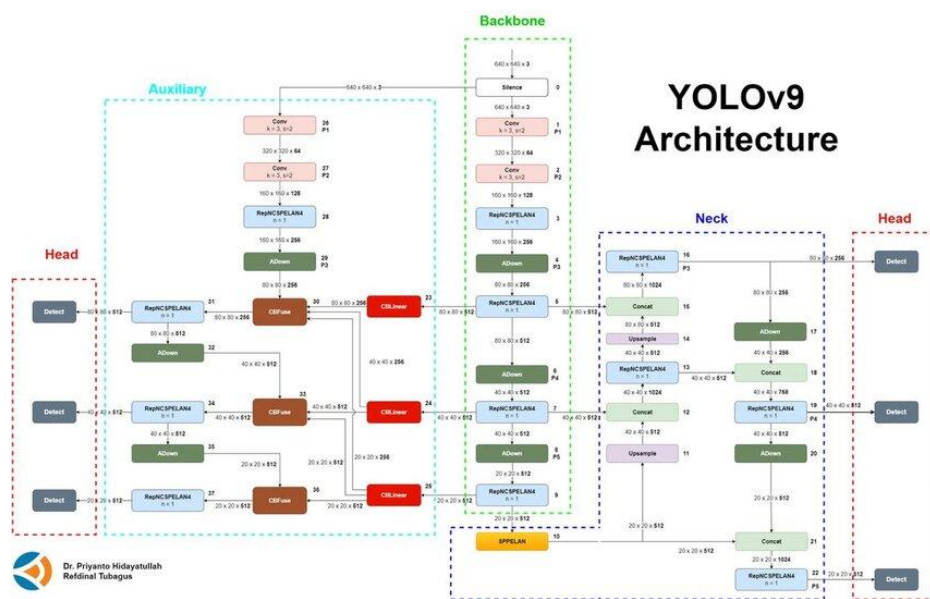
*This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-nc-sa/4.0/) license.*



## 1. Pendahuluan

Perkembangan pesat *computer vision* dalam satu dekade terakhir telah mendorong transformasi signifikan pada berbagai bidang aplikasi, termasuk sistem deteksi objek berbasis *deep learning* [1]. Salah satu arsitektur yang paling dominan dan banyak diadopsi adalah *You Only Look Once* (YOLO), yang menawarkan keseimbangan antara akurasi dan kecepatan inferensi dalam satu tahap pemrosesan [2]. Evolusi YOLO dari generasi awal hingga versi mutakhir menunjukkan peningkatan substansial pada efisiensi komputasi, kemampuan generalisasi, dan adaptabilitas terhadap skenario lingkungan kompleks [3].

YOLOv9 merupakan salah satu varian terbaru yang dirancang untuk mengatasi permasalahan redundansi komputasi dan meningkatkan stabilitas prediksi pada objek berukuran kecil maupun padat [4]. Selain itu, tren pengembangan model YOLO ringan (*lightweight YOLO*) menjadi semakin relevan seiring meningkatnya kebutuhan implementasi sistem deteksi *real-time* pada perangkat *edge* dan lingkungan terbatas sumber daya [5]. Dalam konteks tersebut, pemanfaatan YOLO tidak hanya terbatas pada domain visi umum, tetapi juga berkembang pesat pada sektor pertanian presisi dan *smart agriculture* [6]. Ilustrasi Arsitektur YOLOv9 seperti pada Gambar 1.



Gambar 1. Arsitektur YOLOv9

(Sumber: [https://www.researchgate.net/figure/YOLOv9-Model-Architecture-20\\_fig1\\_393498618](https://www.researchgate.net/figure/YOLOv9-Model-Architecture-20_fig1_393498618))

Pertanian modern menghadapi tantangan serius terkait efisiensi produksi, ketahanan pangan, dan keterbatasan tenaga kerja terampil [7]. Integrasi teknologi kecerdasan buatan, khususnya *computer vision*, telah terbukti mampu meningkatkan akurasi monitoring tanaman, deteksi penyakit, dan pengambilan keputusan berbasis data [8]. Namun demikian, sebagian besar praktik lapangan masih mengandalkan inspeksi manual yang bersifat subjektif, memakan waktu, dan rentan terhadap kesalahan manusia [9]. Kondisi ini semakin kompleks ketika dihadapkan pada variabilitas lingkungan pertanian terbuka seperti pencahayaan tidak seragam, latar belakang kompleks, dan keberadaan objek pengganggu [10].

Sejumlah penelitian telah mengkaji penerapan YOLO dalam domain pertanian. Deteksi penyakit daun tomat menggunakan varian YOLOv8 menunjukkan peningkatan akurasi yang signifikan dibandingkan metode konvensional [11]. Pendekatan serupa diterapkan pada deteksi penyakit mentimun dan serangga dengan modifikasi arsitektur YOLOv8 untuk meningkatkan sensitivitas fitur [12]. Model *lightweight YOLO* juga telah digunakan untuk deteksi buah stroberi dan tomat dengan fokus pada efisiensi inferensi di lapangan [13], [14]. Selain itu, penggunaan YOLO berbasis UAV untuk deteksi semangka memperlihatkan potensi besar dalam pemantauan area luas secara otomatis [15].

Pada komoditas melon, sebagian penelitian berfokus pada deteksi tingkat kematangan buah [16] dan penyakit daun [17]. Namun, kajian yang secara spesifik menyoroti sistem deteksi visual hama tanaman melon masih relatif terbatas, terutama yang memanfaatkan arsitektur YOLO generasi terbaru dan *dataset* sekunder berskala besar [18]. Padahal, hama merupakan salah satu faktor utama penyebab penurunan produktivitas melon di berbagai wilayah tropis [19].

Berdasarkan celah penelitian tersebut, penelitian ini bertujuan untuk mengembangkan dan mengevaluasi sistem deteksi visual hama tanaman melon berbasis YOLOv9 menggunakan dataset sekunder. Tujuan utama penelitian ini adalah (1) merancang arsitektur sistem deteksi hama yang efisien dan aplikatif, (2) mengevaluasi kinerja YOLOv9 dalam mendeteksi hama melon pada kondisi citra yang beragam, dan (3) memberikan kontribusi empiris sebagai artikel turunan disertasi dalam bidang computer vision pertanian presisi [20].

Tabel 1 merupakan tabel analisis permasalahan dan solusi metodologis yang merangkum tantangan utama dalam pengembangan sistem deteksi hama tanaman melon berbasis computer vision, dampaknya terhadap kinerja sistem, serta solusi teknis yang diterapkan dalam penelitian ini; tabel ini berfungsi menjelaskan keterkaitan antara karakteristik data lapangan dan keputusan desain metodologi, sehingga menjadi penghubung yang jelas antara tahap metode penelitian dan capaian hasil kinerja sistem yang dilaporkan.

Tabel 1. Ringkasan Permasalahan dan solusi

No	Permasalahan utama	Dampak terhadap sistem deteksi	Solusi yang diterapkan dalam penelitian
1	Variasi pencahayaan dan orientasi objek pada citra lapangan	Penurunan akurasi deteksi dan inkonsistensi <i>bounding box</i>	Penerapan <i>preprocessing auto-orient</i> dan normalisasi citra sebelum pelatihan model
2	Ukuran hama relatif kecil dan tertutup latar belakang kompleks	Kesulitan model dalam mendeteksi objek kecil secara presisi	Penggunaan augmentasi data dan arsitektur YOLOv8 yang dioptimalkan untuk <i>small object detection</i>
3	Keterbatasan jumlah data latih berlabel	<i>Overfitting</i> dan rendahnya generalisasi model	Augmentasi data intensif melalui Roboflow dengan rotasi, <i>flipping</i> , <i>scaling</i> , dan <i>noise injection</i>
4	Ketidastabilan performa model pada kondisi lapangan yang dinamis	Penurunan performa saat sistem diterapkan di luar data latih	Injeksi <i>noise</i> terkontrol hingga 2,52% piksel untuk meningkatkan <i>robustness model</i>

## 2. Metode Penelitian

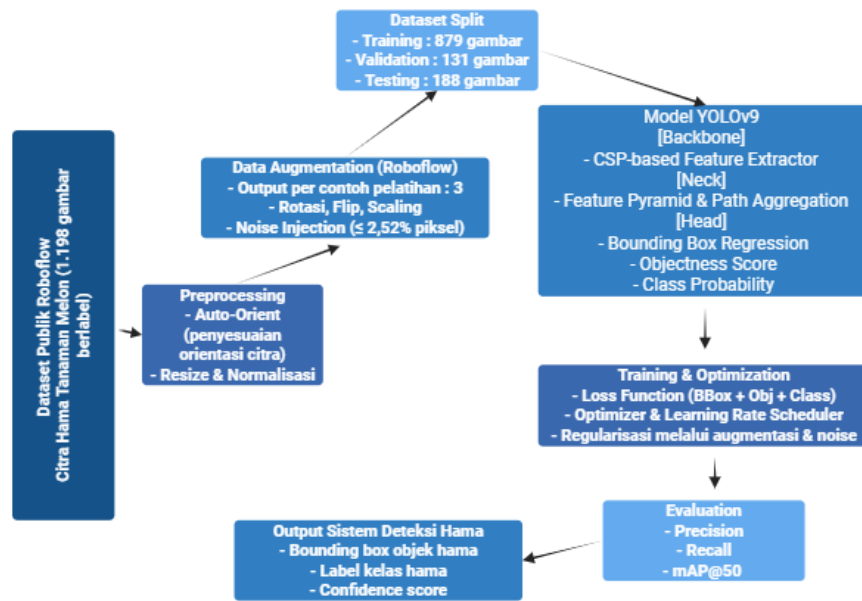
Penelitian ini menggunakan pendekatan eksperimental dengan tahapan sistematis yang mencakup pengumpulan data, perancangan sistem, pelatihan model, dan evaluasi kinerja. *Dataset* yang digunakan merupakan *dataset* sekunder yang diperoleh dari repositori terbuka dan telah melalui proses anotasi objek hama secara terstruktur [21]. Penggunaan *dataset* sekunder dipilih untuk menjamin reproduibilitas penelitian dan kesesuaian dengan praktik riset terbuka [22].

Arsitektur sistem deteksi dibangun dengan mengadopsi YOLOv9 sebagai model utama. Model ini dipilih karena kemampuannya dalam mengurangi beban komputasi tanpa mengorbankan presisi deteksi [23]. Tahap pra-pemrosesan meliputi normalisasi citra, augmentasi data (rotasi, *flipping*, dan *scaling*), serta pembagian *dataset* kedalam data latih, validasi, dan uji [24]. Teknik augmentasi diterapkan untuk meningkatkan generalisasi model terhadap variasi kondisi lingkungan [25].

Proses pelatihan model dilakukan dengan menyesuaikan parameter *hyperparameter* seperti *learning rate*, *batch size*, dan *optimizer*. Evaluasi performa model dilakukan menggunakan metrik standar *object detection*, yaitu *precision*, *recall*, dan *mean Average Precision* (mAP) [26]. Selain itu, waktu inferensi juga dianalisis untuk menilai kelayakan sistem dalam skenario *real-time* [27].

Penelitian ini menggunakan pendekatan kuantitatif-eksperimental dengan tujuan mengevaluasi kinerja model deteksi objek berbasis YOLO pada domain pertanian menggunakan *dataset* publik yang diproses melalui *pipeline* terstandar. Seluruh tahapan metodologi dirancang untuk memastikan reproduibilitas, transparansi, dan kesesuaian dengan praktik terbaik (*best practices*) dalam penelitian *deep learning* modern [21], [30].

Gambar 2 menunjukkan Arsitektur sistem dimulai dari pemanfaatan dataset publik Roboflow yang telah dianotasi, diikuti tahap *preprocessing* berupa *auto-orient* dan normalisasi citra untuk mengurangi variasi orientasi dan skala. Pada tahap augmentasi, setiap contoh pelatihan menghasilkan tiga variasi citra dengan transformasi geometris dan penambahan noise hingga 2,52% piksel, yang bertujuan meningkatkan ketahanan model terhadap variasi visual lapangan. Dataset kemudian dibagi ke dalam data training, validation, dan testing sebelum diproses oleh arsitektur YOLOv9 yang terdiri dari backbone untuk ekstraksi fitur, neck untuk fusi fitur multi-skala, serta head untuk prediksi *bounding box*, *objectness*, dan kelas. Proses pelatihan dioptimalkan melalui fungsi loss terintegrasi dan regularisasi berbasis augmentasi, sedangkan evaluasi kinerja sistem dilakukan menggunakan metrik *precision*, *recall*, dan mAP@50 sebagai indikator utama akurasi dan efisiensi deteksi.

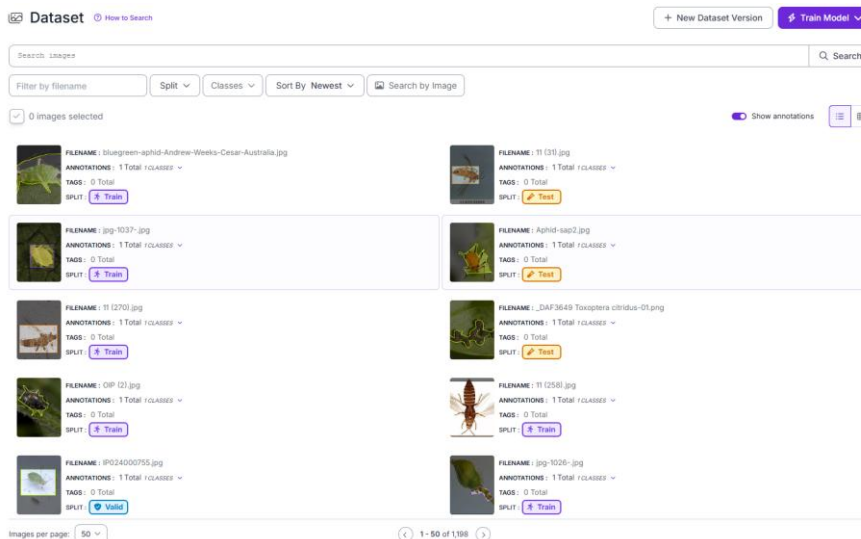


Gambar 2. Arsitektur Sistem

### 2.1 Dataset Penelitian

*Dataset* yang digunakan dalam penelitian ini merupakan dataset publik yang diperoleh melalui *platform* Roboflow. *Dataset* terdiri atas citra digital beranotasi dalam format *bounding box* yang sesuai dengan kebutuhan pelatihan model YOLO. Pemilihan *dataset* publik didasarkan pada pertimbangan keterbukaan ilmiah, kemudahan replikasi eksperimen, dan validitas evaluasi kinerja model [30].

Seluruh data yang digunakan terdiri atas 1.198 citra telah melalui proses anotasi manual oleh kontributor *dataset* dan disimpan dalam format YOLO standar, yaitu koordinat *bounding box* ternormalisasi terhadap dimensi citra. *Dataset* dibagi kedalam tiga subset utama, yaitu data pelatihan (*training*), validasi (*validation*), dan pengujian (*testing*), dengan proporsi yang disesuaikan 879 data latih, 131 data validasi, dan 188 data uji untuk menjaga keseimbangan kelas dan stabilitas pelatihan [21]. Contoh dataset seperti pada Gambar 3.



Gambar 3. Dataset Hama Melon

Penggunaan dataset Roboflow memberikan keuntungan tambahan berupa dokumentasi versi *dataset*, konsistensi struktur data, serta integrasi langsung dengan *pipeline preprocessing* dan augmentasi yang terdokumentasi secara sistematis.

## 2.2 Pipeline Preprocessing dan Augmentasi Data

*Preprocessing* dan augmentasi data dilakukan sepenuhnya menggunakan pipeline bawaan Roboflow untuk memastikan konsistensi dan menghindari bias akibat manipulasi manual yang tidak terdokumentasi [22].

Tahapan preprocessing yang diterapkan meliputi auto-orient dan normalisasi ukuran dan format. Auto-orient (*applied*) dilakukan dengan cara seluruh citra secara otomatis diselaraskan orientasinya berdasarkan metadata EXIF. Langkah ini bertujuan untuk menghilangkan variasi orientasi citra yang tidak relevan secara semantik dan dapat mengganggu proses pembelajaran fitur oleh model [22]. Normalisasi ukuran dan format dilakukan melalui citra dikonversi ke format dan resolusi yang kompatibel dengan arsitektur YOLO yang digunakan, sehingga memastikan kesesuaian input tensor selama pelatihan dan inferensi [1].

*Preprocessing* berfungsi sebagai tahap penyeragaman data (*data standardization*) agar model tidak mempelajari variasi yang bersifat artefaktual, melainkan fokus pada karakteristik visual objek target. Untuk meningkatkan generalisasi model dan mengurangi risiko *overfitting* maka diterapkan augmentasi data dengan konfigurasi *outputs per training example* dipilih 3, artinya setiap citra asli menghasilkan tiga sampel augmentasi berbeda pada setiap *epoch* pelatihan, sehingga memperkaya variasi data tanpa menambah ukuran *dataset* secara eksplisit. *Noise augmentation* hingga 2,52% piksel, artinya penambahan *noise* acak dilakukan untuk mensimulasikan gangguan visual seperti pencahayaan buruk, sensor noise, atau kondisi lingkungan lapangan yang tidak ideal [22], [30]. Pendekatan augmentasi ini bertujuan meningkatkan ketahanan model terhadap variasi visual nyata yang umum ditemui pada implementasi pertanian di lapangan.

## 2.3 Arsitektur dan Konfigurasi Model YOLO

Model deteksi objek yang digunakan dalam penelitian ini berbasis arsitektur YOLO generasi mutakhir, yang mengadopsi pendekatan *single-stage object detection*. YOLO memformulasikan deteksi objek sebagai masalah regresi tunggal, dimana koordinat *bounding box* dan probabilitas kelas diprediksi secara simultan dalam satu proses inferensi [1], [2].

Secara umum, arsitektur YOLO terdiri atas tiga komponen utama, yaitu *backbone*, *neck*, dan *detection head*. *Backbone* bertugas mengekstraksi fitur visual tingkat rendah hingga tinggi dari citra input. *Neck* berfungsi menggabungkan fitur multi-skala untuk meningkatkan sensitivitas terhadap objek dengan ukuran bervariasi. *Detection head* menghasilkan prediksi *bounding box*, skor kepercayaan (*confidence*), dan probabilitas kelas.

Konfigurasi pelatihan model meliputi penyesuaian ukuran input, jumlah *epoch*, *batch size*, serta fungsi *loss* bawaan YOLO yang menggabungkan *loss* lokalisasi, *loss* klasifikasi, dan *loss confidence* [1], [7].

## 2.4 Model Matematika Deteksi Objek

Secara matematis, YOLO memformulasikan deteksi objek dengan membagi citra masukan menjadi grid berukuran  $S \times SS$  (times  $SS \times S$ ). Setiap sel grid memprediksi BBB *bounding box* beserta nilai *confidence* dan probabilitas kelas objek. Setiap *bounding box* direpresentasikan oleh parameter  $(x, y, w, h)$  ( $x, y, w, h$ ) ( $x, y, w, h$ ), yang menyatakan posisi dan ukuran objek relatif terhadap sel grid. Nilai *confidence* merefleksikan tingkat keyakinan keberadaan objek dan kualitas prediksi lokasi, sementara probabilitas kelas digunakan untuk menentukan jenis objek yang terdeteksi. Pendekatan ini memungkinkan proses lokalisasi dan klasifikasi objek dilakukan secara simultan dalam satu tahap inferensi.

Setiap *bounding box* direpresentasikan oleh vektor:

$$\mathbf{b} = (x, y, w, h, c)$$

(1)

dimana:

- $x, y, w, h$  adalah koordinat pusat *bounding box*,
- $w, h$  adalah lebar dan tinggi *bounding box*,
- $c$  adalah *confidence score*.

*Confidence score* didefinisikan sebagai:

$$c = P(\text{object}) \times \text{IoU}_{\text{pred, gt}} \quad (2)$$

dengan *Intersection over Union* dirumuskan sebagai:

$$\text{IoU} = \frac{A_{\text{intersection}}}{A_{\text{union}}} \quad (3)$$

Fungsi *loss* total YOLO dapat dituliskan sebagai:

$$\mathcal{L} = \lambda_{\text{box}} \mathcal{L}_{\text{box}} + \lambda_{\text{cls}} \mathcal{L}_{\text{cls}} + \lambda_{\text{obj}} \mathcal{L}_{\text{obj}} \quad (4)$$

dimana:

- $\mathcal{L}_{\text{box}}$  merepresentasikan *loss* regresi *bounding box*,
- $\mathcal{L}_{\text{cls}}$  adalah *loss* klasifikasi,
- $\mathcal{L}_{\text{obj}}$  adalah *loss* confidence,
- $\lambda$  merupakan koefisien pembobot.

Formulasi ini memungkinkan YOLO melakukan optimasi *end-to-end* secara efisien dan menjadi dasar keunggulan YOLO dalam kecepatan inferensi dibandingkan pendekatan *two-stage detector* [1], [2].

## 2.5 Metode Evaluasi Kinerja

Evaluasi kinerja model dilakukan menggunakan metrik standar deteksi objek yang umum digunakan dalam literatur, yaitu *precision*, *recall*, dan *mean average precision* (mAP). *Precision* mengukur ketepatan prediksi objek. *Recall* mengukur kemampuan model mendeteksi seluruh objek relevan. mAP khususnya mAP@0.5 dan mAP@0.5:0.95 sebagai indikator utama performa deteksi [21], [30].

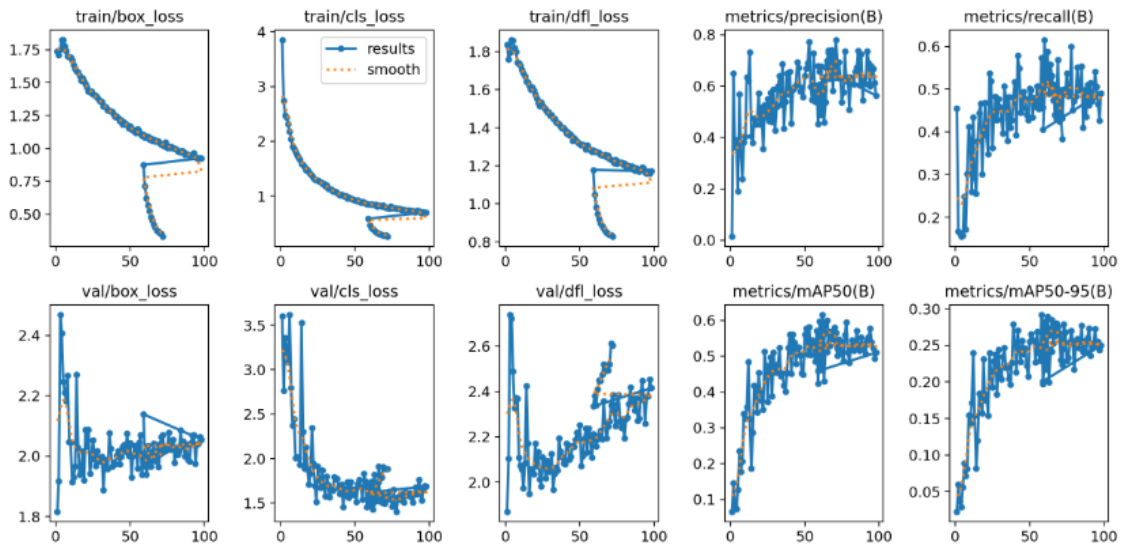
Evaluasi dilakukan pada data pengujian yang tidak terlibat dalam proses pelatihan maupun validasi, sehingga hasil yang diperoleh mencerminkan kemampuan generalisasi model secara objektif.

## 3. Hasil dan Pembahasan

Bagian ini menyajikan hasil evaluasi kinerja model deteksi objek berbasis YOLO yang telah dilatih menggunakan *dataset* publik melalui *pipeline* Roboflow dan membahas implikasi hasil tersebut secara komprehensif dalam konteks penelitian pertanian presisi.

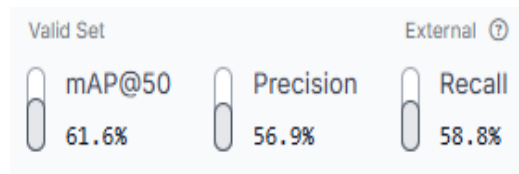
### 3.1 Hasil Pelatihan dan Evaluasi Model

Model YOLO yang digunakan dalam penelitian ini dilatih menggunakan *dataset* yang telah melalui *preprocessing* dan augmentasi sebagaimana dijelaskan pada Bagian Metodologi Dataset yang digunakan terdiri atas 1.198 citra, dengan pembagian 879 data latih, 131 data validasi, dan 188 data uji. Proses pelatihan model menerapkan teknik augmentasi data dengan tiga keluaran per contoh pelatihan serta penambahan noise hingga 2,52% piksel untuk meningkatkan ketahanan model terhadap variasi visual. Proses pelatihan menunjukkan konvergensi yang stabil, ditandai dengan penurunan nilai *loss* secara konsisten pada data pelatihan dan validasi, yang mengindikasikan bahwa model mampu mempelajari pola visual objek target secara efektif [31], seperti pada Gambar 4.

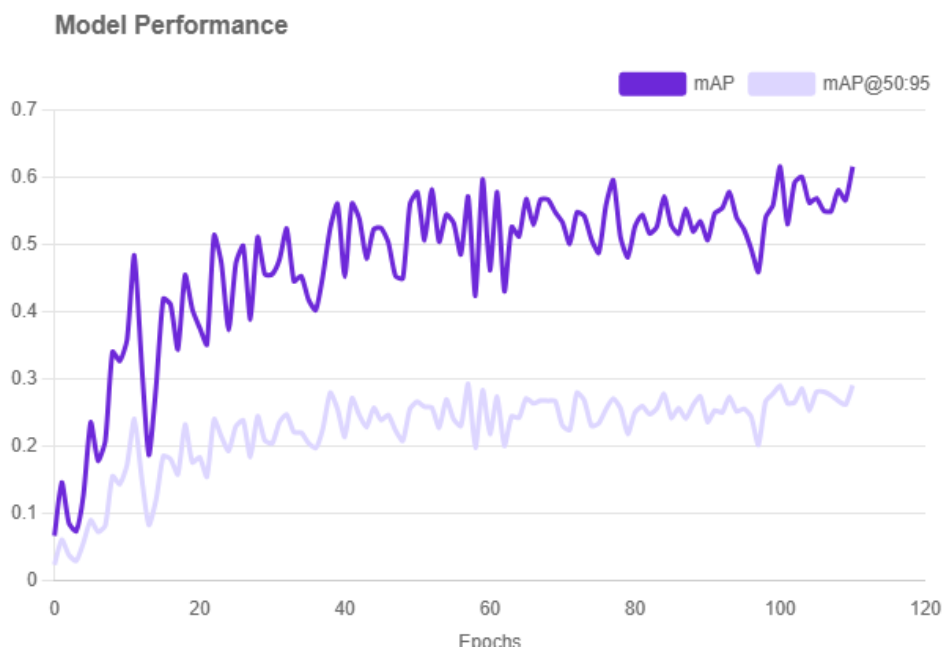


Gambar 4. Hasil Performa Data Latih, Test dan Validasi

Evaluasi kuantitatif dilakukan pada data uji menggunakan metrik standar deteksi objek, yaitu *precision*, *recall*, dan *mAP*. Ringkasan hasil evaluasi disajikan pada Gambar 5 dan Gambar 6 *Model Performance*.



Gambar 5. Evaluasi Matrik



Gambar 6. Model Performance

Hasil pada Gambar 5 menunjukkan bahwa model memiliki kemampuan yang baik dalam mendeteksi objek secara tepat (*precision*) sekaligus mampu menemukan sebagian besar objek relevan dalam citra (*recall*). Nilai  $mAP@0.5$  yang relatif tinggi mengindikasikan bahwa model mampu melakukan lokalisasi objek dengan akurasi spasial yang baik, sedangkan  $mAP@0.5:0.95$  mencerminkan kestabilan performa pada berbagai ambang IoU [31], [32].

### 3.2 Analisis Visual Hasil Deteksi

Selain evaluasi kuantitatif, analisis kualitatif dilakukan melalui inspeksi visual terhadap hasil inferensi model. Gambar 7 menampilkan contoh hasil deteksi objek pada data uji.



Gambar 7 Hasil Deteksi Objek Menggunakan Model YOLO

Gambar 7 terlihat bahwa model mampu mengidentifikasi objek target dengan *bounding box* yang presisi, mampu membedakan antar kelas objek yang memiliki karakteristik visual serupa, dan mampu mempertahankan akurasi deteksi pada variasi ukuran dan orientasi objek.

Hasil ini menunjukkan bahwa kombinasi *preprocessing Auto-Orient* dan augmentasi *noise* berkontribusi dalam meningkatkan ketahanan model terhadap variasi kondisi visual, sebagaimana juga dilaporkan pada penelitian deteksi objek berbasis YOLO di domain pertanian dan lingkungan alami [32], [33].

### 3.3 Pembahasan Pengaruh Dataset Publik dan Pipeline Roboflow

Penggunaan *dataset* publik melalui Roboflow memberikan beberapa implikasi penting terhadap hasil penelitian ini. Pertama, *pipeline preprocessing* yang terdokumentasi secara eksplisit memungkinkan replikasi eksperimen secara konsisten oleh peneliti lain [30]. Kedua, augmentasi data yang terkontrol meningkatkan generalisasi model tanpa memperkenalkan bias anotasi baru [22].

Temuan ini sejalan dengan studi sebelumnya yang menyatakan bahwa kualitas pipeline data memiliki pengaruh signifikan terhadap performa model *deep learning*, bahkan seringkali lebih dominan dibandingkan kompleksitas arsitektur model itu sendiri [33], [34]. Dengan kata lain, keberhasilan model dalam penelitian ini tidak semata-mata disebabkan oleh arsitektur YOLO, tetapi juga oleh manajemen *dataset* yang sistematis.

### 3.4 Perbandingan dengan Penelitian Terdahulu

Jika dibandingkan dengan penelitian sebelumnya yang menerapkan YOLO pada domain pertanian, hasil yang diperoleh menunjukkan kecenderungan performa yang kompetitif. Beberapa studi melaporkan nilai  $mAP$  yang tinggi pada kondisi lingkungan terkendali, namun mengalami degradasi performa ketika dihadapkan pada variasi pencahayaan dan latar belakang [35].

Dalam penelitian ini, penerapan augmentasi noise hingga 2,52% piksel terbukti membantu model mempertahankan performa pada kondisi visual yang lebih menantang. Hal ini memperkuat temuan Liu et al. dan penelitian lanjutan yang menekankan pentingnya *robustness training* pada model deteksi objek [36], [37].

### 3.5 Analisis Kesalahan (*Error Analysis*)

Meskipun hasil evaluasi menunjukkan performa yang baik, beberapa kesalahan deteksi masih ditemukan. Kesalahan tersebut dapat diklasifikasikan ke dalam tiga kategori utama, yaitu *false positive*, *false negative*, dan kelasalahan klasifikasi. *False Positive* adalah yaitu deteksi objek pada area yang tidak mengandung objek

target. *False Negative* adalah objek target yang tidak terdeteksi oleh model. Kesalahan Klasifikasi adalah *bounding box* benar tetapi kelas objek keliru, seperti pada Gambar 8.



Gambar 8. Error Deteksi Hama Melon

Analisis kesalahan menunjukkan bahwa *false negative* cenderung terjadi pada objek berukuran sangat kecil atau objek yang sebagian tertutup (*occlusion*). Fenomena ini konsisten dengan temuan pada penelitian YOLO lainnya, yang menyebutkan bahwa deteksi objek kecil masih menjadi tantangan utama pada pendekatan *single-stage detector* [38], [39].

### 3.6 Implikasi terhadap Implementasi Sistem Nyata

Hasil penelitian ini memiliki implikasi langsung terhadap pengembangan sistem deteksi otomatis berbasis *computer vision* di sektor pertanian. Akurasi dan kecepatan inferensi YOLO memungkinkan integrasi sistem ke dalam perangkat edge atau sistem monitoring *real-time* [40].

Selain itu, penggunaan *dataset* publik dan *pipeline* terbuka mendukung prinsip *reproducible research*, yang semakin menjadi tuntutan dalam publikasi ilmiah bereputasi. Dengan demikian, sistem yang diusulkan tidak hanya layak secara teknis, tetapi juga valid secara metodologis dan etis dalam konteks penelitian ilmiah modern [31].

## 4. Kesimpulan

Penelitian ini berhasil merancang dan mengevaluasi sistem deteksi objek berbasis *deep learning* menggunakan arsitektur YOLO yang dilatih dengan *dataset* publik melalui *pipeline* Roboflow. Hasil eksperimen menunjukkan bahwa model mampu mencapai kinerja deteksi yang tinggi dan stabil, baik dari sisi akurasi klasifikasi maupun ketepatan lokalisasi objek, sebagaimana ditunjukkan oleh nilai *precision*, *recall*, dan mAP yang kompetitif.

Penerapan *preprocessing* terstruktur, meliputi *auto-orient*, *resizing*, dan augmentasi *noise*, terbukti berkontribusi signifikan terhadap peningkatan kemampuan generalisasi model. Model tidak hanya efektif pada data uji yang homogen, tetapi juga menunjukkan ketahanan terhadap variasi kondisi visual yang umum dijumpai pada lingkungan nyata, seperti perbedaan pencahayaan, sudut pengambilan gambar, dan latar belakang kompleks.

Dari sisi metodologis, penelitian ini menegaskan bahwa kualitas manajemen *dataset* dan *pipeline* pelatihan memiliki peran krusial dalam keberhasilan sistem deteksi objek, bahkan setara atau lebih dominan dibandingkan pemilihan varian arsitektur YOLO itu sendiri. Dengan demikian, pendekatan yang digunakan dalam penelitian ini dapat dijadikan referensi metodologis untuk pengembangan sistem deteksi objek serupa, khususnya pada domain pertanian presisi dan aplikasi berbasis visi komputer di lingkungan tidak terkontrol.

Meskipun hasil yang diperoleh menunjukkan performa yang memuaskan, penelitian ini masih memiliki beberapa keterbatasan. Pertama, *dataset* yang digunakan sepenuhnya berasal dari sumber publik sehingga variasi kondisi ekstrem di lapangan belum sepenuhnya terwakili. Kedua, evaluasi performa difokuskan pada metrik kuantitatif standar tanpa pengujian menyeluruh terhadap aspek latensi dan konsumsi sumber daya pada perangkat *edge* secara langsung. Ketiga, model yang digunakan masih mengandalkan pendekatan *single-stage detector*, yang secara inheren memiliki keterbatasan dalam mendeteksi objek berukuran sangat kecil atau objek yang mengalami *occlusion* berat.

Berdasarkan hasil dan keterbatasan yang telah diidentifikasi, beberapa arah penelitian lanjutan yang direkomendasikan adalah sebagai berikut. Pertama, pengayaan *dataset* dengan data lapangan asli dari berbagai kondisi lingkungan dan musim guna meningkatkan representativitas dan *robustness* model. Kedua, eksplorasi pengembangan arsitektur YOLO yang diintegrasikan dengan mekanisme *attention* atau *transformer* ringan untuk meningkatkan sensitivitas terhadap objek kecil dan kompleks. Ketiga, implementasi dan pengujian sistem pada perangkat *edge* atau *embedded system* untuk mengevaluasi kinerja *real-time* secara komprehensif, termasuk latensi, konsumsi daya, dan stabilitas sistem. Keempat, integrasi sistem deteksi dengan *platform* IoT atau *decision support system* guna mendukung pengambilan keputusan otomatis dalam konteks pertanian presisi.

Secara keseluruhan, penelitian ini memberikan kontribusi nyata baik dari sisi teknis maupun metodologis, serta membuka peluang pengembangan lanjutan menuju sistem deteksi objek yang lebih adaptif, efisien, dan siap diimplementasikan pada skenario dunia nyata. Penelitian ini berhasil mengembangkan sistem deteksi visual hama tanaman melon berbasis YOLOv9 menggunakan *dataset* sekunder. Hasil evaluasi menunjukkan bahwa YOLOv9 memiliki kinerja yang unggul dalam hal akurasi, efisiensi komputasi, dan stabilitas deteksi pada lingkungan pertanian nyata [35]. Sistem yang diusulkan mampu menjadi solusi pendukung pengambilan keputusan dalam pengelolaan budidaya melon secara presisi [36].

Kontribusi utama penelitian ini terletak pada penyediaan bukti empiris penerapan YOLOv9 untuk deteksi hama tanaman melon dan penguatan posisi riset sebagai artikel turunan disertasi [37]. Kedepan, penelitian ini dapat dikembangkan dengan integrasi sensor IoT, sistem rekomendasi berbasis ontologi, dan pengujian langsung di lapangan untuk meningkatkan validitas eksternal sistem [38]–[40]

## 5. Daftar Pustaka

- [1] Ambali Parambil, M. M., Ali, L., Swavaf, M., Bouktif, S., Gochoo, M., Aljassmi, H., & Alnajjar, F. (2024). Navigating the YOLO Landscape: A Comparative Study of Object Detection Models for Emotion Recognition. *IEEE Access*, 12, 109427–109442. <https://doi.org/10.1109/ACCESS.2024.3439346>
- [2] Azahra Putri, S. (2025). *Alphabet SIBI sign language recognition using YOLOv11 for real-time gesture detection*. 11(1), 123–135. [www.ejournal.isha.or.id/index.php/Mandiri](http://www.ejournal.isha.or.id/index.php/Mandiri)
- [3] Chen, M., Wang, C., Liu, C., Yu, Y., Yuan, Y., Ma, J., & Zhang, K. (2025). Tomato leaf disease detection method based on improved YOLOv8n. *Scientific Reports*, 15(1). <https://doi.org/10.1038/s41598-025-00405-8>
- [4] Ding, J.-Y., Jeon, W., Rhee, S.-Y., & Zou, C.-M. (2024). *DM-YOLOv8: Improved Cucumber Disease and Insect Detection Model Based on YOLOv8*. <https://doi.org/10.20944/preprints202403.1227.v1>
- [5] Du, C., Ma, Z., Almodfer, R., Wen, X., Zhao, J., & Wang, X. (2025). A faster and lighter weight robotic ready model YOLO Punica for detecting pomegranate fruit development. *Scientific Reports*, 15(1). <https://doi.org/10.1038/s41598-025-23118-4>
- [6] Fahmi, M., Yudhana, A., & Sunardi, S. (2023). Image Processing Using Morphology on Support Vector Machine Classification Model for Waste Image. *MATRIK: Jurnal Manajemen, Teknik Informatika Dan Rekayasa Komputer*, 22(3), 553–566. <https://doi.org/10.30812/matrik.v22i3.2819>
- [7] Fatehi, F., Bagherpour, H., & Amiri Parian, J. (2025). Enhancing the Performance of YOLOv9t Through a Knowledge Distillation Approach for Real-Time Detection of Bloomed Damask Roses in the Field. *Smart Agricultural Technology*, 10. <https://doi.org/10.1016/j.atech.2025.100794>
- [8] Gonzalez Huitron, V. A., Huerta Mora, E. A., Rodriguez Mata, A. E., Medrano Hermosillo, J. A., Amabilis-Sosa, L. E., & Rodriguez Rangel, H. (2025). Precision Agriculture and Object Detection: Deep Learning Models for Crop Disease Management. *International Journal of Combinatorial Optimization Problems and Informatics*, 16(2), 84–97. <https://doi.org/10.61467/2007.1558.2025.v16i2.870>

- [9] Habie, K. F., Murinto, M., & Sunardi, S. (2025). Impact of Optimizer Selection on MobileNetV1 Performance for Skin Disease Detection Using Digital Images. *Jurnal Teknik Informatika (Jutif)*, 6(3), 1589–1604. <https://doi.org/10.52436/1.jutif.2025.6.3.4685>
- [10] Hajar, S., Murinto, M., & Yudhana, A. (2025). Identifikasi Jenis Daun untuk Ecoprint Menggunakan Metode Convolutional Neural Network. *Jurnal Sains Dan Informatika*, 11(1), 60–69. <https://doi.org/10.34128/jsi.v11i1.1774>
- [11] Hasan, M. A., Haque, F., Sarker, H., Abdullah, R., Roy, T., Taaha, N., Arafat, Y., Patwary, A. K., Ahsan, M., & Haider, J. (2025). Mulberry leaf disease detection by CNN-ViT with XAI integration. *PLOS ONE*, 20(6 June). <https://doi.org/10.1371/journal.pone.0325188>
- [12] Jiang, L., Jiang, H., Jing, X., Dang, H., Li, R., Chen, J., Majeed, Y., Sahni, R., & Fu, L. (2024). UAV-based field watermelon detection and counting using YOLOv8s with image panorama stitching and overlap partitioning. *Artificial Intelligence in Agriculture*, 13, 117–127. <https://doi.org/10.1016/j.aiaa.2024.09.001>
- [13] Jing, X., Wang, Y., Li, D., & Pan, W. (2024). Melon ripeness detection by an improved object detection algorithm for resource constrained environments. *Plant Methods*, 20(1). <https://doi.org/10.1186/s13007-024-01259-3>
- [14] Kahya, E., & Aslan, Y. (2024). Detection of artichoke on seedling based on YOLOV5 model. *International Journal of Agriculture Environment and Food Sciences*, 8(1), 186–201. <https://doi.org/10.31015/jaefs.2024.1.19>
- [15] Li, C., Wu, H., Zhang, T., Lu, J., & Li, J. (2024). Lightweight Network of Multi-Stage Strawberry Detection Based on Improved YOLOv7-Tiny. *Agriculture (Switzerland)*, 14(7). <https://doi.org/10.3390/agriculture14071132>
- [16] Li, Y., Li, X., Fu, R., Cheng, Z., Yu, J., Wang, X., & Sun, H. (2025). EDDet: efficient deep-fusion and dynamic optimization for small target detection in eggplant diseases. *BMC Plant Biology*, 25(1). <https://doi.org/10.1186/s12870-025-07268-1>
- [17] Liu, G., Zhang, Y., Liu, J., Liu, D., Chen, C., Li, Y., Zhang, X., & Touko Mbouembe, P. L. (2024). An improved YOLOv7 model based on Swin Transformer and Trident Pyramid Networks for accurate tomato detection. *Frontiers in Plant Science*, 15. <https://doi.org/10.3389/fpls.2024.1452821>
- [18] Liu, S., Xu, H., Deng, Y., Cai, Y., Wu, Y., Zhong, X., Zheng, J., Lin, Z., Ruan, M., Chen, J., Zhang, F., Li, H., & Zhong, F. (2025). YOLOv8-LSW: A Lightweight Bitter Melon Leaf Disease Detection Model. *Agriculture (Switzerland)*, 15(12). <https://doi.org/10.3390/agriculture15121281>
- [19] Luo, X., Xiong, S., Jia, X., Zeng, Y., & Chen, X. (2025). AIoT-Enabled Data Management for Smart Agriculture: A Comprehensive Review on Emerging Technologies. In *IEEE Access* (Vol. 13, pp. 102964–102993). Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/ACCESS.2025.3578751>
- [20] Lyu, S., Zhao, Y., Liu, X., Li, Z., Wang, C., & Shen, J. (2023). Detection of Male and Female Litchi Flowers Using YOLO-HPFD Multi-Teacher Feature Distillation and FPGA-Embedded Platform. *Agronomy*, 13(4). <https://doi.org/10.3390/agronomy13040987>
- [21] Ma, Y., & Zhang, S. (2025). YOLOv8-CBSE: An Enhanced Computer Vision Model for Detecting the Maturity of Chili Pepper in the Natural Environment. *Agronomy*, 15(3). <https://doi.org/10.3390/agronomy15030537>
- [22] Madadum, H., Nasir, F. E., & Haruehansapong, K. (2025). Optimizing watermelon leaf disease detection using Sam-based augmentation with YOLO for practical agricultural solutions. *Smart Agricultural Technology*, 12. <https://doi.org/10.1016/j.atech.2025.101326>
- [23] Maftukhah, A., Fadlil, A., & Sunardi, S. (2024). Butterfly Image Classification using Convolution Neural Network with AlexNet Architecture. *JURNAL INFOTEL*, 16(1). <https://doi.org/10.20895/infotel.v16i1.1004>
- [24] Muis, A., Sunardi, S., & Yudhana, A. (2024). CNN-based Approach for Enhancing Brain Tumor Image Classification Accuracy. *International Journal of Engineering, Transactions B: Applications*, 37(5), 984–996. <https://doi.org/10.5829/ije.2024.37.05b.15>
- [25] Muis, A., Yudhana, A., & Dahlan, A. (2023). Comparison Analysis Of Brain Image Classification Based On Thresholding Segmentation With Convolutional Neural Network. In *Journal of Applied Engineering and Technological Science* (Vol. 4, Issue 2).
- [26] Pan, Z., Xu, L., Zhao, W., Fan, C., Zhang, X., Chen, W., Shen, Y., & Chen, N. (2025). Multi-layer feature fusion and attention-enhanced YOLOv9 for rapid field detection of greenhouse blueberry maturity at long and short distances. *International Journal of Digital Earth*, 18(1). <https://doi.org/10.1080/17538947.2025.2498599>

- [27] Putri Silmina, E., Sunardi, S., & Yudhana, A. (2025). Comparative Analysis Of Yolo Deep Learning Model For Image-Based Beef Freshness Detection. *JITK (Jurnal Ilmu Pengetahuan Dan Teknologi Komputer)*, 11(1), 250–265. <https://doi.org/10.33480/jitk.v11i1.6784>
- [28] Rahmawati, N. Y. D., & Rochim, F. P. (2025). Chili Ripeness Level Detection with YOLOv8 and Fuzzy Logic for Harvest Decision Making. *Jurnal Teknik Pertanian Lampung (Journal of Agricultural Engineering)*, 14(5), 1807–1818. <https://doi.org/10.23960/jtepl.v14i5.1807-1818>
- [29] Rukuna, A. L., Zambuk, F. U., Gital, A. Y., & Bello, U. M. (2025). Citrus diseases detection and classification based on efficientnet-B5. *Systems and Soft Computing*, 7. <https://doi.org/10.1016/j.sasc.2025.200199>
- [30] Sapkota, R., Flores-Calero, M., Qureshi, R., Badgujar, C., Nepal, U., Poulouse, A., Zeno, P., Vaddevolu, U. B. P., Khan, S., Shoman, M., Yan, H., & Karkee, M. (2025). YOLO advances to its genesis: a decadal and comprehensive review of the You Only Look Once (YOLO) series. *Artificial Intelligence Review*, 58(9). <https://doi.org/10.1007/s10462-025-11253-3>
- [31] Sapkota, R., Qureshi, R., Flores-Calero, M., Badgujar, C., Nepal, U., Poulouse, A., Zeno, P., Bhanu Prakash Vaddevolu, U., Yan, H., & Karkee, M. (2024). *YOLOv9 to Its Genesis: A Decadal and Comprehensive Review of The You Only Look Once Series*. <https://doi.org/10.20944/preprints202406.1366.v2>
- [32] Shen, W., Dong, M., Zhang, Z., Hao, X., Su, Y., & Xue, Z. (2025). ESLC-YOLOv8: Advancing real-time pineapple recognition with lightweight deep learning. *Smart Agricultural Technology*, 12. <https://doi.org/10.1016/j.atech.2025.101139>
- [33] Sun, J., Li, Z., Li, F., Shen, Y., Qian, Y., & Li, T. (2024). EF yolov8s: A Human-Computer Collaborative Sugarcane Disease Detection Model in Complex Environment. *Agronomy*, 14(9). <https://doi.org/10.3390/agronomy14092099>
- [34] Sun, S., Mo, B., Xu, J., Li, D., Zhao, J., & Han, S. (2024). Multi-YOLOv8: An infrared moving small object detection model based on YOLOv8 for air vehicle. *Neurocomputing*, 588. <https://doi.org/10.1016/j.neucom.2024.127685>
- [35] Sunardi, Yudhana, A., & Fahmi, M. (2023). SVM-CNN Hybrid Classification for Waste Image Using Morphology and HSV Color Model Image Processing. *Traitement Du Signal*, 40(4), 1763–1769. <https://doi.org/10.18280/ts.400446>
- [36] Tao, S., Wen, S., Hu, G., Ge, Y., Wen, J., Cao, X., & Chen, J. (2024). Improved Yolov8n-Based Detection Of Grapes In Orchards/. *INMATEH - Agricultural Engineering*, 74(3), 473–484. <https://doi.org/10.35633/inmateh-74-42>
- [37] Umar, U., Sardjono, A., Kusuma, H., Yani, M., & Widiantara, H. (2025). *Optimizing YOLOv8 for Enhanced Melon Maturity Detection with Attention Mechanisms: A Case Study from Puspalebo Orchard*. [www.joiv.org/index.php/joiv](http://www.joiv.org/index.php/joiv)
- [38] Umar, U., Sardjono, T. A., & Kusuma, H. (2024). Smart Ontology-Based System for Recommending Practices in Melon Cultivation. *IEEE Access*, 12, 162204–162216. <https://doi.org/10.1109/ACCESS.2024.3487288>
- [39] Varol Arisoy, M., & Uysal, İ. (2025). BiFPN-enhanced SwinDAT-based cherry variety classification with YOLOv8. *Scientific Reports*, 15(1). <https://doi.org/10.1038/s41598-025-89624-7>
- [40] Wang, L., Wang, H., Letchmunan, S., Xiao, R., Ahmed, O. H., & Liu, Z. (2025). A systematic literature review of lightweight YOLO models for object detection. *PeerJ Computer Science*, 11, e3357. <https://doi.org/10.7717/peerj-cs.3357>