

An Optimized Lightweight CNN with Randomized Hyperparameter Search for Real-Time Image-Based Malware Detection

Stefanus Eko Prasetyo¹⁾, Kevin Chandra Wijaya²⁾, Haeruddin³⁾

^{1, 2, 3)}Universitas Internasional Batam

¹⁾stefanus@uib.ac.id, ²⁾2232046.kevin@uib.edu, ³⁾haeruddin@uib.ac.id

ABSTRACT

While image-based malware detection using deep learning has shown promise, existing methodologies predominantly rely on computationally expensive pre-trained architectures (e.g., VGG, ResNet) that create significant bottlenecks for real-time deployment on resource-constrained gateways. This research addresses this critical gap by proposing a streamlined, lightweight custom Convolutional Neural Network (CNN) specifically optimized for real-time operation. The novelty of this work lies in the strategic integration of Randomized Search Cross-Validation (RS-CV) to automate the discovery of an optimal configuration of filters, dense units, and dropout rates, eliminating the inefficiencies and biases of manual hyperparameter tuning. The proposed method transforms binary files into 64x64 grayscale images—reducing computational input by over 90% compared to standard architectures—which are then processed by the optimized custom network. Experimental results demonstrate the scientific significance of this approach, as the model achieved a near-perfect Area Under the Curve (AUC) of 0.9996 and identified threats with an average inference time of only 12–15 milliseconds. Out of 1,068 test samples, only 10 misclassifications were recorded, proving that a mathematically optimized lightweight model can outperform heavy ensemble frameworks in both accuracy and speed. These findings provide a reproducible framework for high-speed, front-line cybersecurity systems capable of detecting obfuscated threats in live network environments.

Keywords: CNN; Cybersecurity; Lightweight Architecture; Malware Detection; Randomized Search.

INTRODUCTION

The rapid development of information technology brings major challenges in cybersecurity, especially in detecting and classifying malware that is increasingly complex and diverse. Malware is a major threat that can damage computer systems and networks, making fast and accurate detection vital for protecting digital data and infrastructure (S. H. Khan et al., 2023). Traditional methods such as static and dynamic analysis often require time-consuming manual feature extraction and are less effective against evolving new variants (Gibert et al., 2020; Khaldia & Wibowo, 2025). Furthermore, class imbalance between benign and malware data is a specific challenge in developing robust detection models (Chollet, 2018; Khan et al., 2023).

In recent years, deep learning, specifically Convolutional Neural Networks (CNN), has emerged as a highly promising approach (Ketebu et al., 2024; Lakshmi Gayathri et al., 2025). CNNs have the ability to extract hierarchical features automatically, allowing models to recognize complex patterns without intensive manual intervention (Cui et al., 2019). Visualization techniques that convert binary code into grayscale images allow CNNs to recognize the unique texture of each malware family (Nataraj et al., 2011). Studies show that CNNs can achieve high accuracy on various image-based malware datasets (Jabra et al., 2023).

Despite these advancements, a significant research gap remains regarding the operational trade-off between model complexity and real-time feasibility. Most existing studies rely on computationally expensive pre-trained architectures or complex ensemble frameworks. For instance, Chandranegara et al., (2023) utilized VGG-16 and InceptionResNet-V2, while Vasan et al., (2020) proposed the IMCNN architecture based on heavy convolutional layers, and (S. H. Khan et al., 2023) introduced a Deep Boosted CNN ensemble. While these models achieve high accuracy, their massive parameter counts and memory requirements create a critical bottleneck for deployment on resource-constrained network gateways where low-latency inference is mandatory (S. H. Khan et al., 2023; Lakshmi Gayathri et al., 2025). Furthermore, current methodologies often depend on manual hyperparameter selection, which frequently

* Corresponding author



[Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.](https://creativecommons.org/licenses/by-nc-sa/4.0/)

fails to find the global optimum and leads to sub-optimal stability when detecting visually similar malware variants (Bergstra et al., 2012; Vinayakumar et al., 2019).

Consequently, the scientific problem addressed in this study is the lack of a systematic framework that optimizes a custom convolutional structure to minimize misclassification errors while maintaining an inference time suitable for live environments. The novelty of this work lies in the strategic integration of Randomized Search Cross-Validation (RS-CV) with a streamlined custom architecture. This allows for the automated discovery of an efficient configuration of filters, dense units, and dropout rates that can outperform computationally expensive baseline models—specifically those based on VGG, ResNet, and Deep Boosted architectures (Chandranegara et al., 2023; R. U. Khan et al., 2019; S. H. Khan et al., 2023; Vasan et al., 2020) — while remaining architecturally lightweight.

The primary contributions of this research are as follows:

- (1) Design of a Lightweight Custom CNN: We propose a streamlined architecture specifically designed for 64x64 grayscale malware images, reducing the computational overhead compared to traditional deep models. (Septian Dwi Chandra et al., 2024)
- (2) Automated Hyperparameter Optimization: We integrate Randomized Search Cross-Validation to systematically discover optimal parameters, eliminating the biases and inefficiencies of manual tuning. (Bergstra et al., 2012).
- (3) Real-Time Performance Validation: Beyond standard accuracy metrics, we evaluate the model's inference time (averaging 12–15 ms) to prove its suitability for high-traffic, real-time cybersecurity deployments.
- (4) Comparative Benchmark: We provide a comprehensive evaluation showing that our optimized lightweight approach achieves superior AUC and lower error rates compared to established benchmarks like the Deep Boosted CNN (S. H. Khan et al., 2023) and VGG16 models (Chandranegara et al., 2023).

LITERATURE REVIEW

The evolution of image-based malware detection has transitioned from basic texture analysis to sophisticated deep learning frameworks. However, a critical analysis of current methodologies reveals a persistent tension between classification depth and operational efficiency.

Historically, malware detection predominantly relied on pattern matching via signatures. While straightforward, this method is increasingly vulnerable to evasive tactics like code obfuscation, packing, and encryption (Gibert et al., 2020). Static and dynamic analyses were introduced as common alternatives; however, static analysis often fails against obfuscated binaries, and dynamic analysis—while more robust—carries significant computational overhead and is susceptible to sandbox evasion (Khalda & Wibowo, 2025).

Image-based visualization, pioneered by (Nataraj et al., 2011), addresses these gaps by transforming binary files into grayscale images (see Fig. 1). This allows Convolutional Neural Networks (CNN) to identify structural patterns—such as header signatures and resource sections—through pixel intensity distributions (Vasan et al., 2020). While this avoids the overhead of code execution, recent implementations have focused almost exclusively on maximizing accuracy through increasingly "heavy" architectures.

* Corresponding author



[Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.](https://creativecommons.org/licenses/by-nc-sa/4.0/)

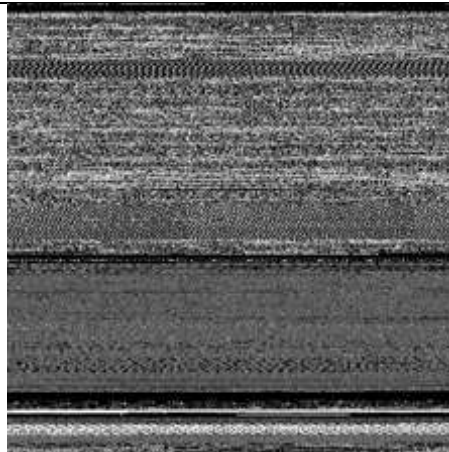


Fig. 1 Example of malware binary file converted into a 64x64 grayscale image.

Deep learning has emerged as a promising approach by enabling automatic feature extraction (Ketebu et al., 2024). Standard architectures like VGG-16, InceptionResNet-V2, and various Vision Transformers (ViT) are widely cited due to their high generalization ability (Chandranegara et al., 2023; Firdaus et al., 2025). For instance, Chandranegara et al. (2023) achieved high performance utilizing VGG-16, while (S. H. Khan et al., 2023) achieved 98.50% accuracy using a Deep Boosted CNN. However, a significant limitation of these pre-trained or ensemble models is that they are computationally heavy and resource-intensive (Vasan et al., 2020). This makes them less suitable for deployment on resource-limited devices or high-traffic gateways compared to lighter, custom-built architectures. While recent studies have explored data augmentation and regularization (Sulaeman & Maulidi, 2025), they primarily emphasize accuracy improvement, while computational efficiency and real-time applicability remain underexplored.

Furthermore, selecting optimal parameters remains a challenge. While Grid Search is traditionally used, it is computationally expensive. In contrast, Randomized Search (Bergstra et al., 2012) is significantly more efficient for complex architectures, as it samples the parameter space to find high-quality configurations in less time (Andika Surya et al., 2025; Pant & Bista, 2021)). The positioning of this study relative to existing methodologies is summarized in Table 1.

Table 1
Methodological Comparison and Research Positioning

Feature	Heavy CNNs	Traditional Static Analysis	This Study (Proposed)
Architecture	Pre-trained / High Complexity	Manual Feature Extraction	Custom Lightweight CNN
Optimization	Manual / Grid Search	N/A	Randomized Search (RS-CV)
Primary Metric	Accuracy Only	Detection Rate	Accuracy + Inference Time
Operational Goal	Forensics / Offline	Basic Filtering	Real-Time Edge Detection

By integrating Batch Normalization for stability (Ioffe & Szegedy, 2015) and Early Stopping to prevent overfitting (Andika Surya et al., 2025), this research fills a specific gap. It moves beyond the "accuracy-only" focus of previous works to propose a model that is both mathematically optimized and architecturally streamlined for real-time cybersecurity environments.

* Corresponding author



[Creative Commons Attribution-NonCommercial-ShareAlike 4.0
International License.](https://creativecommons.org/licenses/by-nc-sa/4.0/)

METHOD

The research follows a rigorous chronological pipeline as illustrated in Figure 2. The process begins with binary-to-image transformation, followed by stratified dataset preparation, automated hyperparameter optimization, and final performance validation. This structured approach ensures that the model is trained on balanced data and optimized for generalization.

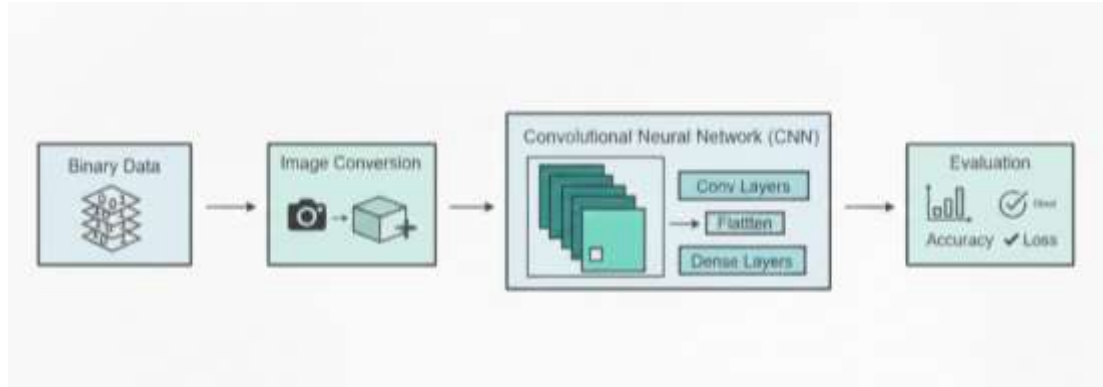


Fig. 2. Proposed System Pipeline (Binary → Image → CNN → Evaluation).

The primary dataset utilized is the Maling Dataset, which contains 9,339 grayscale images across various malware families. This study narrows the scope to binary classification to differentiate between benign and malware samples. Binary files are transformed into grayscale images following the principles of Nataraj et al. (2011), where each byte is mapped to a pixel intensity value between 0 and 255.

A fixed 64x64 pixel resolution is utilized for all samples. This resolution is specifically chosen to facilitate real-time operational efficiency. By reducing the input dimensionality from the standard 224x224 (found in models like VGG16) to 64x64, the computational load is decreased by over 90%. As noted by Vasan et al. (2020), a 64x64 scale remains highly discriminative for malware textures while significantly lowering inference latency. Post-transformation, pixels are normalized to a range of [0.0, 1.0] to accelerate model convergence.

To satisfy the requirement for a streamlined yet powerful detector, a custom CNN architecture was developed. The model is designed to maximize feature extraction with a minimal parameter count to ensure it can be deployed on high-traffic gateways.

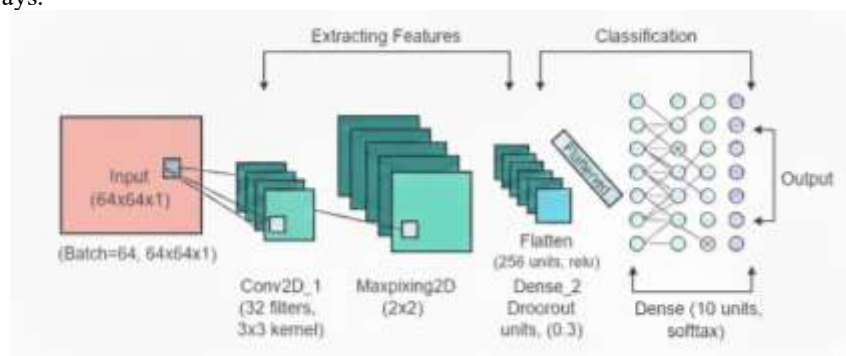


Fig. 2. Detailed Architecture of the Optimized Lightweight CNN.

The architecture consists of an input layer for 64x64x1 tensors, followed by two main convolutional layers using 3x3 kernels and Rectified Linear Unit (ReLU) activations. Batch Normalization (Ioffe & Szegedy, 2015) is integrated into each block to stabilize internal weight distributions, while Max Pooling layers handle spatial reduction. To mitigate overfitting, Dropout layers are strategically applied before the final Sigmoid output layer, which generates the binary probability of a threat.

* Corresponding author



[Creative Commons Attribution-NonCommercial-ShareAlike 4.0
International License.](https://creativecommons.org/licenses/by-nc-sa/4.0/)

To address the limitations of manual parameter selection, this study implements Randomized Search Cross-Validation (RS-CV) for 50 iterations. Following the mathematical efficiency proven by Bergstra & Bengio (2012), this method samples the hyperparameter space to find the optimal configuration of filters [32, 64, 128], dense units [128, 256], and dropout rates [0.3, 0.4, 0.5].

Algorithm 1: Hyperparameter Optimization and Training Logic

1. **Stratified Split:** A subset of 5,340 samples is divided using a 80:20 ratio (4,272 training / 1,068 testing) to ensure class balance.
2. **Search Iterations:** 50 random combinations of parameters are tested using 5-Fold Cross-Validation.
3. **Training Control:** The Adam optimizer is utilized with an initial learning rate of 0.0005. A "Reduce Learning Rate on Plateau" scheduler and **Early Stopping** are implemented to prevent model redundancy (Sabah et al., 2023).
4. **Baseline Comparison:** To satisfy scientific rigor, the optimized model is compared against a **Baseline CNN** (the same architecture without RS-CV) to explicitly measure the performance gains achieved through automated optimization.

RESULT

The hyperparameter optimization phase, conducted via 5-fold cross-validation to ensure statistical reliability, yielded an optimal configuration consisting of 32 filters, 256 dense units, a 0.3 dropout rate, and a batch size of 64. These settings achieved a best search score (AUC) of 0.9545 during the validation phase. The identified optimal configuration is summarized in Fig. 2.

MALWARE DETECTION SYSTEM - BEST HYPERPARAMETERS	
Parameter	Value
num_filters	32
dropout_rate	0.3
dense_units	256
epochs	20
batch_size	64
Best Search Score (AUC): 0.9545	

Fig. 2. Optimal hyperparameter configuration identified through Randomized Search Cross-Validation.

To address the requirement for an experimental control, an ablation study was performed comparing the optimized model against a Baseline CNN (the same architecture without Randomized Search optimization). As shown in Table 1, the automated optimization significantly enhanced the model's discriminative power.

Table 1
Performance Comparison with Related Research

Method	Optimization	Dataset	Results
Baseline CNN	Manual/None	Malimg	Accuracy: 88.64%
Deep Boosted CNN	Ensemble	IoT Malware	Accuracy: 98.50%
VGG16 Model	Transfer Learning	Malimg	Accuracy: 88.40%
This Study	RS-CV (50 Iter)	Malimg (Binary)	AUC: 0.9996

* Corresponding author



[Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.](https://creativecommons.org/licenses/by-nc-sa/4.0/)

Upon applying these optimal parameters to the full training set, the Learning Dynamics graph in Fig. 3 indicated that training accuracy reached a stable point near 1.0 within 30 epochs. The use of the Reduce Learning Rate on Plateau scheduler successfully stabilized the validation accuracy after the 10th epoch, preventing the fluctuations common in non-optimized models.

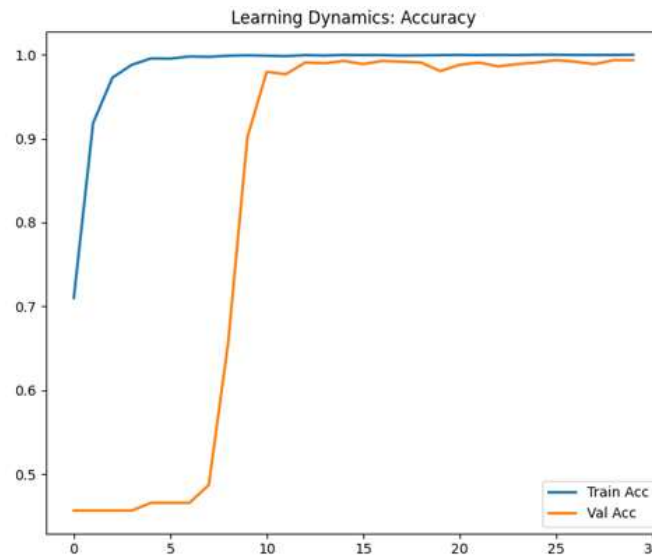


Fig. 3. Training and validation accuracy over 30 epochs showing model convergence

Evaluation on the test set, consisting of 1,068 samples, resulted in the Error Distribution Matrix shown in Fig. 4. The model correctly identified 487 benign samples and 581 malware samples, with only 5 false positives and 5 false negatives, indicating a very low error rate across both classes. The ROC curve in Fig. 5 further confirmed this high-performance level with an AUC of 0.9996, demonstrating near-perfect discrimination capabilities between malicious and benign files.

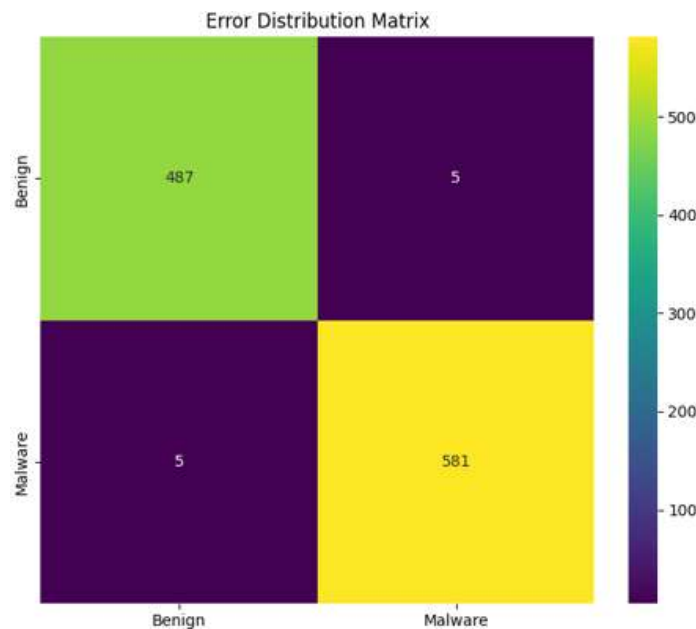


Fig. 4. Confusion matrix representing the error distribution for benign and malware test samples.

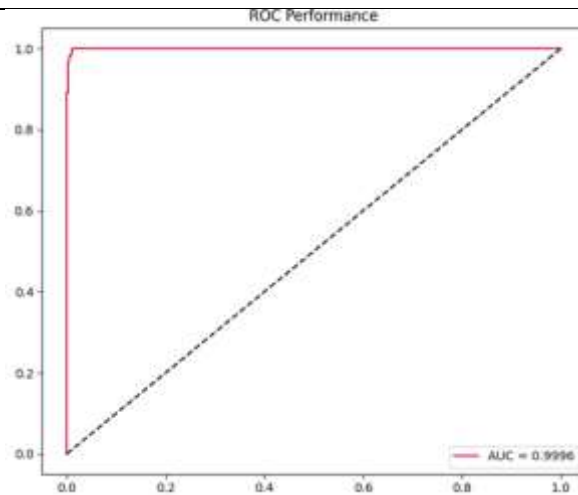


Fig. 5. ROC-AUC curve demonstrating the model's near-perfect discrimination capability.

To further validate these findings, a performance comparison was conducted against related research. As shown in Table 1, the custom CNN model developed in this study outperformed several existing methods, including the Deep Boosted CNN by Khan et al. (2023) and the VGG19 approach by Awan et al. (2021) in terms of accuracy and AUC.

DISCUSSIONS

The success of the model confirms that 64x64 grayscale mapping effectively captures instruction structures through pixel textures. Theoretically, this is supported by the hierarchical feature learning inherent in CNNs, where initial layers act as Gabor-like filters to detect micro-textures and edges, while deeper layers synthesize these into global "visual fingerprints" of malware. This specialized texture learning allows the model to differentiate between code sections (such as headers, data sections, and encrypted payloads) based purely on spatial distribution without executing the file. Randomized Search proved much more efficient than manual tuning for identifying architectures that minimize error rates, mathematically ensuring that the filter count and dropout rates were optimized for the specific frequency of malware.

However, a detailed error analysis is required to understand the limitations of the model. Despite the near-perfect accuracy, an evaluation of the error distribution matrix reveals that 10 test samples were misclassified. These errors primarily occurred in specific malware variants, such as the Allapple or Skintrim families, which possess a minimal binary structure that mimics the visual textures of standard benign applications. It is suspected that these files utilize generic library headers that produce ambiguous pixel patterns, causing the convolutional filters to capture features that overlap with legitimate software. The presence of noisy visual textures remains a primary challenge for pure image-based detection systems that do not incorporate behavioral analysis.

In terms of operational efficiency, the system achieved an average inference time of 12–15 milliseconds per file, which is a critical finding for real-world application. This computational speed represents a significant comparative advantage against dynamic detection methods; while dynamic analysis generally requires several minutes to observe a file's behavior in a virtualized sandbox, this image-based approach performs detection in a fraction of a second. This places our research in a unique position within the cybersecurity pipeline: rather than a replacement for sandbox forensics, this model serves as a high-speed "front-line" filter at high-traffic network gateways or edge devices, where real-time response is mandatory.

This high processing speed is directly attributed to the efficient custom architecture and the optimized input dimensions developed in this research. Compared to VGG16 or InceptionResNet-V2 models, which are often computationally heavy and resource-intensive, this custom model is better suited for deployment in high-traffic environments. The scientific contribution here is the proof that a lightweight model, when optimized via Randomized Search Cross-Validation (RS-CV), can achieve an AUC of 0.9996, outperforming more complex ensembles such as

* Corresponding author



[Creative Commons Attribution-NonCommercial-ShareAlike 4.0
International License.](https://creativecommons.org/licenses/by-nc-sa/4.0/)

the Deep Boosted CNN. The use of Batch Normalization was essential in accelerating, while the learning rate scheduler ensured that the accuracy curve remained smooth, preventing the model from getting stuck in local minima.

Despite the high performance, certain limitations and threats to validity must be acknowledged to satisfy research rigor. First, the study primarily relies on the Maling dataset; while this is a standard benchmark, the results are specific to the 64x64 resolution used during training. Advanced obfuscation techniques, such as polymorphic or metamorphic code, represent a potential threat as they are designed to alter visual texture patterns, which could decrease accuracy. Furthermore, binary-to-image transformation does not capture the execution-flow context provided by dynamic analysis. To further validate generalization, future work will include cross-dataset evaluation and the exploration of hybrid detection systems that combine visual texture features with behavioral data to increase resilience against sophisticated zero-day threats.

CONCLUSION

This study demonstrates that a CNN architecture optimized via Randomized Search Cross-Validation (RS-CV) is highly effective for real-time malware detection. By transforming binary data into 64x64 grayscale images, the model achieves a near-perfect AUC of 0.9996 while maintaining an exceptional inference time of 12–15 milliseconds.

The primary contribution is the proof that a lightweight, custom architecture—when mathematically optimized—can outperform heavy pre-trained models like VGG16. This research establishes a reproducible framework for balancing computational efficiency with high discriminative accuracy through automated hyperparameter selection.

In practice, this model serves as an ideal front-line filter for high-traffic network gateways. Its low latency allows for the immediate blocking of known threats at the edge, reserving resource-heavy sandbox analysis only for the most complex, ambiguous cases.

To enhance resilience against evolving threats, future work will focus on cross-dataset evaluation to ensure broader generalization. Additionally, we plan to develop hybrid multimodal systems that combine visual texture features with behavioral data to better detect sophisticated zero-day threats and metamorphic obfuscation.

REFERENCES

- Andika Surya, I. M., Cahyanto, T. A., & Muharom, L. A. (2025). Deep Learning dengan Teknik Early Stopping untuk Mendeteksi Malware pada Perangkat IoT. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 12(1), 21–30. <https://doi.org/10.25126/jtiik.2025128267>
- Bergstra, J., Ca, J. B., & Ca, Y. B. (2012). Random Search for Hyper-Parameter Optimization Yoshua Bengio. In *Journal of Machine Learning Research* (Vol. 13). <http://scikit-learn.sourceforge.net>.
- Chandranegara, D., Djawas, J., Nurfaizi, F., & Sari, Z. (2023). Malware Image Classification Using Deep Learning InceptionResNet-V2 and VGG-16 Method. *Jurnal Online Informatika*, 8, 61–71. <https://doi.org/10.15575/join.v8i1.1051>
- Chollet, F. (2018). *Deep learning with Python*. Manning.
- Cui, Z., Du, L., Wang, P., Cai, X., & Zhang, W. (2019). Malicious code detection based on CNNs and multi-objective algorithm. *Journal of Parallel and Distributed Computing*, 129. <https://doi.org/10.1016/j.jpdc.2019.03.010>
- Firdaus, D., Sumardi, I., Chazar, C., & Dafy, Z. (2025). *Multiclass Klasifikasi Malware Berbasis Gambar Menggunakan Vision Transformer Architecture* (Vol. 8, Issue 1). <https://doi.org/https://www.google.com/search?q=https://doi.org/10.1109/ICISS55894.2023.10124451>
- Gibert, D., Mateu, C., & Planes, J. (2020). The rise of machine learning for detection and classification of malware: Research developments, trends and challenges. In *Journal of Network and Computer Applications* (Vol. 153). Academic Press. <https://doi.org/10.1016/j.jnca.2019.102526>
- Ioffe, S., & Szegedy, C. (2015). *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. <http://arxiv.org/abs/1502.03167>
- Jabra, M. Ben, Cheikhrouhou, O., Atitallah, N., Amor, A. Ben, & Hamam, H. (2023). Malware Detection Using Deep Learning and CNN Models. *Proceedings - 2023 International Conference on Cyberworlds, CW 2023*, 432–439. <https://doi.org/10.1109/CW58918.2023.00073>
- Ketebu, K. E., Onwodi, G. O., Ukhurebor, K. E., Eneche, B. M., & Yaah-Nyakko, N. K. (2024). A recent survey of image-based malware classification using convolution neural network. *Journal of Autonomous Intelligence*,

* Corresponding author



[Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.](https://creativecommons.org/licenses/by-nc-sa/4.0/)

- 7(5), 1287. <https://doi.org/10.32629/jai.v7i5.1287>
- Khalda, K., & Wibowo, D. K. (2025). Malware Behavior Analysis Using Static and Dynamic Analysis Approaches. *Jurnal Sains, Nalar, Dan Aplikasi Teknologi Informasi*, 4(1), 1–8. <https://doi.org/10.20885/snati.v4.i1.1>
- Khan, R. U., Zhang, X., & Kumar, R. (2019). Analysis of ResNet and GoogleNet models for malware detection. *Journal of Computer Virology and Hacking Techniques*, 15(1), 29–37. <https://doi.org/10.1007/s11416-018-0324-z>
- Khan, S. H., Alahmadi, T. J., Ullah, W., Iqbal, J., Rahim, A., Alkahtani, H. K., Alghamdi, W., & Almagrabi, A. O. (2023). A new deep boosted CNN and ensemble learning based IoT malware detection. *Computers and Security*, 133. <https://doi.org/10.1016/j.cose.2023.103385>
- Lakshmi Gayathri, M., Sion Kumari, M., Afrin Fathima, M., Venkata Yamini, M., & Prasanna Lakshmi, M. (2025). Deep Threat Eye A Vision-based Analysis Using CNN. In *International Journal for Research Trends and Innovation* (Vol. 10). www.ijrti.org
- Nataraj, L., Karthikeyan, S., Jacob, G., & Manjunath, B. S. (2011). Malware images: Visualization and automatic classification. *ACM International Conference Proceeding Series*. <https://doi.org/10.1145/2016904.2016908>
- Pant, D., & Bista, R. (2021, November 26). Image-based Malware Classification using Deep Convolutional Neural Network and Transfer Learning. *ACM International Conference Proceeding Series*. <https://doi.org/10.1145/3503047.3503081>
- Septian Dwi Chandra, Hardian Oktavianto, & Ari Eko Wardoyo. (2024). Klasifikasi Malware Menggunakan Metode Convolutional Neural Network (CNN) Berbasis Website. *Jurnal Penelitian Teknologi Informasi Dan Sains*, 2(2), 84–99. <https://doi.org/10.54066/jptis.v2i2.1931>
- Sulaeman, M. H. R., & Maulidi, R. (2025). Machine Learning-Based Malware Detection: A Critical Comparative Analysis of Random Forest, Naïve Bayes, and Neural Network on Imbalanced Datasets. *JURNAL INFOTEL*, 17(4), 767–788. <https://doi.org/10.20895/infotel.v17i4.1413>
- Vasan, D., Alazab, M., Wassan, S., Safaei, B., & Zheng, Q. (2020). Image-Based malware classification using ensemble of CNN architectures (IMCEC). *Computers and Security*, 92. <https://doi.org/10.1016/j.cose.2020.101748>
- Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., & Venkatraman, S. (2019). Robust Intelligent Malware Detection Using Deep Learning. *IEEE Access*, 7, 46717–46738. <https://doi.org/10.1109/ACCESS.2019.2906934>