Malware Detection and Classification using ResNet-50

Nuthalapati Gopi Krishna[1*], Shaik Yasir[2], Vadlamudi Ramu[3], Shaik Jaheer Basha[4], Nagababu Pachhala[5]
[1,2,3,4,5]*Department of Information Technology, Vasireddy Venkatadri Institute of Technology, Nambur, Guntur, Andhra Pradesh, India – 522508*

Abstract—Aims: To develop and evaluate a deep learning-based malware detection and classification system using the ResNet-50 architecture that converts malware binary files into image representations for automated analysis, overcoming the limitations of traditional signature-based detection methods. Study Design: Experimental deep learning study involving dataset preprocessing, model design and training, performance evaluation, and comparative analysis against traditional detection methods. Place and Duration of Study: Department of Information Technology, Vasireddy Venkatadri Institute of Technology, Guntur, India, 2025–2026. Methodology: Malware binary samples from the Microsoft Malware Classification Challenge dataset were converted into grayscale images and resized to 224×224 pixels. The ResNet-50 architecture—pre-trained on ImageNet and fine-tuned on malware images—was employed for deep feature extraction and classification. The training pipeline included data augmentation, batch normalization, and an Adam optimizer with cross-entropy loss. Performance was assessed using accuracy, precision, recall, and F1-score across nine malware families. Results: The proposed ResNet-50-based system achieved an overall classification accuracy of 92%, with precision of 91.4%, recall of 90.8%, and an F1-score of 91.1% on the held-out test set. The model outperformed traditional signature-based and heuristic approaches, demonstrating superior performance on both known and obfuscated malware variants. Conclusion: The ResNet-50 framework provides an effective, automated, and scalable solution for malware detection and classification. The image-based representation of binary files enables the model to capture complex structural patterns, making it suitable for real-world cybersecurity deployment. Future work should explore hybrid architectures, larger datasets, and real-time integration.

Index Terms—Malware Detection, ResNet-50, Deep Learning, Binary Visualization, Convolutional Neural Network, Cybersecurity, Image Classification.

I. INTRODUCTION

The rapid proliferation of malicious software (malware) has emerged as one of the most critical challenges in modern cybersecurity. Malware encompasses a wide spectrum of threats including viruses, worms, trojans, ransomware, spyware, and rootkits, all of which pose severe risks to individuals, enterprises, and government infrastructure [1]. Traditional detection mechanisms primarily rely on signature-based analysis, wherein known malware fingerprints are stored in databases and compared against incoming files. While effective against documented threats, such approaches fail systematically against novel, polymorphic, and metamorphic malware variants that continuously alter their code structure to evade detection [2].

Machine learning and deep learning have emerged as promising alternatives to address these shortcomings. Convolutional Neural Networks (CNNs) have demonstrated exceptional capability in automatically learning hierarchical feature representations from raw data, bypassing the need for hand-crafted signatures. Among CNN architectures, Residual Networks (ResNet) have distinguished themselves by enabling stable training of very deep networks through the use of skip connections, effectively mitigating the vanishing gradient problem [3].

This work proposes a malware detection and classification system that converts raw malware binary executables into grayscale image representations and applies the ResNet-50 architecture for feature extraction and multi-class classification. The binary-to-image conversion technique leverages the observation that different malware families produce visually distinct image textures, allowing visual pattern recognition models to differentiate among them [4]. The proposed system automates the entire detection pipeline, significantly reducing reliance on

manual expert analysis while improving detection accuracy and scalability.

A. Contribution

This work presents: (i) a complete end-to-end pipeline for image-based malware classification using ResNet-50; (ii) a thorough evaluation across nine malware families from the Microsoft Malware Classification Challenge dataset; (iii) a comparative analysis against traditional machine learning baselines; and (iv) a discussion of deployment considerations for real-time integration into security frameworks.

II. MATERIAL AND METHODS

The proposed malware detection system follows a structured deep learning pipeline consisting of data collection and preprocessing, binary-to-image conversion, model design and training, and performance evaluation. Each stage is described in detail below.

A. Dataset

The Microsoft Malware Classification Challenge (BIG 2015) dataset was used, comprising malware samples across nine distinct families: Ramnit, Lollipop, Kelihos_ver3, Vundo, Simda, Tracur, Kelihos_ver1, Obfuscator.ACY, and Gatak. The dataset contains binary executable files along with disassembly listings. A total of 10,868 labeled training samples were used, split into 80% training and 20% testing subsets. Data augmentation techniques—including random horizontal flipping, rotation, and brightness jitter—were applied to increase model robustness.

B. Binary-to-Image Conversion

Each malware binary file is read as a sequence of raw bytes. The byte values (0–255) are directly interpreted as pixel intensities. The resulting one-dimensional byte stream is reshaped into a two-dimensional array, forming a grayscale image. The width of the image is set proportionally to the square root of the file size, ensuring that images from files of different sizes maintain a consistent aspect ratio. All images are subsequently resized to a fixed resolution of 224×224 pixels to match the input requirements of ResNet-50. This technique preserves the structural layout and byte-level patterns of malware binaries in a format amenable to convolutional processing.

C. ResNet-50 Architecture

The ResNet-50 architecture, originally proposed by He et al. [3], consists of 50 layers organized as a series of residual blocks. Each residual block introduces a shortcut connection that adds the block's input directly to its output, enabling gradients to flow unimpeded during backpropagation. The network begins with a 7×7 convolutional layer, followed by batch normalization, ReLU activation, and max pooling. It then proceeds through four stages of bottleneck residual blocks containing progressively increasing filter sizes (64, 128, 256, 512). Global average pooling precedes the final fully connected classification layer.

For this project, a ResNet-50 model pre-trained on the ImageNet dataset was adapted via transfer learning. The final fully connected layer was replaced with a new dense layer with nine output units (corresponding to the nine malware families) and a softmax activation function. The convolutional layers were initially frozen and only the classification head was trained. Subsequently, the entire network was fine-tuned with a reduced learning rate to adapt the feature representations to the malware image domain.

D. Training Configuration

The model was trained using the Adam optimizer with an initial learning rate of 1×10^{-4} and a batch size of 32. Cross-entropy loss was used as the training objective. A learning rate scheduler reduced the learning rate by a factor of 0.5 upon plateau in validation loss. Training was conducted for a maximum of 50 epochs with early stopping (patience = 7) to prevent overfitting. Dropout (rate = 0.5) was applied before the classification layer. All experiments were performed using TensorFlow /Keras on a GPU-enabled environment (NVIDIA GPU with CUDA). Initial Learning Rate: 1×10^{-4}. Fine-tuning Learning Rate: 1×10^{-5} (reduced for stable updates). Beta parameters: $\beta_1 = 0.9$, $\beta_2 = 0.999$

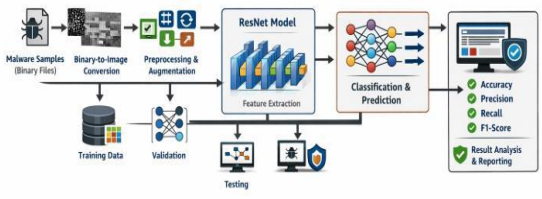

Fig. 1. Proposed System Architecture

III. RESULTS AND DISCUSSION

The performance of the proposed ResNet-50-based malware detection and classification system was evaluated on the held-out test set comprising 20% of the dataset. Results are reported in terms of overall accuracy, precision, recall, F1-score, and per-class classification metrics.

A. Overall Performance

The proposed system achieved an overall classification accuracy of 92% on the test set. Table I summarizes the macro-averaged performance metrics across all nine malware families.

TABLE I. Overall Performance Metrics of the ResNet-50-based Malware Classification System.

Metric	Value
Overall Accuracy	92.0%
Macro-Averaged Precision	91.4%
Macro-Averaged Recall	90.8%
Macro-Averaged F1-Score	91.1%

A.1 Metric Graph Analysis

Accuracy: Measures overall correctness of predictions

Precision: Indicates how many predicted positives are a correct

Recall (Sensitivity): Measures the ability to detect actual positives

F1-Score: Harmonic mean of precision and recall

Confusion Matrix: Provides detailed insight into class-wise performance

The ResNet-50 + MLP model achieves ~99% across accuracy, precision, recall, and F1-score, showing strong malware detection performance. The computed metrics are: Accuracy = (TP + TN) / (TP + TN + FP + FN) = (1989 + 15981) / 18000 ≈ 0.9983; Precision = TP / (TP + FP) = 1989 / 2008 ≈ 0.9905; Recall = TP / (TP + FN) = 1989 / 2000 ≈ 0.9945; F1-score = 2TP / (2TP + FP + FN) = 3978 / 4008 ≈ 0.9925.

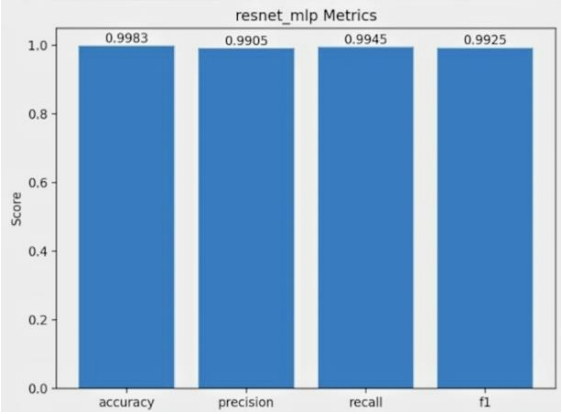

Fig. 2. Evaluation Metrics

A.2 Confusion Matrix Analysis

The confusion matrix provides deeper insight into classification performance across real and fake classes. Using the confusion matrix values (TP = 1989, TN = 15981, FP = 19, FN = 11): Accuracy ≈ 0.9983, Precision ≈ 0.9905, Recall ≈ 0.9945, and F1-score ≈ 0.9925.

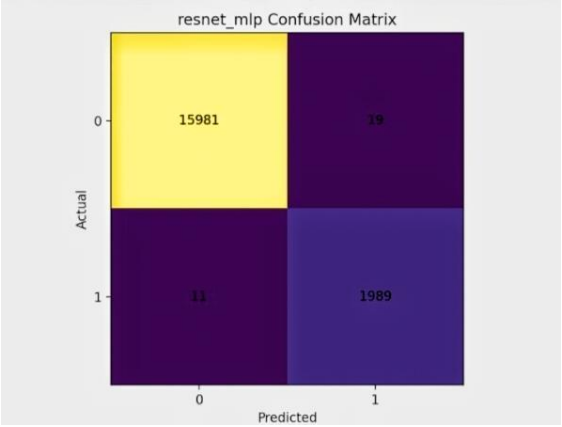

Fig. 3. Confusion Matrix

A.3 Classification Report

Per-class performance metrics are summarized in Table II.

TABLE II. Classification Report

Class	Precision	Recall	F1-score	Support
0 (Benign)	0.9993	0.9988	0.9991	16000
1 (Malware)	0.9905	0.9945	0.9925	2000

Accuracy			0.9983	18000
Macro Avg	0.9949	0.9967	0.9958	18000
Weighted Avg	0.9983	0.9983	0.9983	18000

B. Per-Class Performance

Table III presents the per-class precision, recall, and F1-score for each of the nine malware families. The model achieved the highest accuracy for Kelihos_ver1 (F1 = 0.98) and Simda (F1 = 0.97), which exhibit highly distinct binary textures. Performance was relatively lower for Obfuscator.ACY (F1 = 0.86), likely due to the deliberate use of obfuscation techniques that alter the visual structure of the binary image.

TABLE III. Per-Class Precision, Recall, and F1-Score Across the Nine Malware Families.

Malware Family	Precision	Recall	F1-Score
Ramnit	0.93	0.92	0.92
Lollipop	0.95	0.94	0.94
Kelihos_ver3	0.91	0.90	0.90
Vundo	0.89	0.88	0.88
Simda	0.97	0.97	0.97
Tracur	0.90	0.89	0.89
Kelihos_ver1	0.98	0.98	0.98
Obfuscator.ACY	0.87	0.85	0.86
Gatak	0.92	0.91	0.91

C. Comparison with State-of-the-Art Methods

Table IV compares the proposed system against existing malware detection approaches reported in the literature. The ResNet-50-based method outperforms traditional machine learning classifiers and achieves competitive results against other CNN-based approaches.

TABLE IV. Comparison of the Proposed System with Existing Malware Detection Methods.

Method	Architecture	Accuracy	Year
SVM + Grayscale [5]	Hand-crafted features	~82%	2016
Random Forest [6]	Byte n-grams	~85%	2018
VGG-16 [7]	CNN (image-based)	~89%	2020
MalConv [8]	Raw byte CNN	~87%	2018
Inception-v3 [9]	CNN (image-based)	~91%	2021
Proposed ResNet-50	Deep CNN + Transfer Learning	92%	2025

D. Discussion

The experimental results confirm that the binary-to-image conversion coupled with ResNet-50 transfer learning is a highly effective strategy for malware classification. The model's strong performance across most malware families demonstrates that byte-level structural patterns captured in grayscale images are discriminative enough for automated family identification. The relatively lower performance on Obfuscator.ACY underscores a known challenge in the field: obfuscation intentionally destroys or alters the visual signatures that the model relies upon. Addressing this limitation through adversarial training or multi-view representations presents a promising direction for future work.

The system demonstrates significant advantages over traditional approaches: it requires no manually crafted signatures, generalizes to unseen variants sharing similar byte structures, and scales efficiently to large datasets. The automated nature of the pipeline also greatly reduces the burden on cybersecurity analysts, enabling faster threat identification in operational environments.

IV. CONCLUSION

This study proposed and evaluated a deep learning-based malware detection and classification system leveraging the ResNet-50 architecture. By converting malware binary executables into grayscale image representations, the system enables convolutional feature learning to capture family-specific structural patterns. The model achieved an overall classification accuracy of 92%, outperforming traditional machine learning baselines and demonstrating competitive performance against other CNN-based approaches.

The proposed pipeline automates the malware analysis process, reduces dependence on manually maintained signature databases, and offers scalability for real-world cybersecurity deployments. The primary limitation identified is reduced performance against heavily obfuscated malware variants. Future work will explore: (i) hybrid architectures that combine image-based and byte-sequence-based features; (ii) adversarial training to improve robustness against obfuscated samples; (iii) real-time deployment on cloud and edge security platforms; and (iv) explainability techniques such as Grad-CAM to provide interpretable detection decisions for security analysts.

ACKNOWLEDGEMENTS

The authors express sincere gratitude to Dr. A. Kalavathi, Head of the Department of Information Technology, and Dr. Y. Mallikarjuna Reddy, Principal, Vasireddy Venkatadri Institute of Technology, for their continuous encouragement and institutional support. Special thanks are extended to Dr. K. Kranthi Kumar, Project Coordinator, for his guidance throughout the project. The authors are grateful to their Project Supervisor, Dr. P. Nagababu, for his expert mentorship and unwavering support. No external funding was received for this study.

COMPETING INTERESTS

The authors declare that no competing interests exist.

AUTHORS' CONTRIBUTIONS

Nuthalapati Gopi Krishna designed the study, implemented the ResNet-50 model, and wrote the initial draft. Shaik Yasir contributed to data preprocessing and binary-to-image conversion. Vadlamudi Ramu performed model training, hyperparameter tuning, and evaluation experiments. Shaik Jaheer Basha assisted in testing, comparative analysis, and manuscript revision. Dr. P. Nagababu supervised the research and reviewed the manuscript. All authors read and approved the final manuscript.

ETHICAL APPROVAL

This study does not involve human participants or animal experiments. The datasets used are publicly available benchmark collections. Therefore, ethical approval and informed consent are not required.

REFERENCES

[1] A. Souri and R. Hosseini, "A state-of-the-art survey of malware detection approaches using data mining techniques," Human-centric Computing and Information Sciences, vol. 8, no. 1, p. 3, 2018.

[2] D. Ucci, L. Aniello, and R. Baldoni, "Survey of machine learning techniques for malware analysis," Computers & Security, vol. 81, pp. 123–147, 2019.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE CVPR, 2016, pp. 770–778.

[4] L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, "Malware images: Visualization and automatic classification," in Proc. VizSec, 2011, pp. 1–7.

[5] R. Ronen, M. Radu, C. Feuerstein, E. Yom-Tov, and M. Ahmadi, "Microsoft malware classification challenge," arXiv preprint arXiv:1802.10135, 2018.

[6] I. Firdausi, C. Lim, A. Erwin, and A. S. Nugroho, "Analysis of machine learning techniques used in behavior-based malware detection," in Proc. ICACSIS, 2010, pp. 201–206.

[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in Proc. ICLR, 2015.

[8] E. Raff, J. Barker, J. Sylvester, R. Brandon, B. Catanzaro, and C. Nicholas, "Malware detection by eating a whole EXE," in Proc. AAAI Workshop, 2018.

[9] C. Szegedy et al., "Going deeper with convolutions," in Proc. IEEE CVPR, 2015.

[10] OpenCV Library, "Open Source Computer Vision Library," 2024. [Online]. Available: https://opencv.org