

Analisis Komparatif Kinerja Arsitektur SSD-VGG16, SSD-ResNet18, dan SSD-MobileNetV2 untuk Deteksi Penyakit Brownspot Padi

Muammar Khadafi*¹, Oddy Virgantara Putra²

¹²Teknik Informatika, ¹²Universitas Darussalam Gontor

¹²Ponorogo, ¹²Jawa Timur - Indonesia

Email: [1muammarkhadafi17@mhs.unida.gontor.ac.id](mailto:muammarkhadafi17@mhs.unida.gontor.ac.id),

[2oddy@unida.gontor.ac.id](mailto:oddy@unida.gontor.ac.id)

Abstract

Early detection of rice leaf diseases, particularly brown spot, is crucial to prevent yield loss, yet manual inspection is often subjective and inefficient. While deep learning has shown promise, existing studies often prioritize accuracy without adequately addressing the trade-off between model size and inference speed for varying deployment scenarios. This study aims to evaluate and compare three backbones—VGG16, ResNet18, and MobileNetV2—within the Single Shot Detector (SSD) framework to identify the optimal architecture for specific constraints. We utilized 1,000 annotated images with pre-augmented variations and trained the models using the SSD300 architecture on a standardized GPU environment. The results demonstrate a clear accuracy-efficiency trade-off: SSD-VGG16 achieved the highest accuracy (mAP 0.825) but was the slowest (11.53 FPS), whereas SSD-MobileNetV2 was the most efficient (29.42 FPS) and lightweight (43.73 MB) suitable for real-time edge devices, albeit with lower accuracy (mAP 0.707). SSD-ResNet18 offered a balanced compromise. These findings provide empirical guidelines for selecting object detection models based on the priority of either precision or real-time capability in agricultural applications.

Keywords: Object Detection, Deep Learning, Brownspot, SSD (Single Shot Detector), Accuracy-Efficiency Trade-Off, MobileNetV2, VGG16, ResNet18

Abstraksi

Deteksi dini penyakit daun padi, khususnya brown spot, sangat krusial untuk mencegah kerugian panen, namun inspeksi manual sering kali subjektif dan tidak efisien. Meskipun deep learning menjanjikan solusi, studi yang ada sering kali hanya berfokus pada akurasi tanpa membahas secara mendalam kompromi (trade-off) antara ukuran model dan kecepatan inferensi untuk berbagai skenario penerapan. Penelitian ini bertujuan untuk mengevaluasi dan membandingkan tiga backbone VGG16, ResNet18, dan MobileNetV2 dalam kerangka kerja Single Shot Detector (SSD) guna menentukan arsitektur optimal sesuai batasan sumber daya. Kami menggunakan 1.000 citra beranotasi yang telah diaugmentasi dan melatih model menggunakan arsitektur SSD300 pada lingkungan GPU standar. Hasil penelitian menunjukkan trade-off yang jelas: SSD-VGG16 mencapai akurasi tertinggi (mAP 0,825) namun paling lambat (11,53 FPS), sementara SSD-MobileNetV2 adalah yang paling efisien (29,42 FPS) dan ringan (43,73 MB) sehingga cocok untuk perangkat real-time, meskipun dengan akurasi lebih rendah (mAP 0,707). SSD-ResNet18 menawarkan keseimbangan di antara keduanya (Hasil). Temuan ini memberikan panduan empiris dalam pemilihan model deteksi objek

berdasarkan prioritas antara presisi tinggi atau kapabilitas real-time dalam aplikasi pertanian.

Kata Kunci: Deteksi Objek, Deep Learning, Brownspot, SSD (Single Shot Detector), Trade-Off Akurasi-Efisiensi, MobileNetV2, VGG16, ResNet18

1. PENDAHULUAN

Padi (*Oryza sativa*) merupakan komoditas strategis yang berperan penting dalam menjaga ketahanan pangan nasional. Salah satu tantangan utama dalam budidaya tanaman ini adalah penurunan produktivitas akibat penyakit daun, khususnya *brown spot* yang disebabkan oleh jamur *Bipolaris oryzae* [1]. Deteksi dini terhadap penyakit ini sangat penting untuk mencegah kerugian panen. Namun, metode konvensional seperti pemeriksaan manual sering terkendala subjektivitas dan keterlambatan diagnosis. Oleh karena itu, diperlukan penerapan teknologi *computer vision* berbasis *deep learning* yang mampu memberikan deteksi cepat, objektif, dan konsisten di lapangan [2].

Penelitian terkini menunjukkan bahwa pengembangan model berbasis arsitektur ringan (*lightweight*) semakin diupayakan agar diagnosis dapat dilakukan secara *real-time*, bahkan pada perangkat dengan keterbatasan sumber daya, tanpa mengorbankan akurasi secara signifikan [3]. *Single Shot Detector* (SSD) tetap menjadi pendekatan yang relevan dalam tugas deteksi objek karena kemampuannya mengintegrasikan proses lokalisasi dan klasifikasi dalam satu forward pass yang efisien. Tren riset 2024–2025 menunjukkan peningkatan fokus pada optimalisasi SSD melalui desain *backbone* dan integrasi *attention modules* guna meningkatkan akurasi, khususnya dalam mendeteksi objek kecil seperti bercak daun [4].

Dalam konteks tanaman padi, sejumlah studi menunjukkan bahwa model ringan berbasis *MobileNet* (V2/V3) memiliki potensi besar dalam klasifikasi dan deteksi penyakit daun, dengan akurasi kompetitif dan ukuran model kecil. Beberapa penelitian terbaru melaporkan peningkatan performa melalui modifikasi arsitektur *inverted residual* serta integrasi algoritma SSD dan YOLO yang disederhanakan, terbukti efektif untuk mendeteksi penyakit seperti *brown spot*, *blast*, dan *blight* [5].

Meskipun demikian, masih terdapat ketidakpastian mengenai arsitektur mana yang paling optimal ketika dihadapkan pada keterbatasan perangkat keras nyata. Pemilihan model sering kali didasarkan pada popularitas, bukan pada pertimbangan *trade-off* antara presisi deteksi dan efisiensi komputasi. Oleh karena itu, penelitian ini mengajukan pertanyaan utama: bagaimana karakteristik *backbone* berbeda (VGG16, ResNet18, dan MobileNetV2) mempengaruhi keseimbangan antara akurasi deteksi (*mAP/IOU*) dan efisiensi (*ukuran model/FPS*) dalam kerangka kerja SSD.

Penelitian ini bertujuan untuk melakukan evaluasi komparatif terhadap ketiga arsitektur tersebut. Kontribusi utama penelitian ini adalah: (1) menyajikan analisis empiris mengenai performa *backbone* berat (VGG16) versus ringan (MobileNetV2) untuk deteksi penyakit *brown spot* padi, dan (2) memberikan rekomendasi berbasis data mengenai pemilihan model yang sesuai dengan skenario implementasi, baik untuk analisis *offline*

yang menekankan presisi maupun aplikasi *mobile* yang menuntut kecepatan *real-time*.

2. TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Penelitian terdahulu menunjukkan bahwa pendekatan *deep learning*, khususnya dengan arsitektur *Convolutional Neural Network* (CNN), telah banyak digunakan dalam deteksi penyakit tanaman. Chatterjee dan Ghosh [6] melalui studi *PlantOHealth* membandingkan beberapa arsitektur CNN seperti VGG16, VGG19, ResNet, dan MobileNetV2 untuk klasifikasi penyakit tanaman menggunakan dataset PlantVillage. Hasil penelitian tersebut menunjukkan bahwa MobileNetV2 memberikan akurasi tertinggi (99,40%) dengan efisiensi komputasi terbaik, sehingga cocok untuk implementasi *real-time* pada perangkat dengan sumber daya terbatas. Namun, penelitian ini hanya berfokus pada klasifikasi citra tanpa mencakup deteksi area infeksi secara spasial, sehingga belum menjawab kebutuhan akan analisis *trade-off* antara akurasi dan efisiensi pada tugas deteksi objek.

Wang et al. [7] melakukan penelitian lanjutan dengan menerapkan *Single Shot Multibox Detector* (SSD) berbasis *transfer learning* untuk mendeteksi hama pada tanaman jeruk. Mereka membandingkan beberapa *backbone* CNN seperti VGG16, ResNet50, dan MobileNetV3, dan menemukan bahwa SSD-MobileNetV3 mampu mencapai *mean average precision* (mAP) 86,10% dengan latensi rendah 185 milidetik. Hasil ini membuktikan efektivitas arsitektur SSD dengan *backbone* ringan untuk deteksi cepat pada perangkat *mobile*. Meskipun demikian, penelitian tersebut belum melakukan evaluasi komprehensif terhadap *trade-off* akurasi dan efisiensi pada berbagai arsitektur dalam konteks penyakit tanaman.

Penelitian lain oleh Roseno et al. [8] membandingkan ResNet50, VGG16, dan MobileNetV3 untuk klasifikasi penyakit daun padi, termasuk *brown spot*. MobileNetV3 mencapai akurasi tertinggi sebesar 79% dan terbukti lebih efisien dibandingkan model yang lebih kompleks. Namun, penelitian ini masih terbatas pada klasifikasi tanpa deteksi spasial dan tidak mengevaluasi kinerja model dari segi kecepatan inferensi maupun ukuran model.

Dari ketiga penelitian tersebut dapat disimpulkan bahwa meskipun CNN telah terbukti efektif untuk klasifikasi penyakit tanaman, masih terdapat kesenjangan penelitian dalam penerapan arsitektur SSD untuk tugas deteksi penyakit secara spasial dengan mempertimbangkan *trade-off* antara akurasi dan efisiensi. Oleh karena itu, penelitian ini berupaya mengisi celah tersebut dengan melakukan analisis komparatif terhadap tiga arsitektur SSD VGG16, ResNet18, dan MobileNetV2 untuk mendeteksi penyakit *brown spot* pada padi, guna memperoleh pemahaman empiris mengenai keseimbangan optimal antara presisi dan performa komputasi.

3. METODE PENELITIAN

3.1. Pengumpulan Data

Penelitian ini menggunakan dataset citra daun padi publik dari platform Kaggle (Sankalana, 2020). Dari dataset tersebut, sebanyak 1000 citra yang mengandung penyakit *Brown Spot* dipilih untuk eksperimen. Untuk melatih model deteksi objek, dilakukan proses anotasi manual menggunakan perangkat lunak *Labellmg*. Setiap area yang teridentifikasi *Brown Spot* diberi kotak pembatas (*bounding box*) dan disimpan dalam format *label.xml*.



Gambar 1. Citra Penyakit brownspot

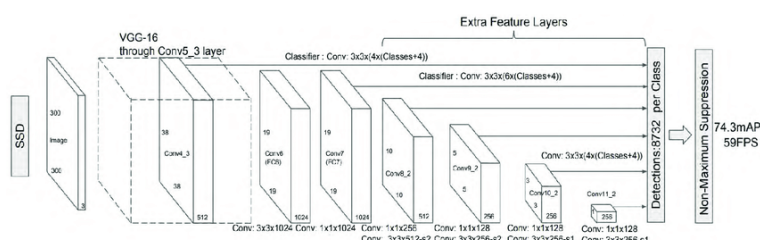
Kumpulan data yang telah dianotasi ini terdiri dari 1000 citra dan 1000 file label kemudian diunggah oleh peneliti sebagai dataset baru ke platform Kaggle untuk digunakan dalam lingkungan kerja (*Kaggle Notebook*). Dataset inilah yang menjadi sumber data utama yang siap digunakan untuk tahap pra-pemrosesan. Penelitian ini difokuskan sebagai deteksi satu kelas (*single-class detection*), di mana model dilatih untuk membedakan antara kelas *Brownspot* dan *Background*.

3.2. Pre-Processing

Penelitian ini memanfaatkan dataset yang telah melalui proses augmentasi awal (*offline augmentation*) dari penyedia dataset untuk mengatasi keterbatasan jumlah sampel data latih tanpa menambah beban komputasi melalui augmentasi *on-the-fly* selama pelatihan. Berdasarkan observasi karakteristik dataset, teknik augmentasi yang diberikan mencakup manipulasi posisi citra, yaitu pembalikan citra secara horizontal dan vertikal untuk mensimulasikan variasi orientasi daun, serta rotasi sudut tetap seperti 90°, 180°, dan 270° guna menghasilkan perubahan orientasi tanpa mengganggu struktur visual daun. Setelah proses augmentasi awal tersebut, tahap pra-pemrosesan difokuskan pada penyeragaman dimensi masukan dengan melakukan *resize* seluruh citra beserta anotasinya menjadi 300×300 piksel agar sesuai dengan spesifikasi arsitektur SSD300. Selanjutnya, dataset dibagi secara acak menjadi dua subset, yaitu 80% (800 citra) sebagai data latih dan 20% (200 citra) sebagai data validasi [9].

3.3. Arsitektur Model

Kerangka kerja deteksi objek utama yang digunakan dalam penelitian ini adalah Single Shot Detector (SSD). Metode SSD dipilih karena kemampuannya yang unggul dalam deteksi objek secara real-time. Berbeda dengan arsitektur dua tahap seperti R-CNN, SSD melakukan prediksi lokasi (bounding box) dan klasifikasi objek secara simultan dalam satu kali proses inferensi (single pass), menjadikannya sangat efisien [10]. Penelitian ini menggunakan varian SSD300 yang menerima citra masukan beresolusi 300×300 piksel. Fokus utama analisis terletak pada evaluasi tiga backbone berbeda yang terintegrasi dengan kerangka kerja SSD untuk mengkaji trade-off antara akurasi deteksi dan efisiensi komputasi.

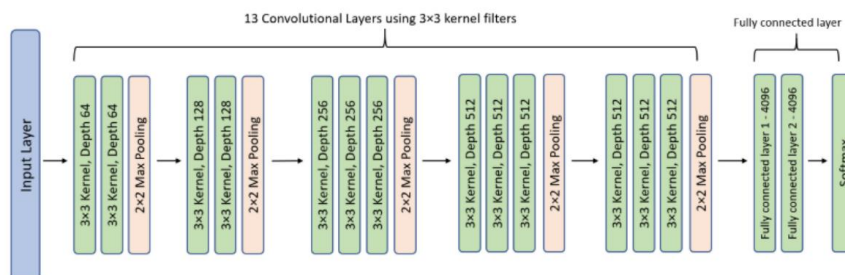


Gambar 2. Arsitektur SSD300

Fokus utama dari analisis komparatif ini terletak pada penggunaan arsitektur *backbone* (jaringan ekstraktor fitur) yang berbeda, yang terintegrasi dengan kerangka kerja SSD. Tiga backbone yang berbeda diimplementasikan dan dievaluasi untuk menganalisis secara kuantitatif *trade-off* antara akurasi deteksi dan efisiensi komputasi.

3.3.1. SSD-VGG16

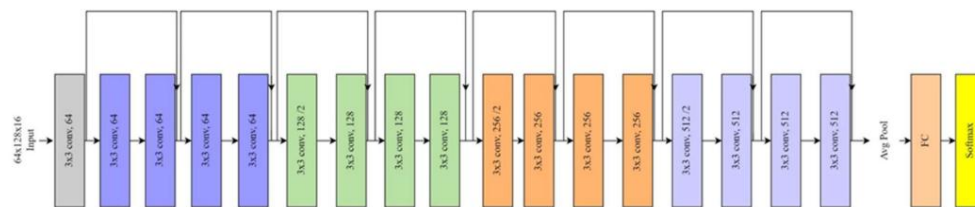
VGG16 digunakan sebagai *backbone baseline* atau pembanding utama karena secara historis merupakan *backbone* standar pada arsitektur SSD orisinal [11]. Berdasarkan implementasi, model ini diinisialisasi menggunakan `ssd300_vgg16` yang telah tersedia di *library torchvision.models.detection*. Implementasi memanfaatkan bobot *pre-trained SSD300_VGG16_Weights.DEFAULT*, dengan modifikasi pada *classification head* untuk menyesuaikan jumlah kelas penelitian (`num_classes=2`). Pendekatan ini merupakan bentuk *transfer learning* standar dengan penyesuaian lapisan akhir model [11].



Gambar 2. Arsitektur VGG16

3.3.2. SSD-ResNet18

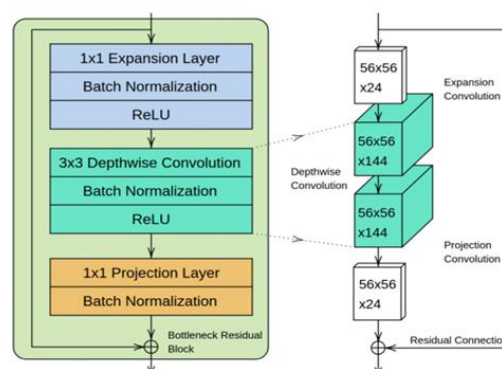
Arsitektur ini menggunakan ResNet18 sebagai backbone. Berbeda dengan VGG16, library torchvision tidak menyediakan model `ssd300_resnet18` siap pakai, sehingga arsitektur harus dibangun secara manual. Model ResNet18 pre-trained dimuat, dengan menghapus layer `avgpool` dan `fully connected` untuk mengambil feature map dasar berukuran 512 channel. Karena SSD membutuhkan feature maps dari berbagai skala, serangkaian extra convolutional layers ditambahkan secara manual untuk menghasilkan representasi multi-skala yang diperlukan oleh detection head, sebagaimana dijelaskan dalam penelitian serupa yang memanfaatkan ResNet sebagai backbone SSD [12], [13].



Gambar 3. Arsitektur RestNet18

3.3.3. SSD-MobileNetV2

Arsitektur ini menggunakan MobileNetV2 sebagai backbone. Serupa dengan implementasi ResNet18, arsitektur SSD-MobileNetV2 merupakan rakitan kustom karena tidak tersedia secara default di torchvision. Bagian ekstraktor fitur dari model `mobilenet_v2` pre-trained dimuat, yang terdiri dari blok inverted residual dan depthwise separable convolutions sehingga lebih ringan dan efisien. Feature map keluaran backbone ini memiliki 1280 channel, dan seperti pada ResNet18, extra convolutional layers ditambahkan secara manual untuk menciptakan multi-scale feature maps yang diperlukan oleh SSD [14].



Gambar 4. Arsitektur MobileNetV2

3.4. Hyperparameter

Ketiga model (SSD-VGG16, SSD-ResNet18, SSD-MobileNetV2) dilatih menggunakan *framework* PyTorch. Proses pelatihan dieksekusi pada platform Kaggle dengan *hardware* GPU NVIDIA T4 x2. Untuk perbandingan yang adil, ketiga model dilatih menggunakan pengaturan *hyper-parameter* yang sama.

Tabel 1. *Hyperparameter*

<i>Hyper-parameter</i>	<i>Value</i>
Optimizer	Stochastic Gradient Descent
Learning Rate	0.002
Momentum	0.9
Weight Decay	5e-4
Batch Size	64
Epochs	200

3.5. Metode Evaluasi

Kinerja model dievaluasi menggunakan dua metrik standar dalam deteksi objek, yaitu *Intersection over Union* (IoU) dan *mean Average Precision* (mAP). Pemilihan kedua metrik ini didasarkan pada pertimbangan komprehensif: IoU digunakan untuk mengukur akurasi lokalisasi (bounding box), sedangkan mAP digunakan untuk mengukur ketepatan klasifikasi sekaligus sensitivitas model terhadap objek yang ada [15].

Intersection over Union (IoU) IoU adalah metrik evaluasi fundamental yang mengukur seberapa akurat kotak prediksi (predicted box) menutupi objek yang sebenarnya (ground truth). Secara teknis, IoU dihitung sebagai rasio antara luas area irisan (intersection) dan luas area gabungan (union) dari kedua kotak tersebut. Rumus IoU didefinisikan sebagai:

$$IoU = \frac{Area(B_p \cap B_{gt})}{Area(B_p \cup B_{gt})}$$

Di mana B_p adalah bounding box hasil prediksi dan B_{gt} adalah bounding box anotasi asli (*ground truth*). Sebuah prediksi dianggap sebagai True Positive hanya jika nilai IoU melampaui ambang batas (*threshold*) tertentu, yang memastikan bahwa deteksi tidak hanya benar secara kelas, tetapi juga akurat secara posisi [15].

mean Average Precision (mAP) mAP merupakan rata-rata nilai *Average Precision* (AP) dari seluruh kelas yang diuji. Metrik ini dipilih karena kemampuannya menggambarkan kurva *Precision-Recall* (PR), yaitu keseimbangan antara seberapa akurat prediksi model (*Precision*) dan seberapa banyak objek asli yang berhasil ditemukan (*Recall*) [16].

$$AP = \frac{1}{11} \sum_{\{r \in \{0,0.1,...,1\}\}} p_{\{interp\}}(r)$$

Pendekatan ini memastikan bahwa fluktuasi kecil pada kurva presisi-recall tidak membiaskan evaluasi kinerja model secara keseluruhan.

4. HASIL DAN PEMBAHASAN

Bab ini menyajikan analisis komparatif kinerja tiga arsitektur *backbone* SSD (VGG16, ResNet18, dan MobileNetV2) dalam mendeteksi penyakit *brown spot*. Evaluasi dilakukan berdasarkan metrik akurasi (mAP, IoU) dan efisiensi (Ukuran Model, FPS).

4.1. Dinamika Pelatihan Model

Proses pelatihan ketiga model selama 200 epoch divisualisasikan dalam Gambar 5. Grafik tersebut memperlihatkan perbandingan antara penurunan *training loss* (kurva biru) dan peningkatan *validation mAP* (kurva merah). Secara umum, SSD-VGG16 menunjukkan stabilitas pelatihan terbaik dengan konvergensi loss yang paling rendah, mengindikasikan kapasitas model yang besar dalam mempelajari fitur kompleks dari citra



penyakit. Sebaliknya, pada grafik *SSD-MobileNetV2*, terlihat kesenjangan (gap) yang lebih lebar antara loss pelatihan dan akurasi validasi dibandingkan dua model lainnya. Fenomena ini mengindikasikan bahwa arsitektur yang sangat ringan memiliki kecenderungan lebih cepat mencapai batas kapasitas belajarnya (*capacity saturation*) dibandingkan arsitektur yang lebih dalam.

Gambar 5. Perbandingan training loss (biru) dan validation mAP (merah) pada tiga arsitektur backbone SSD: VGG16, MobileNetV2, dan ResNet18

4.2. Evaluasi Kinerja Kuantitatif

Rangkuman performa akhir dari ketiga model setelah dilakukan pengujian silang rata-rata (*average runs*) disajikan secara rinci pada Tabel 2.

Tabel 2. Hasil Perbandingan Kinerja Tiga Arsitektur Backbone SSD

Model	mAP (<i>mean</i> \pm <i>std</i>)	IoU (<i>mean</i> \pm <i>std</i>)	Ukuran Model	Kecepatan (FPS)
VVG16	(0.8253 \pm 0.0152)	(0.5477 \pm 0.0171)	90.71 MB	11.53
ResNet18	(0.7959 \pm 0.0152)	(0.5130 \pm 0.0236)	55.90 MB	21.08
MobileNetV2	(0.7068 \pm 0.0178)	(0.4680 \pm 0.0123)	43.73 MB	29.42

Berdasarkan data pada Tabel 2, terlihat pola *trade-off* yang jelas antara akurasi dan efisiensi komputasi. SSD-VGG16 menunjukkan performa terbaik dalam hal akurasi dengan nilai mAP 0.825 dan IoU 0.548. Keunggulan ini konsisten dengan temuan Chatterjee dan Ghosh [6], yang menjelaskan bahwa kedalaman jaringan serta jumlah parameter yang besar pada arsitektur VGG memungkinkan ekstraksi fitur visual yang lebih kaya dan stabil. Namun, kompleksitas ini berdampak pada efisiensi komputasi; model memiliki ukuran terbesar (90.71 MB) dan kecepatan inferensi terendah (11.53 FPS), yang berada jauh di bawah ambang batas real-time.

Sebaliknya, SSD-MobileNetV2 memperlihatkan karakteristik yang berlawanan. Model ini menonjol dalam efisiensi, dengan ukuran file terkecil (43.73 MB) dan kecepatan inferensi tertinggi, mencapai 29.42 FPS. Hasil ini sejalan dengan penelitian Wang et al. [7], yang menyoroti potensi arsitektur ringan berbasis MobileNet dalam penerapan deteksi objek real-time pada perangkat dengan keterbatasan sumber daya. Kecepatan 29 FPS menjadikan SSD-MobileNetV2 satu-satunya model yang memenuhi kriteria real-time inference (>25 FPS). Namun, efisiensi ini diiringi oleh penurunan akurasi hingga mAP 0.707, yang mengonfirmasi bahwa penggunaan *depthwise separable convolution*, meskipun efisien, mengurangi kedalaman representasi fitur dibandingkan konvolusi standar pada VGG.

Sebagai kompromi, SSD-ResNet18 menunjukkan keseimbangan terbaik antara akurasi dan efisiensi. Dengan nilai mAP 0.796 dan kecepatan 21.08 FPS, model ini mendekati akurasi SSD-VGG16 namun dengan ukuran model yang jauh lebih kecil dan kecepatan inferensi lebih tinggi. Hasil ini sejalan dengan temuan Roseno et al. [8], yang menunjukkan bahwa arsitektur ResNet memiliki kemampuan generalisasi yang baik berkat mekanisme *residual connection*, sehingga dapat mempertahankan performa tinggi tanpa meningkatkan jumlah parameter secara signifikan.

Secara keseluruhan, hasil ini memperkuat kesimpulan bahwa pemilihan *backbone* sangat memengaruhi *trade-off* antara akurasi dan efisiensi komputasi pada arsitektur SSD. SSD-VGG16 unggul untuk aplikasi yang menuntut akurasi tinggi, SSD-ResNet18 menawarkan keseimbangan optimal, sementara SSD-MobileNetV2 menjadi pilihan terbaik untuk sistem real-time yang beroperasi pada perangkat dengan sumber daya terbatas.

4.3. Validasi Visual Deteksi

Kualitas deteksi model juga divalidasi secara kualitatif. Seperti diperlihatkan pada Gambar 6, model mampu melokalisasi bercak *brown spot* dengan *bounding box* yang cukup presisi. Meskipun ketiga model mampu mendeteksi objek, variasi nilai IoU pada Tabel 2 mencerminkan seberapa rapat kotak prediksi tersebut menempel pada objek asli. VGG16 cenderung menghasilkan kotak yang sangat pas (*tight fit*), sementara MobileNetV2 terkadang menghasilkan kotak yang sedikit bergeser atau kurang presisi pada tepi bercak yang kabur.



Gambar 6. Visualisasi sampel citra daun padi dengan anotasi bounding box yang menandai area penyakit brown spot

Temuan keseluruhan penelitian ini memperjelas posisi masing-masing arsitektur dalam spektrum aplikasi pertanian presisi. Jika Simhadri dkk. [3] menekankan pentingnya pergeseran ke arah model ringan (*lightweight*) untuk implementasi lapangan, hasil penelitian ini memberikan batasan empiris bahwa pergeseran tersebut (menggunakan MobileNetV2) akan mengorbankan sekitar 12% akurasi (selisih mAP 0.825 vs 0.707) dibandingkan model berat. Oleh karena itu, pemilihan model tidak dapat digeneralisasi, melainkan harus disesuaikan dengan prioritas spesifik pengguna: apakah mengejar presisi diagnosis maksimal atau responsivitas sistem di lapangan.

5. KESIMPULAN

Penelitian ini memberikan kontribusi empiris dengan memetakan *trade-off* krusial antara akurasi deteksi dan efisiensi komputasi pada arsitektur SSD untuk kasus penyakit *brown spot* padi. Hasil evaluasi menegaskan bahwa pemilihan *backbone* sangat bergantung pada skenario implementasi; SSD-VGG16 terbukti memberikan kontribusi akurasi tertinggi (mAP 0,825) yang ideal untuk analisis presisi berbasis server (*offline*), sementara SSD-MobileNetV2 menawarkan efisiensi terbaik sebagai satu-satunya model yang mencapai kecepatan real-time (29,42 FPS), menjadikannya solusi paling layak untuk implementasi pada perangkat *mobile* di lapangan meskipun dengan sedikit penurunan akurasi.

Berdasarkan temuan tersebut, penelitian selanjutnya disarankan untuk berfokus pada upaya menjembatani kesenjangan antara akurasi dan kecepatan. Pengembangan di

masa depan dapat mengeksplorasi integrasi mekanisme atensi (*attention modules*) pada arsitektur ringan seperti MobileNetV2 untuk mendongkrak akurasi tanpa membebani komputasi secara signifikan, atau menerapkan teknik kompresi model (seperti *pruning* dan *quantization*) pada model yang lebih berat agar dapat beroperasi lebih efisien di perangkat dengan sumber daya terbatas.

DAFTAR PUSTAKA

- [1] Y. M. Shabana, A. E. Ismail, Y. M. Rashad, and D. Survey, "Control of brown spot pathogen of rice (*Bipolaris oryzae*) using some phenolic components," *Brazilian J. Microbiol.*, vol. 39, pp. 438–444, 2008.
- [2] V. Agarwal, M. Kshirsagar, R. Jain, and S. Jain, "Smart Mobile System for Detection and Classification of the (*Oryza Sativa*) Rice Plant Disease Using Deep Learning and Transfer Learning," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 10, no. 8, pp. 1935–1940, 2022, doi: 10.22214/ijraset.2022.46554.
- [3] C. G. Simhadri, H. K. Kondaveeti, V. K. Vatsavayi, A. Mitra, and P. Ananthachari, "Deep learning for rice leaf disease detection: A systematic literature review on emerging trends, methodologies and techniques," *Inf. Process. Agric.*, vol. 12, no. 2, pp. 151–168, 2025, doi: 10.1016/j.inpa.2024.04.006.
- [4] L. Wang *et al.*, "A novel deep learning-based single shot multibox detector model for object detection in optical remote sensing images," *Geosci. Data J.*, vol. 11, no. 3, pp. 237–251, 2024, doi: 10.1002/gdj3.162.
- [5] Y. Xu, D. Li, C. Li, Z. Yuan, and Z. Dai, "LiSA-MobileNetV2: an extremely lightweight deep learning model with Swish activation and attention mechanism for accurate rice disease classification," *Front. Plant Sci.*, vol. 16, no. August, pp. 1–14, 2025, doi: 10.3389/fpls.2025.1619365.
- [6] Sumanta Chatterjee, "PlantOHealth: Comparative Evaluation of Deep Learning Models for Plant Disease Detection Using Leaf Images," *J. Inf. Syst. Eng. Manag.*, vol. 10, no. 46s, pp. 661–680, 2025, doi: 10.52783/jisem.v10i46s.9006.
- [7] L. Wang *et al.*, "Transfer Learning-Based Lightweight SSD Model for Detection of Pests in Citrus," *Agronomy*, vol. 13, no. 7, pp. 1–15, 2023, doi: 10.3390/agronomy13071710.
- [8] M. T. Roseno, S. Oktarina, Y. Nearti, H. Syaputra, and N. Jayanti, "Comparing CNN Models for Rice Disease Detection: ResNet50, VGG16, and MobileNetV3-Small," *J. Inf. Syst. Informatics*, vol. 6, no. 3, pp. 2099–2109, 2024, doi: 10.51519/journalisi.v6i3.865.
- [9] Z. Zhang, J. Hou, and Z. Lin, "Exploring Integration methods for Image Data Augmentation," *2024 6th Int. Conf. Internet Things, Autom. Artif. Intell. IoTAAI 2024*, pp. 707–713, 2024, doi: 10.1109/IoTAAI62601.2024.10692583.
- [10] Hao Zhang, X. Hong, and L. Zhu, "Detecting Small Objects in Thermal Images Using Single-Shot Detector," *Autom. Control Comput. Sci.*, vol. 55, no. 2, pp. 202–211, Mar. 2021, doi: 10.3103/S0146411621020097.
- [11] Y. Kang, "Research on SSD base network," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 768, no. 7, pp. 0–7, 2020, doi: 10.1088/1757-899X/768/7/072031.
- [12] F. Sun, X. Zhang, Y. Liu, and H. Jiang, "Multi-Object Detection in Security Screening Scene Based on Convolutional Neural Network," *Sensors*, vol. 22, no. 20, 2022, doi: 10.3390/s22207836.

- [13] Y. Zhou, H. Qian, and P. Ding, "MSSD: multi-scale object detector based on spatial pyramid depthwise convolution and efficient channel attention mechanism," *J. Real-Time Image Process.*, vol. 20, no. 5, p. 103, Oct. 2023, doi: 10.1007/s11554-023-01358-9.
- [14] X. Zhong, "CAL-SSD: lightweight SSD object detection based on coordinated attention," *Signal, Image Video Process.*, vol. 19, no. 1, p. 31, Jan. 2025, doi: 10.1007/s11760-024-03716-x.
- [15] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU loss: Faster and better learning for bounding box regression," *AAAI 2020 - 34th AAAI Conf. Artif. Intell.*, no. 2, pp. 12993–13000, 2020, doi: 10.1609/aaai.v34i07.6999.
- [16] U. Alganci, M. Soydas, and E. Sertel, "Comparative research on deep learning approaches for airplane detection from very high-resolution satellite images," *Remote Sens.*, vol. 12, no. 3, 2020, doi: 10.3390/rs12030458.