IT-EXPLORE
Jurnal Penerapan Teknologi Informasi dan Komunikasi
ISSN 2828-7940, e-ISSN 2829-1727, DOI: 10.24246/itexplore.v5i2.2026.pp232-240
Volume 05 Nomor 02 Tahun 2026
Journal homepage: https://ejournal.uksw.edu/itexplore

Klasifikasi serangan *malware* berbasis citra menggunakan algoritma YOLOv11

Daniel Darren Richardo [1], Theophilus Wellem [2] *
[1,2] Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana, Salatiga, Indonesia

Informasi Artikel

Riwayat Artikel:

Dikirim:
12 Nopember 2025
Direvisi:
13 Februari 2026
Diterima:
28 Maret 2026

Kata Kunci:

Malware,
YOLOv11m,
Deep Learning,
Klasifikasi Citra

ABSTRAK

Malware merupakan ancaman keamanan siber yang terus berkembang dan menuntut metode deteksi yang lebih efektif. Sistem deteksi konvensional berbasis *signature* memiliki keterbatasan dalam mengidentifikasi varian baru mendorong pengembangan pendekatan berbasis deep learning. Penelitian ini mengimplementasikan dan mengevaluasi empat varian algoritma YOLOv11 (n, s, m, l) untuk klasifikasi *malware* berbasis representasi visual citra. Dataset yang digunakan terdiri dari 22.056 citra *malware* dan *benign*, yang dibagi menjadi 70% *training*, 15% *validation*, dan 15% *testing* dengan 8 kelas (*adware*, *backdoor*, *benign*, *downloader*, *spyware*, *trojan*, *virus*, *worm*). Setiap model dilatih selama 100 *epoch* dengan *batch size* 32 menggunakan Google Colab dengan GPU. Hasil penelitian menunjukkan bahwa semua varian mencapai akurasi tinggi (97,8%-98,1%) dengan YOLOv11m sebagai model terbaik (98,1%). YOLOv11n menawarkan keseimbangan optimal antara akurasi (97,9%) dan efisiensi (1,5M parameter, 0,3 ms/img inferensi) yang ideal untuk aplikasi real-time. Penelitian ini melampaui metode sebelumnya seperti *k*-NN (97,18%) dan *hybrid* CNN (96,55%) dengan keunggulan kecepatan inferensi yang sangat superior (0,3-0,9 ms/img vs puluhan hingga ratusan ms/img), membuktikan efektivitas YOLOv11 untuk deteksi *malware* yang cepat, akurat, dan *scalable*.

Keywords:

Malware,
YOLOv11m,
Deep Learning,
Image Classification

ABSTRACT

Malware represents an evolving cybersecurity threat that demands more effective detection methods. Conventional signature-based detection systems have limitations in identifying new variants, driving the development of deep learning-based approaches. This research implements and evaluates four variants of the YOLOv11 algorithm (n, s, m, l) for malware classification based on visual image representation. The dataset consists of 22,056 malware and benign images, divided into 70% training, 15% validation, and 15% testing across 8 classes (adware, backdoor, benign, downloader, spyware, trojan, virus, worm). Each model was trained for 100 epochs with batch size 32 using Google Colab with GPU support. Results demonstrate that all variants achieve high accuracy (97.8%-98.1%) with YOLOv11m as the best performer (98.1%). YOLOv11n offers optimal balance between accuracy (97.9%) and efficiency (1.5M parameters, 0.3 ms/img inference) ideal for real-time applications. This research surpasses previous methods such as K-NN (97.18%) and hybrid CNN (96.55%) with superior inference speed (0.3-0.9 ms/img vs tens to hundreds of ms/img), proving the effectiveness of YOLOv11 for fast, accurate, and scalable malware detection.

Corresponding Author:

Theophilus Wellem
Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana
Jl. O. Notohamidjojo 1-10, Salatiga, Indonesia
Email: theophilus.wellem@uksw.edu

1. PENDAHULUAN

Ancaman keamanan siber dalam bentuk perangkat lunak berbahaya atau malware menjadi semakin kompleks seiring perkembangan teknologi informasi yang pesat [1]. *Malicious Software (malware*) adalah program komputer yang bertujuan untuk merusak, mengganggu, atau memperoleh akses ilegal ke sistem tanpa sepengetahuan pengguna [2]. Sistem deteksi *malware* konvensional yang mengandalkan *byte pattern* (*signature-based detection*) menunjukkan keterbatasan signifikan dalam mengidentifikasi varian *malware* baru atau yang telah mengalami modifikasi dengan *malware* yang sudah ada sebelumnya [3]. Kondisi ini mendorong pengembangan metode inovatif yang memanfaatkan teknologi *machine learning* dan *deep learning* untuk menutupi kelemahan tersebut. Di antara berbagai pendekatan yang dikembangkan, transformasi file *malware* menjadi representasi visual dalam bentuk citra yang dianalisis menggunakan teknik *computer vision* menunjukkan potensi yang signifikan [4].

Pendekatan ini memungkinkan pemanfaatan struktur visual yang unik dari setiap jenis *malware* sebagai dasar identifikasi. Algoritma *You Only Look Once* (YOLO) telah terbukti sebagai salah satu metode deteksi objek paling efektif dalam bidang *computer vision* [5]. YOLOv11 yang merupakan salah satu algortima terbaru dari keluarga YOLO, menghadirkan berbagai penyempurnaan signifikan dalam aspek akurasi dan efisiensi komputasi dibandingkan dengan versi-versi sebelumnya [6]. Meskipun berbagai pendekatan *deep learning berbasis Convolutional Neural Network* (CNN) telah menunjukkan performa yang baik dalam klasifikasi *malware* berbasis citra, sebagian besar penelitian terdahulu masih menghadapi keterbatasan pada latensi inferensi yang relatif tinggi. Keterbatasan ini menjadi kendala utama dalam penerapan sistem deteksi malware secara *real-time*, terutama pada lingkungan jaringan dan sistem yang menuntut respons cepat.

Oleh karena itu, diperlukan model yang tidak hanya akurat, tetapi juga mampu melakukan inferensi secara cepat dan efisien. YOLOv11 dipilih dalam penelitian ini karena arsitektur terbarunya, seperti blok *Cross Stage Partial Kernel Size 2* (C3K2) dan modul *Spatial Pyramid Pooling–Fast* (SPFF), dirancang untuk mengekstraksi fitur spasial secara efisien pada citra bertekstur kompleks. Karakteristik ini menjadikan YOLOv11 sangat relevan untuk citra *malware* yang bersifat abstrak dan tidak menyerupai objek natural. Selain itu, model YOLOv11 tersedia dalam lima varian: n (*nano*) untuk efisiensi maksimal pada perangkat terbatas, s (*small*) untuk keseimbangan optimal, m (*medium*) untuk akurasi lebih tinggi, l (*large*) untuk performa *superior*, dan x (*extra-large*) untuk akurasi maksimal dengan kebutuhan komputasi tertinggi.

Berdasarkan hal tersebut, penelitian ini bertujuan untuk mengevaluasi efektivitas empat varian YOLOv11 (n, s, m, l) dalam klasifikasi citra *malware* dan *benign* dari dataset *Malware Benign Image Classification Dataset*, menganalisis *trade-off* antara akurasi dan efisiensi komputasi dari masing-masing varian, serta mengidentifikasi model yang paling sesuai untuk berbagai skenario *deployment* menggunakan pembagian dataset dengan proporsi *dataset* 70% untuk *training*, 15% untuk *validation*, dan 15% untuk *testing*. Sementara secara praktis, penelitian ini diharapkan dapat menghasilkan sistem deteksi *malware* dengan akurasi tinggi dan efisiensi yang optimal, memberikan solusi aplikatif bagi organisasi dalam meningkatkan keamanan infrastruktur teknologi informasi, serta menjadi dasar bagi pengembangan sistem deteksi *malware* secara *real-time* di masa depan.

2. KAJIAN PUSTAKA

Malware merupakan perangkat lunak berbahaya yang dirancang untuk merusak atau memperoleh akses ilegal pada sistem komputer [5]. Jenisnya sangat beragam dengan mekanisme serangan yang berbeda-beda, dan setiap jenis *malware* memiliki cara kerja serta tujuan yang khas dalam menyerang sistem. Sebagai contoh, *virus* mampu mereplikasi diri dan menginfeksi *file* lain, sedangkan *worm* dapat menyebar otomatis melalui jaringan tanpa memerlukan *host file*. *Trojan* biasanya menyamar sebagai aplikasi sah untuk menipu pengguna, sementara *ransomware* mengenkripsi data korban dan meminta tebusan. Selain itu, *spyware* berfungsi mencuri informasi pribadi, *adware* menampilkan iklan yang mengganggu, dan *backdoor*

memberi akses ilegal berkelanjutan bagi penyerang [5]. Keragaman karakteristik tersebut menjadikan deteksi *malware* sebagai permasalahan yang kompleks dalam bidang keamanan siber.

Metode deteksi *malware* telah berkembang dari pendekatan konvensional menuju teknik berbasis pembelajaran mesin. Pendekatan berbasis pola atau *pattern* (*signature-based detection*) efektif untuk *malware* yang sudah dikenal, namun tidak berlaku untuk mengenali varian baru atau *malware* yang telah dimodifikasi (*obfuscated*). Analisis heuristik mencoba mengatasi hal ini dengan mendeteksi perilaku mencurigakan, tetapi sering menghasilkan tingkat kesalahan (*false positive*) yang cenderung tinggi [6]. Keterbatasan metode ini mendorong pengembangan metode berbasis *machine learning* dan *deep learning* yang dapat mempelajari pola kompleks pada komputer secara otomatis tanpa campur tangan manual.

Salah satu pendekatan inovatif adalah mengubah file biner *malware* menjadi representasi visual dalam bentuk citra *grayscale* atau RGB. Teknik ini pertama kali diperkenalkan oleh Nataraj dkk. yang mengonversi setiap *byte* dari file *executable* menjadi nilai piksel sehingga terbentuk pola visual khas untuk setiap keluarga *malware* [4]. Pendekatan berbasis citra memungkinkan pemanfaatan teknik *computer vision* tanpa memerlukan proses *reverse engineering* manual serta dilaporkan memiliki ketahanan yang lebih baik terhadap variasi kode lokal. Penelitian lanjutan juga menunjukkan bahwa representasi citra berbasis perilaku *runtime* relatif lebih stabil terhadap teknik *obfuscation* dan *polymorphism*, karena perubahan kode lokal tidak selalu mengubah pola spasial global citra secara signifikan [7]. Dalam konteks ini, arsitektur *deep learning* yang mampu mengekstraksi fitur spasial global, seperti YOLOv11, menjadi relevan karena dapat menangkap pola distribusi visual utama yang tetap konsisten meskipun terjadi modifikasi kode lokal akibat *obfuscation* atau *polymorphism*.

Pada penelitian ini, konversi *malware* menjadi citra tidak dilakukan melalui pemetaan *byte* satu dimensi ke dua dimensi secara langsung. Sebaliknya, citra dibentuk berdasarkan representasi perilaku runtime *malware* yang diperoleh melalui analisis dinamis menggunakan *Cuckoo Sandbox*. Urutan pemanggilan API dan argumennya diekstraksi dan diproses menggunakan pendekatan *n-gram* untuk menghasilkan vektor fitur numerik. Vektor ini kemudian dinormalisasi ke rentang 0–255 dan dipetakan ke dalam matriks dua dimensi dengan ukuran tetap. Dengan demikian, variasi ukuran *file* biner tidak memengaruhi struktur spasial citra *input*, sehingga setiap sampel *malware* direpresentasikan secara konsisten dan dapat dibandingkan antar kelas. Pada konteks klasifikasi *malware* berbasis citra, berbagai arsitektur *deep learning* telah diterapkan dengan hasil yang menjanjikan. Venkatraman dkk. mengembangkan pendekatan *hybrid deep learning* berbasis CNN yang mencapai akurasi 96,55% [8]. Penelitian ini membuktikan bahwa *deep learning* mampu mengekstrak fitur visual secara hierarkis tanpa proses *feature engineering* manual. Namun, sebagian besar penelitian sebelumnya menggunakan CNN konvensional yang memerlukan waktu inferensi cukup lama sehingga kurang ideal untuk aplikasi *real-time*.

Algoritma YOLO menjadi solusi potensial karena merupakan metode deteksi objek *single-stage* yang mampu melakukan prediksi dalam satu *forward pass* melalui jaringan [9]. Keunggulannya terletak pada kecepatan dan efisiensi, menjadikannya ideal untuk sistem klasifikasi *real-time*. Versi terbaru, YOLOv11, membawa peningkatan signifikan melalui arsitektur *backbone* baru dengan blok *Cross Stage Partial Kernel Size 2* (C3K2) dan modul *Spatial Pyramid Pooling – Fast* (SPFF) yang meningkatkan ekstraksi fitur multi-skala serta konteks spasial [10]. Meskipun awalnya algortima ini dirancang untuk deteksi objek, kemampuan ekstraksi fiturnya yang kuat membuat YOLOv11 sangat potensial untuk melakukan tugas klasifikasi *malware* berbasis citra. Berdasarkan kajian terhadap penelitian-penelitian sebelumnya, dapat disimpulkan bahwa meskipun pendekatan klasifikasi citra *malware* telah menunjukkan performa yang baik, masih terdapat celah penelitian pada aspek efisiensi komputasi dan latensi inferensi untuk mendukung aplikasi *real-time*. Oleh karena itu, diperlukan evaluasi lebih lanjut terhadap arsitektur *deep learning* yang mampu memberikan keseimbangan antara akurasi dan efisiensi komputasi. Penelitian ini mengisi celah tersebut dengan mengevaluasi performa beberapa varian YOLOv11 dalam tugas klasifikasi citra *malware* dan *benign*.

3. METODE PENELITIAN

Penelitian ini menggunakan pendekatan kuantitatif dengan metode eksperimen untuk menguji performa pada empat varian algoritma YOLOv11 (YOLOv11n, YOLOv11s, YOLOv11m, dan YOLOv11l) dalam klasifikasi *malware* berbasis citra. Untuk menjaga *reproducibility*, seluruh eksperimen dijalankan dengan menetapkan *random seed* 42 pada proses pembagian data dan inisialisasi model. Tahapan penelitian ditunjukkan pada Gambar 1, dimulai dari menyiapkan *dataset*, *preprocessing*, *training* model, evaluasi hasil, hingga *testing*.

Gambar 1: Tahapan penelitian

Dataset yang digunakan berasal dari *Malware Benign Image Classification Dataset* dengan total 22.056 citra *malware* dan *benign* [11]. Dataset ini terdiri dari beberapa kategori *malware*, antara lain *adware*, *trojan*, *virus*, *worm*, *spyware*, *backdoor*, dan *downloader*, serta kategori *benign*. Distribusi data untuk masing-masing kelas ditunjukkan pada Gambar 2. Untuk menjaga konsistensi dan keseimbangan data, setiap kelas dikurangi secara acak sebesar 50% sehingga diperoleh distribusi yang lebih terkontrol. Pengurangan dataset sebesar 50% dilakukan secara acak dan proporsional pada setiap kelas untuk mengatasi ketidakseimbangan data, khususnya pada kelas *benign* yang memiliki jumlah sampel lebih dominan. Pendekatan ini bertujuan untuk mengurangi *bias* model terhadap kelas mayoritas, meningkatkan generalisasi model, serta menekan beban komputasi selama proses *training* tanpa menghilangkan karakteristik utama dari masing-masing kelas *malware*.

Dataset yang digunakan dalam penelitian ini kemudian dibagi menggunakan metode *stratified sampling* dengan proporsi 70% untuk *training* (7.718 citra), 15% untuk *validation* (1.654 citra), dan 15% untuk *testing* (1.655 citra). Tahap *preprocessing* dilakukan dengan mengubah ukuran citra menjadi 224×224 piksel agar sesuai dengan kebutuhan input model YOLOv11. Selain itu, citra dinormalisasi secara otomatis melalui *pipeline default* YOLO. Pada penelitian ini, *data augmentation* tidak diperlukan karena variasi data sudah mencukupi untuk proses pembelajaran model.

	Class	Jumlah Gambar
0	adware	993
1	downloader	1249
2	worm	678
3	virus	1196
4	trojan	1784
5	benign	4317
6	backdoor	337
7	spyware	473

Gambar 2: Distribusi data masing-masing kelas pada dataset

Setiap varian YOLOv11 dilatih selama 100 *epoch* dengan *batch size* 32, *learning rate* 0.001, dan *optimizer* AdamW. Selama proses pelatihan, model secara otomatis menyimpan *weights* terbaik berdasarkan kinerja validasi. Proses ini dilakukan menggunakan Google Colab dengan dukungan GPU, dengan total waktu komputasi sekitar 7 jam. Evaluasi model dilakukan pada data validasi dan data uji menggunakan metrik akurasi, *precision*, *recall*, dan *F1-score*. Selain itu, *confusion matrix* digunakan untuk menganalisis distribusi prediksi tiap kelas. Dari sisi efisiensi, setiap model juga dievaluasi berdasarkan jumlah parameter, ukuran model, GFLOPs, dan waktu inferensi per citra. Perbandingan hasil antar varian digunakan untuk menganalisis *trade-off* antara akurasi dan efisiensi komputasi.

4. HASIL DAN PEMBAHASAN

Penelitian ini mengimplementasikan dan mengevaluasi empat varian model YOLOv11 (YOLOv11n, YOLOv11s, YOLOv11m, dan YOLOv11l) untuk klasifikasi *malware* berbasis citra. Setiap model dilatih menggunakan dataset yang terdiri dari 7.718 citra *training*, 1.654 citra *validation*, dan 1.655 citra *testing* dengan 8 kelas yang berbeda yaitu *adware*, *backdoor*, *benign*, *downloader*, *spyware*, *trojan*, *virus*, dan *worm*.

4.1. Analisis Performa Model

Evaluasi terhadap keempat varian YOLOv11 memperlihatkan akurasi yang sangat tinggi untuk klasifikasi *malware* berbasis citra, yaitu berkisar antara 97,8% hingga 98,1% seperti yang ditunjukkan pada Tabel 1. Model YOLOv11m mencatat akurasi tertinggi 98,1%, disusul YOLOv11n (97,9%), YOLOv11l (98%), dan YOLOv11s (97,8%). Hasil ini menunjukkan bahwa transformasi file *malware* menjadi representasi visual memberikan karakteristik yang dapat dipelajari dengan baik oleh model *deep learning*. Pendekatan *deep learning* telah terbukti menjadi teknik yang menjanjikan dalam berbagai metode deteksi *malware* [12].

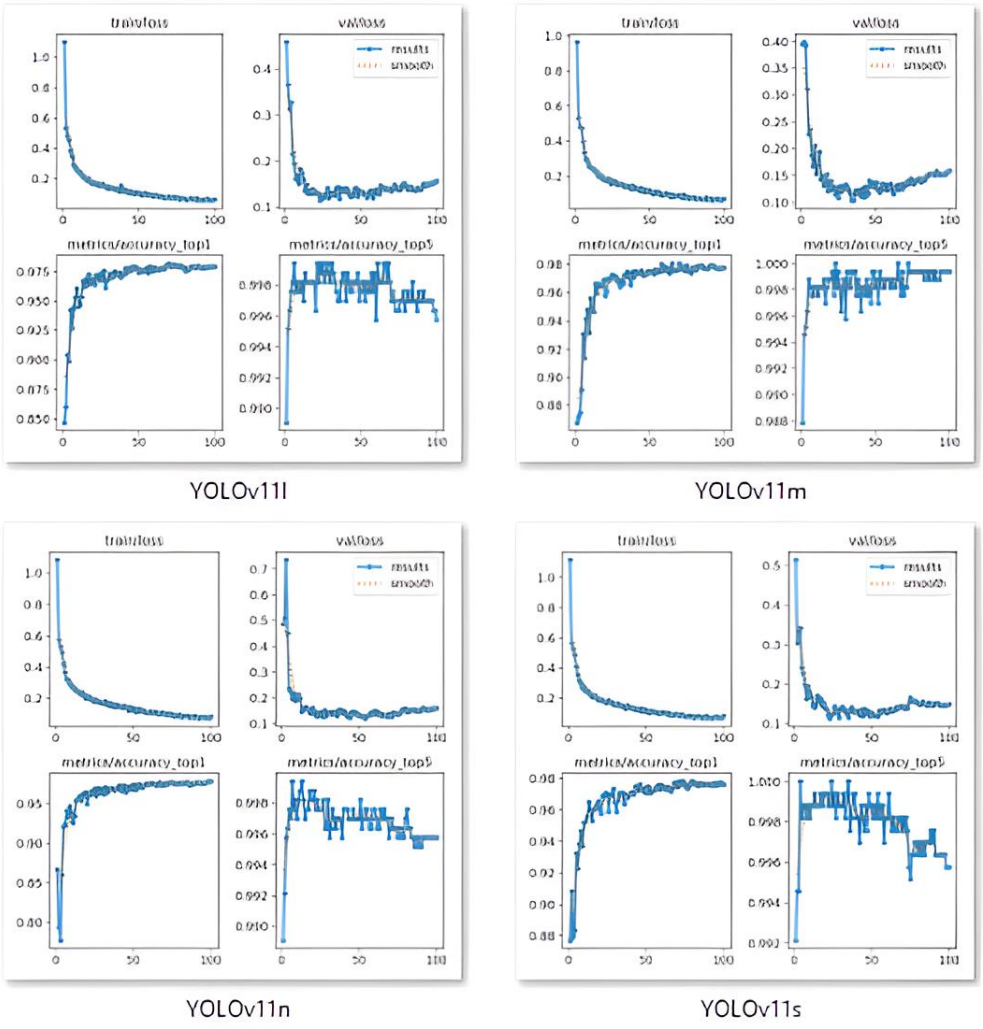
Tabel 1 Perbandingan Antar Model

Model	*Accuracy*	*Precision*	*Recall*	F1-*Score*	*Params*	GFLOPs	*Inference* (ms/img)
YOLOv11n	97.9%	0.9847	0.9843	0.9844	1.5M	3.2	0.3
YOLOv11s	97.8%	0.9852	0.9849	0.9849	5.4M	12.0	0.5
YOLOv11m	98.1%	0.9847	0.9843	0.9844	10.3M	39.3	0.8
YOLOv11l	98%	0.9852	0.9849	0.9849	12.8M	49.3	0.9

Untuk metrik *precision*, *recall*, dan *F1-score*, semua model menunjukkan konsistensi yang sangat baik dengan nilai di atas 0,984. YOLOv11s dan YOLOv11l mencapai *precision* dan *recall* tertinggi dengan nilai 0,9852 dan 0,9849, sementara YOLOv11n dan YOLOv11m mencatat nilai yang sedikit lebih rendah tetapi masih sangat kompetitif dengan nilai 0,9847 dan 0,9843. Konsistensi tinggi pada semua metrik evaluasi ini mengindikasikan bahwa model mampu mengenali pola visual yang unik dari setiap jenis *malware* dengan sangat baik, serta memiliki *robustness* tinggi terhadap variasi data [7].

4.2. Analisis Proses *Training*

Gambar 3 menunjukkan kurva *training* dan *validation loss* serta *accuracy* selama 100 *epoch* pelatihan untuk keempat varian YOLOv11. Semua model berhasil mencapai konvergensi dengan stabil, ditandai dengan penurunan *loss* yang konsisten hingga mendekati titik jenuh. Model YOLOv11n dan YOLOv11s konvergen lebih cepat dan mulai mencapai *plateau* pada *epoch* 60–70, sedangkan YOLOv11m dan YOLOv11l memerlukan waktu lebih lama tetapi menghasilkan *loss* akhir yang lebih rendah — menandakan kemampuan pembelajaran yang lebih dalam.



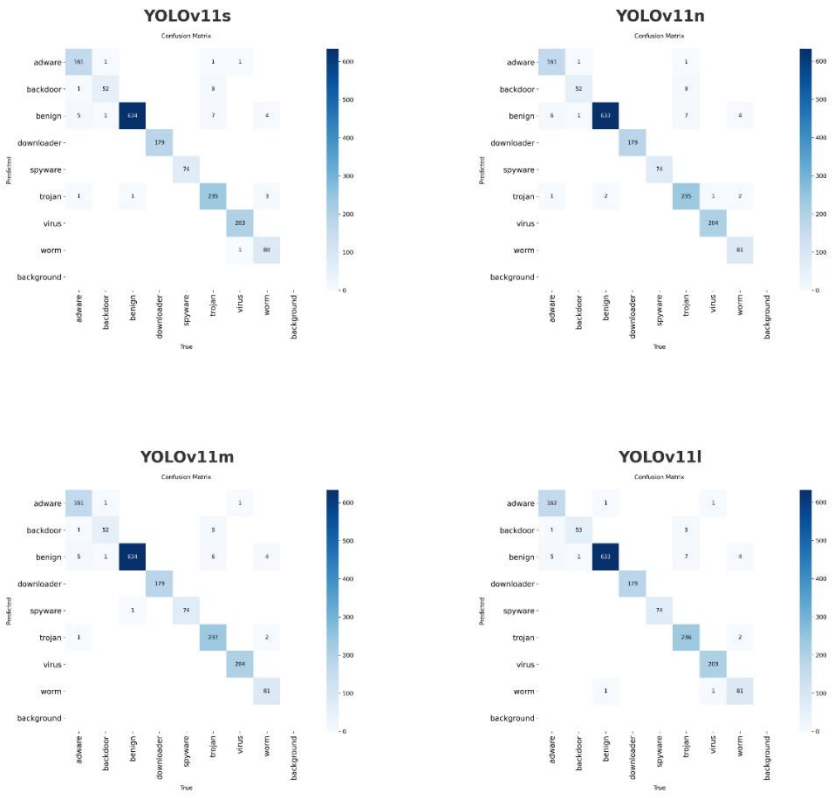
Gambar 3: Grafik hasil *training*

Seperti yang telah dijelaskan sebelumnya, dataset yang digunakan berasal dari *Malware Benign Image Classification Dataset* dengan total 22.056 citra *malware* dan *benign*. Dataset ini terdiri dari beberapa kategori *malware*, serta kategori *benign*. *File malware Windows Portable Executabl*e (PE) pada dataset ini tidak digunakan secara langsung dalam bentuk biner mentah. Dataset yang digunakan telah melalui proses konversi perilaku *malware* menjadi representasi citra berbasis API-*call* yang disediakan oleh sumber dataset. Setiap sampel *malware* dianalisis secara dinamis menggunakan *Cuckoo Sandbox* untuk mengekstraksi urutan pemanggilan API beserta argumennya selama eksekusi *runtime*.

Informasi API-*call* tersebut kemudian diproses menggunakan pendekatan *n-gram* untuk membentuk vektor fitur numerik yang merepresentasikan pola perilaku *malware*. Vektor fitur ini dinormalisasi ke dalam rentang nilai 0–255 dan dipetakan ke dalam bentuk matriks dua dimensi sehingga menghasilkan citra *grayscale* berukuran awal 128×128 piksel. Selanjutnya, seluruh citra dilakukan proses *resizing* menjadi 224×224 piksel menggunakan interpolasi *bilinear* tanpa penerapan *padding* atau *cropping* tambahan. Langkah ini dilakukan untuk menyesuaikan ukuran citra dengan kebutuhan input model YOLOv11.

4.3. Analisis *Confusion Matrix*

Berdasarkan *confusion matrix* yang ditunjukkan pada Gambar 4, semua model menunjukkan kemampuan klasifikasi yang sangat baik dengan tingkat kesalahan klasifikasi yang minimal pada seluruh kelas. Model-model ini menunjukkan kemampuan yang konsisten dalam membedakan antara kelas *malware* dan *benign*, serta mampu membedakan antar jenis malware yang berbeda dengan akurasi tinggi. Distribusi prediksi pada diagonal utama *confusion matrix* menunjukkan tingkat *true positive* yang sangat tinggi untuk semua kelas, dengan intensitas warna yang dominan pada diagonal mengindikasikan klasifikasi yang akurat.



Gambar 4: *Confusion matrix* tiap varian YOLOv11

Analisis lebih mendalam menunjukkan bahwa kesalahan klasifikasi yang muncul umumnya terjadi pada kelas yang memiliki kemiripan visual, seperti antara *virus* dan *worm*, atau antara *trojan* dan *backdoor*. Hal ini dapat dijelaskan karena beberapa jenis *malware* memiliki struktur biner dan pola visual yang mirip, kondisi tersebut menyebabkan tumpang tindih fitur spasial pada citra *malware*, sehingga model mengalami kesulitan dalam membedakan kedua kelas tersebut secara konsisten, khususnya pada sampel dengan karakteristik perilaku yang saling mendekati. Namun demikian, tingkat kesalahan tersebut tergolong sangat

kecil dan tidak berdampak signifikan pada performa keseluruhan. Selain itu, hampir tidak ditemukan kesalahan antara kelas *malware* dan *benign*, menandakan kemampuan model yang sangat baik dalam membedakan file berbahaya dari file sah (*legitimate*). Perbandingan performa antar versi YOLO menunjukkan konsistensi dalam kemampuan deteksi dengan trade-off yang jelas antara akurasi dan kecepatan [12].

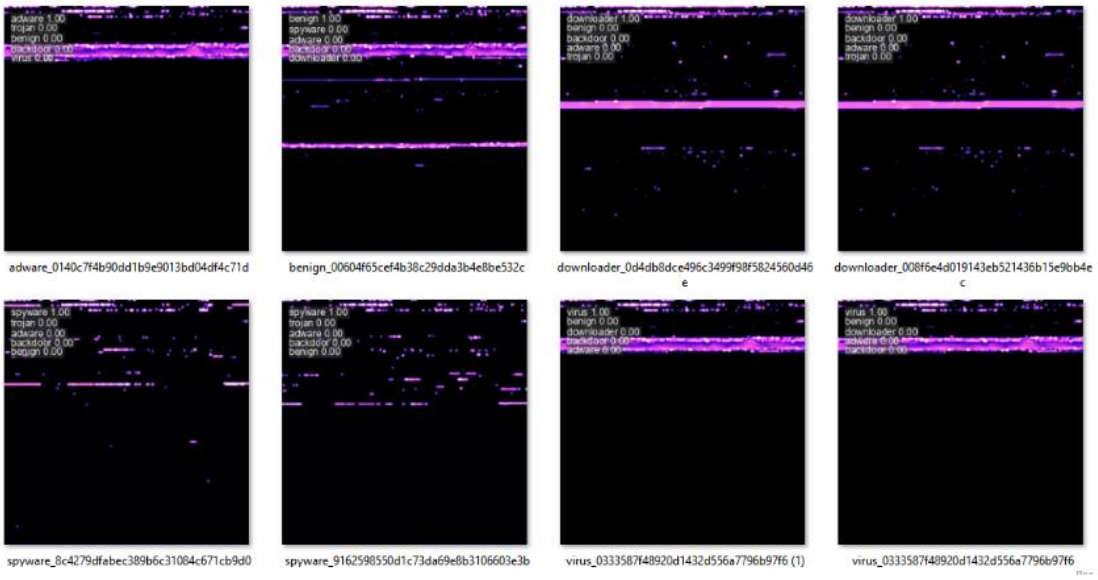
4.4. Efisiensi Komputasi dan *Trade-off*

Analisis efisiensi komputasi menunjukkan adanya *trade-off* yang jelas antara akurasi dan kompleksitas model. YOLOv11n sebagai model terkecil dengan 1,5M parameter dan 3,2 GFLOPs menunjukkan waktu inferensi tercepat yaitu 0,3 ms/img, namun tetap mempertahankan akurasi yang sangat tinggi sebesar 97,9%. Pendekatan alternatif seperti *Graph Neural Network* (GNN) juga menunjukkan potensi dalam klasifikasi *malware* berbasis *similarity* [13].

YOLOv11s dengan 5,4M parameter dan 12,0 GFLOPs memberikan keseimbangan yang optimal antara performa dan efisiensi dengan waktu inferensi 0,5 ms/img. Model ini menawarkan *compromise* yang baik untuk aplikasi yang membutuhkan akurasi tinggi namun tetap mempertimbangkan efisiensi komputasi. Sementara itu, YOLOv11m dan YOLOv11l meskipun memiliki parameter yang lebih besar yakni 10,3M dan 12,8M, hanya memberikan peningkatan akurasi yang marginal namun dengan peningkatan waktu inferensi yang signifikan menjadi 0,8 ms/img dan 0,9 ms/img. Hal ini menunjukkan bahwa untuk aplikasi praktis, model yang lebih kecil dapat memberikan hasil yang sebanding dengan efisiensi yang jauh lebih baik.

4.5. Evaluasi Model Pada Data *Testing*

Evaluasi visual pada data *testing* menunjukkan bahwa sebagian besar prediksi model memiliki *confidence score* di atas 0,95, yang mengindikasikan model memiliki tingkat keyakinan tinggi dalam memprediksi hasil proses klasifikasi. Hasil evaluasi pada data *testing* ditunjukkan pada Gambar 5. Pada konteks implementasi nyata, prediksi dengan *confidence score* di bawah ambang batas tertentu, misalnya 0,90, dapat ditandai sebagai prediksi berisiko dan diarahkan ke tahap analisis lanjutan menggunakan metode *dynamic analysis*. Pendekatan ini bertujuan untuk meningkatkan keandalan sistem deteksi *malware* dengan meminimalkan risiko kesalahan klasifikasi pada prediksi dengan tingkat keyakinan rendah. Selain itu, analisis terhadap sampel hasil prediksi menunjukkan bahwa model mampu mengklasifikasikan berbagai jenis *malware* dengan benar bahkan pada kasus-kasus yang secara visual memiliki pola yang kompleks atau *noise* yang tinggi.



Gambar 5: Hasil *testing* pada model YOLO

Pada kasus kesalahan prediksi, nilai *confidence* cenderung menurun (sekitar 0,85–0,92), menandakan adanya ambiguitas pada pola visual yang dianalisis. Hal ini dapat dimanfaatkan dengan menetapkan ambang batas (*threshold*) tertentu sebagai indikator prediksi yang memerlukan verifikasi manual atau analisis tambahan. Pada implementasi nyata, sistem dapat dikonfigurasi untuk memberikan *alert* atau meminta konfirmasi dari metode deteksi lain apabila *confidence score* di bawah nilai ambang. Secara keseluruhan, hasil *testing* visual ini memvalidasi performa kuantitatif yang telah diukur sebelumnya dan menunjukkan bahwa model tidak hanya akurat secara statistik tetapi juga *reliable* dalam aplikasi praktis dengan berbagai variasi input [8].

Hasil penelitian ini menunjukkan peningkatan yang signifikan dibandingkan dengan penelitian-penelitian sebelumnya dalam bidang klasifikasi *malware* berbasis citra. Nataraj dkk. yang menggunakan algoritma *k-Nearest Neighbor* berhasil mencapai akurasi 97,18% [4], sementara Venkatraman dkk. dengan pendekatan *hybrid CNN* mencapai akurasi 96,55% [8]. Penelitian ini berhasil melampaui kedua hasil tersebut dengan akurasi tertinggi 98,1% menggunakan YOLOv11m, menunjukkan superioritas arsitektur YOLOv11 dalam tugas klasifikasi *malware* berbasis citra. Keunggulan yang paling signifikan dari penelitian ini terletak pada efisiensi waktu inferensi yang sangat cepat, yaitu berkisar antara 0,3 hingga 0,9 ms/img, jauh lebih cepat dibandingkan metode CNN konvensional. Kombinasi antara akurasi tinggi dan efisiensi komputasi yang unggul, menjadikan sistem ini sangat potensial untuk diterapkan dalam deteksi *malware* otomatis berbasis *machine learning* pada lingkungan sistem nyata.

V. SIMPULAN DAN SARAN

Implementasi dan evaluasi empat varian YOLOv11 dalam penelitian ini memberikan hasil yang sangat baik dan memuaskan untuk klasifikasi *malware* berbasis citra. Keempat varian mencatat akurasi tinggi pada rentang 97,8% hingga 98,1%, di mana YOLOv11m meraih performa paling tinggi (98,1%), disusul YOLOv11n (97,9%), YOLOv11l (98%), dan YOLOv11s (97,8%). Nilai *precision*, *recall*, dan F1-*score* di atas 0,984 pada semua model membuktikan konsistensi dalam mengenali pola visual malware. Transformasi *malware* menjadi representasi visual terbukti efektif sebagai input untuk *deep learning*, melampaui penelitian sebelumnya yang menggunakan K-NN (97,18%) dan *hybrid* CNN (96,55%).

Penelitian ini mengidentifikasi *trade-off* jelas antara akurasi dan efisiensi komputasi. YOLOv11n menawarkan keseimbangan optimal dengan akurasi 97,9%, parameter 1,5M, dan waktu inferensi 0,3 ms/img, ideal untuk aplikasi *real-time* dan *edge computing*. YOLOv11m memberikan akurasi tertinggi namun dengan kompleksitas lebih besar (10,3M parameter, 0,8 ms/img inferensi), cocok untuk sistem yang mengutamakan presisi maksimal. Analisis *training* menunjukkan tidak ada *overfitting* signifikan, dan evaluasi visual mengonfirmasi *reliability* model dengan *confidence score* mayoritas di atas 0,95. Pendekatan YOLOv11 terbukti *superior* dalam kecepatan inferensi (0,3-0,9 ms/img) dibanding CNN konvensional (puluhan hingga ratusan ms/img), menjadikannya sangat cocok untuk deteksi *malware real-time* di lingkungan produksi.

Untuk penelitian selanjutnya, disarankan eksplorasi dataset yang lebih besar dan beragam untuk meningkatkan generalisasi model, evaluasi terhadap deteksi *zero-day malware* dapat ditingkatkan melalui penerapan pendekatan *anomaly detection* atau *open-set recognition, m*engingat model YOLO pada dasarnya bersifat *closed-set* dan hanya mampu mengenali kelas *malware* yang telah dilatih sebelumnya [15]. Pengembangan sistem *hybrid* yang mengombinasikan analisis visual dengan pendekatan *static* dan *dynamic analysis* juga berpotensi meningkatkan keandalan sistem dalam menghadapi variasi *malware* modern [16]. Dari sisi implementasi praktis, pengembangan API dan *integration framework* dapat mempermudah adopsi sistem dalam infrastruktur keamanan siber yang telah ada.

DAFTAR PUSTAKA

[1] J. Singh and K. K. Senapati, “Malware analysis and classification,” in *Malware Analysis and Intrusion Detection in Cyber-Physical Systems*, no. April, M. Kumar, Ed. IGI Global, 2023, pp. 42–63.

[2] P. Maniriho, A. N. Mahmood, and M. J. M. Chowdhury, “A Survey of Recent Advances in Deep Learning Models for Detecting Malware in Desktop and Mobile Platforms,” *ACM Comput. Surv.*, vol. 56, no. 6, pp. 145:1--145:41, Jan. 2024, doi: 10.1145/3638240.

[3] M. Egele, T. Scholte, E. Kirda, and C. Kruegel, "A survey on automated dynamic malware-analysis techniques and tools," *ACM Comput. Surv.*, vol. 44, no. 2, pp. 1–49, 2012, doi: 10.1145/2089125.2089126.

[4] L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, "Malware images: visualization and automatic classification," in *Proceedings of the 8th International Symposium on Visualization for Cyber Security*, Jul. 2011, pp. 1–7, doi: 10.1145/2016904.2016908.

[5] A. Damodaran, F. Di Troia, C. A. Visaggio, T. H. Austin, and M. Stamp, "A comparison of static, dynamic, and hybrid analysis for malware detection," *J. Comput. Virol. Hacking Tech.*, vol. 13, no. 1, pp. 1–12, 2017, doi: 10.1007/s11416-015-0261-z.

[6] A. Moser, C. Kruegel, and E. Kirda, "Limits of Static Analysis for Malware Detection," in *Twenty-Third Annual Computer Security Applications Conference (ACSAC 2007)*, 2007, pp. 421–430, doi: 10.1109/ACSAC.2007.21.

[7] T. M. Mohammed, L. Nataraj, S. Chikkagoudar, S. Chandrasekaran, and B. S. Manjunath, "Malware detection using frequency domain-based image visualization and deep learning," *Proc. Annu. Hawaii Int. Conf. Syst. Sci.*, pp. 7132–7141, 2021, doi: 10.24251/hicss.2021.858.

[8] S. Venkatraman, M. Alazab, and R. Vinayakumar, "A hybrid deep learning image-based analysis for effective malware detection," *J. Inf. Secur. Appl.*, vol. 47, pp. 377–389, 2019, doi: 10.1016/j.jisa.2019.06.006.

[9] G. Plastiras, C. Kyrkou, and T. Theocharides, "Efficient convnet-based object detection for unmanned aerial vehicles by selective tile processing," in *ACM International Conference Proceeding Series*, 2018, pp. 1–6, doi: 10.1145/3243394.3243692.

[10] R. Khanam and M. Hussain, "YOLOv11: An Overview of the Key Architectural Enhancements," pp. 1–9, 2024, doi: 10.48550/arXiv.2410.17725.

[11] B. P. Gond, "Malware Benign Image Classification Dataset," 2025. https://www.kaggle.com/datasets/bishwajitprasadgond/malware-benign-image-sample/data.

[12] M. Alshoulie and A. Mehmood, "Deep Learning Approaches for Malware Detection: A Comprehensive Review of Techniques, Challenges, and Future Directions," *IEEE Access*, vol. 13, pp. 118652–118677, 2025, doi: 10.1109/ACCESS.2025.3582875.

[13] Y.-S. Chen and Y.-X. Chen, "The Application and Performance Comparison of Different Versions of YOLO Image Recognition Systems," *Eng. Proc.*, vol. 98, no. 1, pp. 1–7, 2025, doi: 10.3390/engproc2025098001.

[14] Y. H. Chen, J. L. Chen, and R. F. Deng, "Similarity-Based Malware Classification Using Graph Neural Networks," *Appl. Sci.*, vol. 12, no. 21, 2022, doi: 10.3390/app122110837.

[15] M. Kim, H. Cho, and J. H. Yi, "Large-Scale Analysis on Anti-Analysis Techniques in Real-World Malware," *IEEE Access*, vol. 10, no. July, pp. 75802–75815, 2022, doi: 10.1109/ACCESS.2022.3190978.

[16] I. Ahmad, Z. Wan, A. Ahmad, and S. S. Ullah, "A Hybrid Optimization Model for Efficient Detection and Classification of Malware in the Internet of Things," *Mathematics*, vol. 12, no. 10, 2024, doi: 10.3390/math12101437.