Image-Based Malware Classification Using Deep Learning Models

Dipta Dhor, Sujoy Sarkar, Sadhon Kumar Dev, Saikat Sarkar
Department of Computer Science and Engineering
International Islamic University Chittagong (IIUC), Bangladesh
Emails: diptadhor2002@gmail.com, sujoy9221@gmail.com, skdsadhon@gmail.com, sarkarsaikat3467@gmail.com

***Abstract*—Malware detection has become increasingly challenging due to the rapid evolution of malicious software and the limitations of traditional signature-based detection techniques. This paper presents an image-based malware classification framework using deep learning models. Executable binaries are transformed into grayscale images, enabling malware classification as a computer vision problem. A dataset consisting of 29 classes, including benign software and multiple malware families, is used to evaluate three models: a custom Convolutional Neural Network (CNN), MobileNetV2, and Vision Transformer (ViT). To address severe class imbalance, class-weighted loss functions are applied during training. While MobileNetV2 utilizes pretrained weights through transfer learning, the Vision Transformer is trained from scratch due to the unavailability of pretrained weights. Experimental results demonstrate that MobileNetV2 achieves superior performance with a macro F1-score of 91.33%, highlighting the effectiveness of transfer learning for robust malware classification. The contributions of this study include the creation of a custom malware-image dataset, implementation of a lightweight CNN, evaluation of transfer learning and transformer-based models, and insights on class imbalance handling in malware image classification.**

***Index Terms*—Malware Classification, Image-Based Analysis, Deep Learning, MobileNetV2, Cybersecurity**

I. Introduction

Malware represents a significant threat to modern computing systems due to its increasing volume, diversity, and sophistication. New variants such as zero-day malware, polymorphic malware, ransomware, and trojans challenge traditional signature-based detection methods, which rely on known patterns or heuristics. These methods often fail to detect novel or obfuscated malware, leading to severe security risks. As a result, intelligent malware detection techniques leveraging machine learning and deep learning have gained substantial attention.

The rapid expansion of the digital ecosystem, including cloud computing, Internet of Things (IoT) devices, and mobile platforms, has further amplified the attack surface for malware developers. Modern malware is no longer limited to simple destructive payloads; instead, it is often designed to remain stealthy, persistent, and adaptive. This evolution makes static rule-based detection increasingly ineffective, as attackers continuously modify malware signatures to evade detection systems.

In response to these challenges, researchers have explored behavior-based and learning-based detection techniques that can automatically learn discriminative patterns from data. Machine learning models, particularly deep learning architectures, have demonstrated strong capabilities in handling large-scale, high-dimensional data and extracting meaningful representations without extensive manual feature engineering. These advantages make deep learning a promising solution for addressing the limitations of conventional malware detection approaches.

In this study, we focus on image-based malware classification, where binary executables are visualized as grayscale images. This approach preserves structural and content patterns of the binaries, enabling the use of deep learning models originally designed for computer vision. By transforming malware detection into an image classification problem, complex feature extraction is automated, improving the scalability and adaptability of detection systems.

The idea of representing binaries as images is motivated by the observation that malicious and benign executables often exhibit distinct visual textures and spatial distributions when mapped to pixel intensities. These visual characteristics can be effectively captured by convolutional filters, allowing convolutional neural networks (CNNs) to learn hierarchical patterns related to opcode distribution, entropy variations, and file structure. Compared to traditional feature-based malware analysis, image-based techniques significantly reduce domain-specific preprocessing requirements.

Furthermore, recent advances in transfer learning and transformer-based models have opened new research directions in malware analysis. Lightweight pretrained models can leverage knowledge learned from large-scale image datasets, while transformer architectures offer the ability to capture long-range dependencies within visual representations. Evaluating such models in the context of malware classification provides valuable insights into their suitability for cybersecurity applications.

Our contributions in this paper include:

- Creation of a custom malware-image dataset from benign and malicious binaries, ensuring a diverse set of classes for robust evaluation.
- Implementation of a lightweight and interpretable CNN architecture optimized for malware image classification.
- Evaluation of transfer learning with MobileNetV2 and comparison with a Vision Transformer trained from scratch.

- Application of class-weighted loss to handle severe class imbalance in malware datasets.
- Detailed experimental analysis, including confusion matrices, ROC curves, and performance comparisons.

Additionally, we provide a comprehensive analysis of dataset preparation, preprocessing, model evaluation, and limitations, and suggest directions for future work involving pretrained transformers and hybrid CNN-Transformer architectures.

A. Motivation and Research Significance

The rapid growth of malware variants has significantly increased the complexity of cybersecurity defense mechanisms. Traditional signature-based antivirus systems struggle to keep pace with emerging threats, particularly zero-day and polymorphic malware. This motivates the exploration of learning-based approaches that can generalize beyond known attack patterns. Image-based malware classification offers a novel perspective by leveraging advances in computer vision to detect malicious behavior from structural representations of executables.

From a research perspective, this approach bridges the gap between cybersecurity and computer vision, enabling cross-domain innovation. By reusing established vision architectures for malware detection, researchers can build scalable and efficient systems without designing handcrafted features. This work contributes to the growing body of research that demonstrates the feasibility and effectiveness of visual representations for security analytics.

B. Research Objectives

The primary objectives of this research are to investigate the effectiveness of image-based malware classification, evaluate the impact of transfer learning on malware detection performance, and analyze the feasibility of transformer-based architectures in cybersecurity applications. This work also aims to highlight practical challenges such as class imbalance and dataset bias in real-world malware datasets.

II. Literature Review

Malware detection has been extensively studied in the past decade, with image-based techniques gaining popularity due to their effectiveness in visual pattern recognition.

Early malware detection research primarily relied on signature-based and heuristic-driven approaches. While effective against known threats, these techniques struggle with zero-day attacks and polymorphic malware. As malware authors began employing obfuscation and packing techniques, researchers shifted toward machine learning-based detection methods capable of learning patterns from data rather than relying on handcrafted rules.

A significant breakthrough in this domain was the introduction of malware visualization techniques. Nataraj et al. [1] introduced malware visualization by converting binary files into grayscale images, demonstrating that image representations can reveal structural differences among malware families. Their work showed that malware binaries exhibit distinctive visual textures when mapped to pixel intensities, making them suitable for classification using image processing techniques. This study laid the foundation for treating malware detection as a computer vision problem.

Following this pioneering work, deep learning approaches gained traction due to their ability to automatically extract hierarchical features. Kiger et al. [2] further applied convolutional neural networks (CNNs) for malware image classification, showing the effectiveness of deep learning in automated feature extraction. Their results confirmed that CNNs outperform traditional machine learning classifiers by learning spatial patterns directly from malware images without manual feature engineering.

Recent studies have explored more advanced architectures and transfer learning techniques. Ali et al. [4] proposed MalwareVision, a deep learning-driven approach for malware classification, highlighting the potential of CNNs on custom malware datasets. Their work emphasized the importance of dataset quality and preprocessing in achieving robust detection performance. Musa et al. [5] utilized deep CNN models for improved malware detection, demonstrating the importance of model depth and architecture selection in capturing complex malware structures.

With the emergence of transformer-based models in computer vision, several researchers have investigated their applicability to malware classification. Ashawa et al. [6] combined CNNs and Vision Transformers for enhanced image-based malware classification, showing improved performance in complex malware environments. Their hybrid approach leveraged CNNs for local feature extraction and transformers for capturing long-range dependencies within malware images. Alshomrani et al. [7] proposed a hybrid CNN-Transformer architecture that improves explainability and classification performance, addressing one of the major limitations of deep learning-based security systems.

In addition to binary classification, multi-class malware classification has also been widely studied. Yadav et al. [8] focused on multi-class malware classification using visualization and CNNs, demonstrating the feasibility of distinguishing between multiple malware families based on visual characteristics. Chaganti et al. [9] applied EfficientNet models to achieve highly accurate malware classification with optimized computation, highlighting the role of lightweight and scalable architectures for real-world deployment.

Despite these advancements, challenges remain in handling class imbalance, dataset limitations, and the lack of pretrained transformer models for malware images. Many existing studies rely on relatively small or imbalanced datasets, which can negatively impact model generalization. Furthermore, transformer-based models often require large-scale labeled datasets, which are scarce in the malware domain. This work addresses these gaps by comparing CNN-based, transfer learning-based, and transformer-based models on a large, multi-class malware-image dataset, while explicitly addressing class imbalance through weighted loss functions.

III. Methodology

This section describes the overall workflow adopted in this study, including dataset construction, preprocessing steps, class imbalance handling strategies, model architectures, and the experimental environment. The complete methodology is designed to ensure reproducibility, robustness, and fair comparison across different deep learning models.

A. Dataset Description

The dataset used in this study consists of malware and benign executable files converted into grayscale images. Benign samples include legitimate DLL, EXE, MUI, and other verified Windows system files. Malware samples span multiple families, resulting in a 29-class classification problem. Malware samples were collected from publicly available repositories, including the MalImg dataset and Kaggle archives, while benign samples were extracted from verified Windows system files. A summary of the dataset is shown in Table I.

The inclusion of both benign and malicious binaries ensures that the dataset closely reflects real-world malware detection scenarios. By covering multiple malware families, the dataset enables multi-class classification rather than simple binary detection, making the task more challenging and practically relevant.

TABLE I
Dataset Summary

Description	Value
Total images	19,209
Benign images	9,870
Malware images	9,339
Number of classes	29
Training samples	15,379
Validation samples	3,830
Image size	224×224

To ensure fair evaluation, the dataset was randomly shuffled before splitting, and care was taken to maintain class distribution consistency across training and validation sets. This helps reduce evaluation bias caused by uneven class representation.

B. Data Preprocessing

Binary files were read as byte sequences and converted into grayscale images by mapping byte values (0–255) to pixel intensities. Each image was resized to 224×224 pixels and normalized to the range [0,1]. The dataset was split into training and validation sets using an 80:20 ratio. Basic data augmentation techniques, such as small rotations, flips, and shifts, were applied to enhance generalization.

Preprocessing plays a critical role in image-based malware classification, as inconsistencies in image dimensions or pixel distributions can significantly impact model convergence. Normalization was applied to stabilize gradient updates, while resizing ensured compatibility with pretrained architectures such as MobileNetV2. Augmentation techniques were intentionally kept minimal to avoid altering the intrinsic structural patterns of binary images.

C. Class Imbalance Handling

The dataset exhibits severe class imbalance. To address this issue, class weights were computed using a balanced weighting strategy and applied during model training to prevent bias toward majority classes.

Class imbalance is a common challenge in malware datasets, where certain malware families are underrepresented. Without proper handling, deep learning models tend to favor dominant classes, leading to poor detection performance for rare but critical malware types. By incorporating class-weighted loss, the training process penalizes misclassification of minority classes more heavily, resulting in improved overall robustness.

D. Model Architectures

Three deep learning models were evaluated to analyze the trade-offs between accuracy, computational efficiency, and generalization capability.

Custom CNN: A baseline convolutional architecture designed to learn spatial features from malware images. It consists of multiple convolutional and pooling layers followed by fully connected dense layers optimized for malware classification. This model serves as a benchmark to evaluate the effectiveness of handcrafted CNN architectures without external pretrained knowledge.

MobileNetV2: A lightweight pretrained CNN using depthwise separable convolutions and transfer learning with ImageNet weights. Transfer learning allows the model to leverage pre-learned visual features to achieve high accuracy despite a limited dataset. Fine-tuning was selectively applied to higher layers to adapt the pretrained features to malware-specific visual patterns.

Vision Transformer (ViT): A transformer-based model that processes images as fixed-size patches using self-attention mechanisms. The ViT was trained from scratch without pretrained weights due to availability constraints, which affected its generalization performance. Despite this limitation, the model was included to evaluate the feasibility of transformer-based architectures in malware image classification tasks.

E. Binary-to-Image Conversion Process

Each executable file was processed as a raw byte stream. The byte values were sequentially arranged into a two-dimensional matrix, where each byte corresponds to a pixel intensity in a grayscale image. Files shorter than the required size were padded, while larger files were truncated to ensure consistent image dimensions. This representation preserves local structural patterns that are useful for visual feature extraction.

This conversion technique avoids semantic interpretation of executable code and focuses purely on structural information, making it resilient to code obfuscation techniques that manipulate instruction sequences without significantly altering byte-level distributions.

F. Experimental Environment

All experiments were conducted using the TensorFlow and Keras frameworks on Google Colab with GPU acceleration. The Adam optimizer was used with a learning rate of 1×10^{-4}. Models were trained for a maximum of 50 epochs, with early stopping applied to prevent overfitting. Batch normalization and dropout were employed where applicable to enhance generalization.

Early stopping was configured to monitor validation loss, ensuring that training terminated once performance stagnated. This strategy helped reduce unnecessary computation while preventing overfitting, particularly for deeper architectures. All experiments were executed under identical settings to ensure a fair comparison between models.

IV. Evaluation Metrics

Model performance was evaluated using accuracy, macro-averaged precision, recall, and F1-score. Macro averaging ensures equal importance for all classes.

In multi-class malware classification, relying on a single metric can be misleading, especially when class imbalance is present. Therefore, multiple evaluation metrics were employed to provide a comprehensive assessment of classification performance. Macro-averaged metrics were specifically chosen to ensure that minority malware families contribute equally to the final evaluation, regardless of their sample size.

A. Accuracy

Accuracy measures the overall proportion of correctly classified samples across all classes. While it provides a general indication of model performance, accuracy alone may not fully reflect classification quality in imbalanced datasets.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

In the context of malware detection, high accuracy indicates that the model correctly distinguishes between benign and malicious samples, as well as among different malware families. However, additional metrics are required to assess class-specific performance.

B. Precision

Precision measures the proportion of correctly identified positive samples among all samples predicted as positive. It reflects the reliability of positive predictions made by the model.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

High precision is particularly important in malware classification, as false positives may incorrectly label benign software as malicious, leading to unnecessary alerts or system disruptions.

C. Recall

Recall, also known as sensitivity, measures the proportion of actual positive samples that are correctly identified by the model.

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

In cybersecurity applications, recall is a critical metric because false negatives represent undetected malware, which can pose serious security threats. A high recall value indicates that the model is effective at identifying malicious samples.

D. F1-score

The F1-score provides a balanced measure of model performance by combining precision and recall into a single metric. It is particularly useful when dealing with imbalanced datasets.

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4}$$

A high F1-score indicates that the model achieves a good balance between minimizing false positives and false negatives, making it a reliable indicator of overall classification effectiveness in multi-class malware detection tasks.

V. Results

This section presents the experimental results obtained from evaluating the proposed deep learning models on the malware image dataset. Both quantitative metrics and qualitative interpretations are provided to analyze model performance and highlight key observations.

A. Quantitative Evaluation

Table II presents the macro-averaged performance metrics of all models.

The macro-averaged evaluation strategy ensures that each malware class contributes equally to the final scores, making the comparison particularly meaningful in the presence of class imbalance. The reported metrics include accuracy, precision, recall, and F1-score, which collectively provide a comprehensive view of each model's classification capability.

TABLE II
Model Performance Comparison

Model	Accuracy	Precision	Recall	F1-score
Custom CNN	0.7279	0.5897	0.8024	0.6387
MobileNetV2	**0.9708**	**0.9013**	**0.9283**	**0.9133**
Vision Transformer	0.5235	0.5032	0.7311	0.5447

The results demonstrate notable performance differences among the evaluated architectures, reflecting the impact of model design, training strategy, and availability of pretrained weights.

B. Result Interpretation

The results indicate that transfer learning significantly improves malware classification performance. MobileNetV2 consistently outperformed the custom CNN and Vision Transformer across all evaluation metrics.

The superior performance of MobileNetV2 can be attributed to its use of pretrained ImageNet weights, which provide robust low-level and mid-level feature representations. These pretrained features generalize well to malware image patterns, enabling faster convergence and improved accuracy even with limited training data. Additionally, the lightweight architecture of MobileNetV2 makes it computationally efficient while maintaining high classification performance.

The custom CNN achieved moderate performance, demonstrating its ability to capture meaningful spatial features from malware images. However, its limited depth and lack of pretrained knowledge constrained its ability to model complex visual patterns across multiple malware families. This highlights the trade-off between model simplicity and representational capacity.

The Vision Transformer suffered from insufficient training data and lack of pretrained weights, highlighting the importance of large-scale pretraining for transformer-based models. Although the transformer architecture is capable of modeling long-range dependencies, training from scratch on a relatively small dataset led to suboptimal generalization. This result suggests that transformer-based approaches may require extensive pretraining or hybrid architectures to be effective in malware image classification tasks.

Overall, the results confirm that transfer learning-based CNN models offer a strong balance between performance and efficiency for image-based malware detection, while transformer-based models remain challenging to deploy without access to large-scale pretrained malware datasets.

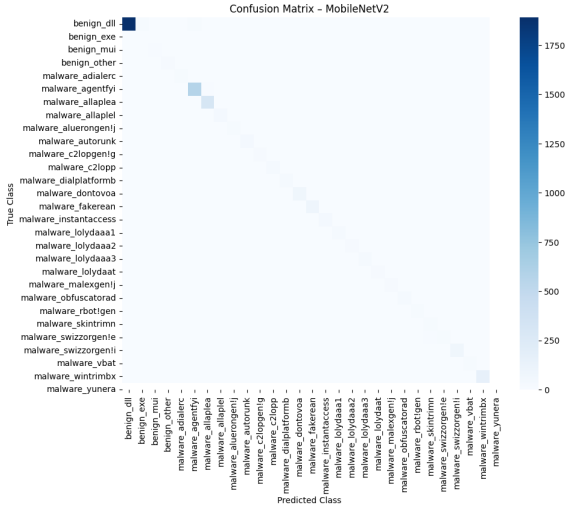
C. Performance Analysis

This subsection presents a detailed performance analysis of the evaluated models using confusion matrices, ROC–AUC curves, and comparative metric visualizations to highlight classification effectiveness across all classes.

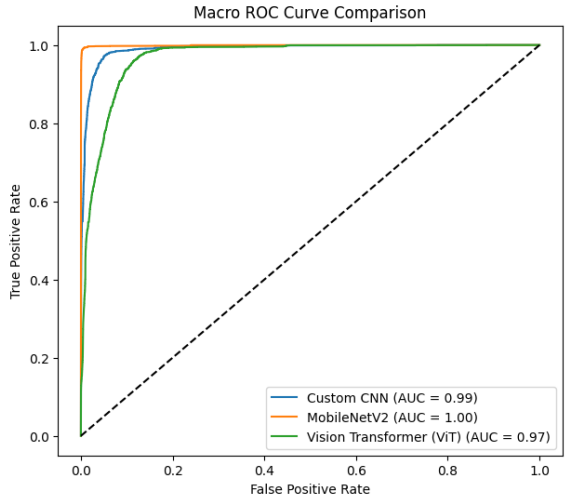
Figure 1 illustrates three key visualizations that support the quantitative findings:

- The confusion matrix of MobileNetV2 (Figure 1a) shows strong diagonal dominance, indicating high classification accuracy across most malware classes. Misclassifications are minimal and primarily concentrated in visually similar malware families. - The macro ROC–AUC curve comparison (Figure 1b) highlights the superior discriminative capability of MobileNetV2, which achieves an AUC of 1.000. The custom CNN and Vision Transformer also perform well, with AUCs of 0.939 and 0.917 respectively, but fall short of MobileNetV2's perfect separation. - The performance comparison bar chart (Figure 1c) provides a clear visual summary of accuracy, precision, recall, and F1-score across all models. MobileNetV2 consistently leads in every metric, reinforcing its robustness and generalization ability.

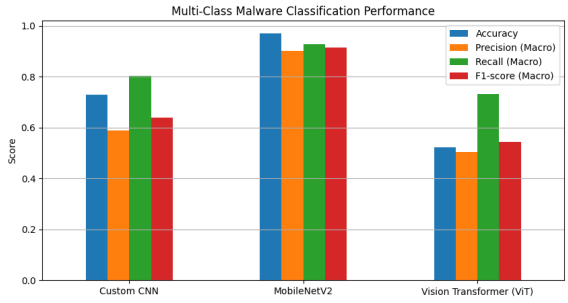
These visualizations confirm the numerical trends observed in Table II and offer deeper insights into model behavior. The confusion matrix helps identify specific class-level strengths and weaknesses, while the ROC–AUC curves and bar charts provide a holistic view of overall performance.



(a) Confusion Matrix of MobileNetV2



(b) Macro ROC–AUC Curve Comparison



(c) Performance Comparison Bar Chart

Fig. 1. Performance analysis of evaluated models using confusion matrix, ROC–AUC curves, and comparative metrics.

VI. Discussion

The experimental results demonstrate that MobileNetV2 outperforms the custom CNN and Vision Transformer across all evaluation metrics. Transfer learning enables MobileNetV2 to extract robust features despite severe class imbalance, whereas the Vision Transformer trained from scratch without pretrained weights struggles to generalize effectively.

The superior performance of MobileNetV2 highlights the importance of leveraging pretrained visual knowledge when working with limited or imbalanced malware datasets. Pretrained convolutional filters capture generic edge, texture, and shape information that transfers effectively to malware image representations, improving convergence speed and classification stability.

The custom CNN provides a reasonable baseline but is constrained by limited depth and absence of transfer learning. While it captures low-level structural patterns, its capacity is insufficient to model complex variations across multiple malware families. Minority classes remain challenging even with class-weighted loss, indicating that image-based static analysis should be complemented with additional techniques for comprehensive detection.

A. Comparison with Existing Studies

The results align with prior research showing CNN-based architectures are effective for malware image classification. MobileNetV2 offers improved accuracy and efficiency compared to traditional CNNs, making it suitable for resource-constrained deployment. Unlike many previous studies that focus on binary classification, this work addresses multi-class malware detection with a larger number of families, demonstrating the scalability of transfer learning-based CNNs. The lower performance of the Vision Transformer is consistent with literature emphasizing the need for large-scale pretraining for transformer models.

B. Security and Deployment Considerations

Image-based malware detection is effective for static analysis but should be integrated with dynamic analysis for comprehensive security. Lightweight models like MobileNetV2 are suitable for endpoint security, cloud-based scanning, and real-time monitoring. Continuous updates, retraining, and monitoring false positives are essential to maintain trust and reliability in production environments.

VII. Limitations

Despite promising results, this study has several limitations:

1. **Static analysis constraints:** The approach relies solely on binary-to-image representations, which may fail to detect heavily obfuscated, packed, or encrypted malware and cannot capture runtime behaviors such as system calls or network activity.

2. **Transformer limitations:** The Vision Transformer was trained from scratch without pretrained weights, resulting in suboptimal convergence and generalization. Transformer-based models require large-scale pretraining to perform effectively in malware classification tasks.

3. **Dataset limitations:** Although multi-class, the dataset may not fully represent newly emerging malware families. Minority classes remain challenging, and visual representations may not fully capture semantic differences between benign and malicious binaries.

4. **Binary-to-image abstraction:** Converting binaries to grayscale images abstracts away execution semantics, potentially limiting differentiation between structurally similar but behaviorally distinct files.

These limitations suggest that image-based malware classification should complement, rather than replace, other static or dynamic analysis techniques.

VIII. Conclusion

This paper presented an image-based malware classification framework using deep learning models. Experimental evaluation confirms that MobileNetV2 achieves superior performance for multi-class malware classification. By creating a custom malware-image dataset and applying class-weighted loss, this study demonstrates the effectiveness of transfer learning and deep CNN architectures in cybersecurity.

Pretrained convolutional models provide strong generalization even with limited or imbalanced training data. The lightweight MobileNetV2 architecture ensures computational efficiency and practical deployment in real-world security systems. Image-based representations simplify malware detection by automating feature extraction, making the approach scalable for large datasets and continuous monitoring.

While transformer-based models show potential, their effectiveness depends on access to large-scale pretraining and extensive labeled datasets. The comparatively lower performance of the Vision Transformer indicates the need for further research before reliable deployment.

A. Future Work

Future research will explore:

- Pretrained Vision Transformers and hybrid CNN-Transformer architectures for enhanced generalization.
- Expansion of the dataset to include more malware families and diverse benign software.
- Advanced augmentation techniques to simulate obfuscation and packing patterns.
- Integration of model explainability methods to visualize learned features.
- Combination of static image-based analysis with dynamic behavioral features for robust malware detection.
- Continual learning strategies to adapt models to evolving malware trends.

Acknowledgment

The authors would like to express their sincere gratitude to Muhammad Mubinur Rahman, Lecturer, Department of Computer Science and Engineering, International Islamic University Chittagong (IIUC), for his valuable guidance, constructive

feedback, and continuous support throughout the course of this research.

The authors also acknowledge the use of Google Colab for providing a secure and efficient computational environment, as well as the open-source research community for making the MalImg and Kaggle malware datasets publicly available. Additionally, verified Windows system files were utilized for benign sample collection, ensuring the reliability and integrity of the dataset used in this study.

REFERENCES

[1] L. Nataraj et al., "Malware Images: Visualization and Automatic Classification," IEEE VizSec, 2011.
[2] J. Kiger et al., "Malware Binary Image Classification Using CNNs," ICCWS, 2022.
[3] Kaggle, "Malware Classification Dataset (MalImg)."
[4] A. Ali, M. Akram, W. Farooq, M. Ali, M. Nazir, A. Muhammad, T. Mazhar, "MalwareVision: A Deep Learning-Driven Approach For Malware Classification," 2024.
[5] H. O. Musa, M. T. Younis, "Image-Based Malware Detection Using Deep CNN Models," IJCI, 2024.
[6] M. Ashawa, N. Owoh, S. Hosseinzadeh, J. Osamor, "Enhanced Image-Based Malware Classification Using Transformer-Based CNNs," Electronics Journal, 2024.
[7] M. Alshomrani, A. Albeshri, A. Alsulami, B. Alturki, "An Explainable Hybrid CNN–Transformer Architecture for Visual Malware Classification," Sensors, 2025.
[8] B. Yadav, S. Tokekar, "Malware Multi-Class Classification based on Malware Visualization using a CNN Model," IJIEEB, 2023.
[9] R. Chaganti, V. Ravi, T. Pham, "Image-based malware representation approach with EfficientNet CNNs for effective malware classification," JISA, 2022.