Open Access and Peer Review Journal ISSN 2394-2231 https://ijctjournal.org/

A Comparative Analysis of Deep Learning Models for Malware Detection Using Image-Based Features

Gursangat Singh
Department of Computer Science and Engineering
Punjabi University
Patiala, India
gsingh.learn@gmail.com

Harmandeep Singh
Department of Computer Science and Engineering
Punjabi University
Patiala, India
harmanjhajj@yahoo.co.in

***Abstract*— Nowadays, malware has been more advanced, trickier, and complex; as a result, we need reliable and scalable detection tools that work across all sorts of threats. Recently, some advancement has been made in image-based malware representation, where threats can be analysed as visual patterns and deep learning models can be used to analyse those patterns, which can be difficult for traditional analysis techniques. In this paper, a systematic comparative analysis of five different deep learning models, such as CNN, VGG16, ResNet-50, DenseNet-121, and EfficientNet-B0, is done for sorting malware into multiple classes using RGB representation. We tested all these 5 models on a public benchmark dataset with matching pre-processing, splits, training setups, and different evaluation metrics to keep it fair and repeatable. Covering lightweight task-specific networks to deep pre-trained models, this study provides hands-on insights for picking and benchmarking models in malware image analysis in cyber defence.**

Keywords—Malware detection, Deep Learning, Convolutional Neural Network, Malware Image Classification, Cybersecurity, Confusion Matrix, False Negative Reduction

I. INTRODUCTION

The threat associated with malware has significantly increased due to the exponential advancement of Inter-connected system and software-driven infrastructures. Modern malicious software no longer relies solely on signature-based threats but has developed into a complex, adaptive, and polymorphic variant that evades traditional detection systems. Because of this, conventional signature-based and rule-based detection approaches are no longer effective in handling emerging and sophisticated malware threats [1].

Recent research highlights deep learning(DL) as a promising alternative for malware detection due to its capability to identify complex data patterns automatically from a large volume of data. Notably, **image-based malware analysis** has become a popular subject of substantial attention, where executable files are transformed into grayscale or colour images and analyzed using convolutional neural network architectures (CNNs) [2][3]. This approach reformulates malware detection as an image classification problem, benefiting from the powerful feature extraction capabilities of deep learning models originally designed for computer vision applications [4][5][6].

Numerous studies have demonstrated that combining malware visualization with CNN-based architectures yields high detection accuracy without the need for manual feature engineering. Early studies revealed that visual similarities and textural characteristics can effectively distinguish between benign and malicious samples [7][8]. Expanding on this idea, researchers have examined multiple deep learning architectures, including lightweight CNNs, transfer learning models, ensemble frameworks, and hybrid approaches, to enhance classification accuracy and robustness [9][10].

Even with these advancements, it remains unclear which deep learning model achieves the best performance when tested within a consistent experimental setup. Many existing studies differ in terms of choice of datasets, preprocessing strategies, image dimensions, and evaluation criteria, making it difficult to conduct a fair assessment [1]

Moreover, some studies prioritize detection accuracy, others aim to optimize efficiency, scalability, or robustness against adversarial attacks, leading to fragmented findings across the literature [11][12][13].

Addressing these limitations requires a structured comparative evaluation of deep learning models using a common dataset, and consistent evaluation metrics are required [14]. This approach allows for fair evaluation, highlighting their strengths and weaknesses while offering a clearer interpretation of model effectiveness in practice.

II. RELATED WORK

In cybersecurity, Malware is defined as malicious software intentionally engineered to compromise system integrity, steal sensitive information, or gain unauthorized access to networks and devices [15]. Traditional malware detection methods relied heavily on signature-based and behavior-based analysis, which are effective for known threats only, while struggling against unseen variants such as polymorphic and zero-day malware variants [8][16]. To address these shortcomings, machine learning techniques were introduced to automatically learn discriminative patterns from malware datasets and enhance detection reliability [17]. However, the rising complexity and sheer volume of modern malware have revealed scalability and adaptability limitations in conventional and shallow learning techniques, prompting the shift towards more sophisticated data-driven detection approaches [13].

In recent years, due to the growing complexity of cyber threats, the research in the field of malware detection has expanded. Traditional signature-based and heuristic approaches have proven inadequate against zero-day attacks, obfuscation, and polymorphic malware, encouraging researchers to explore machine learning (ML) and deep learning (DL)–based solutions [18][2]. This section reviews existing studies related to image-based malware visualization, deep learning models for malware classification, and comparative and ensemble-based approaches, along with key strengths and limitations.

A. *Image-Based Malware Representation*

A widely explored research approach to utilize robust image processing and analysis techniques, which involves mapping malware binaries into visual forms. [6][19]. Early investigations indicated that binary-to-image transformation retains distinctive structural information of malware families, enabling accurate visual-based classification. Chu *et al.* [7] introduced a homology-based visualization method that extracts visual features from malicious code and applies convolutional neural networks (CNNs) for classification, confirming the presence of unique textures.

Later investigations further optimized this approach by testing different image dimensions, colour representation, and feature extraction techniques. Hawana *et al.* [9] analyzed how modifying image resolution influences CNN-based malware detection and concluded that resized images can significantly influence noticeable changes in classification accuracy. Priya and Sofia [10] strengthened image-based detection by integrating Gray-level co-occurrence(GLCO) matrices with sparse convolution, resulting in enhanced discriminative performance. By utilizing transfer learning and expanding training data through augmentation, Li et al. [1] and Nguyen & Nguyen [20] observed notable gains in malware image detection outcomes.

These findings confirm the viability of image-based malware representation, but studies often adopt diverse datasets, preprocessing pipelines, and evaluation settings, which leads to restricted fair comparison [17][21].

B. *Deep Learning Models for Malware Image Classification*

Convolutional neural network (CNN)-based architectures remain the dominant approach for malware image classification, primarily because of their superior feature-learning capabilities [2][3][13]. The MIRACLE framework proposed by Alam *et al.* [22] is a multi-layered feature extraction model that exhibits strong performance across diverse malware classes. Tariq *et al.* [23] implemented a parameter-tuned CNN model that results in measurable gains in detection accuracy when the architecture is carefully optimized.

Recently, transfer learning approaches have become popular, with many studies utilizing pre-trained models such as VGG, ResNet, and lightweight CNNs. Johny *et al.* [24] confirmed that merging visual features with transfer-learned CNN architectures yields improved malware detection accuracy. Hota *et al.* [25] highlight the compromise between model complexity and performance by focusing on lightweight CNN architectures designed for limited resource platforms. Saxe & Berlin [17] and Ye et al. [2] highlighted that DL models surpass traditional ML by extracting features automatically and reliably for classification.

Although these studies report encouraging outcomes, the majority of the studies focus on a small set of models and on a single architecture, which limits broader generalization.

C. *Hybrid, Ensemble, and Advanced Deep Learning Approaches*

To address the shortcomings of standalone models, researchers have turned to ensemble-based and hybrid approaches. Bakır [26] introduced VoteDroid, a voting based ansemble system that integrates multiple fine-tuned deep learning models, which leads to improved robustness. Nazim *et al.* [27] emphasized a multimodal ensemble system that integrates multiple deep neural networks for malware classification.

Federated and adversarial learning strategies have also attracted attention in recent studies. Ambekar *et al.* [28] proposed a federated adversarial Siamese network (FASNet) tailored for resilient malware image classification across distributed environments. Zhao *et al.* [29] examined a hybrid approach that combines transfer learning with adversarial techniques to enhance resistance against evasion attempts. Raff et al. [3] and Mohaisen et al. [21] further showed that the ensemble-based approach improves both the detection accuracy and operational robustness in a real-world malware environment.

Even though these approaches improve robustness, they often come with the cost of added computational complexity and are evaluated under limited uniformity [26][27].

D. *Survey and Review Studies*

Multiple survey papers and review articles present broader perspectives on existing malware detection techniques. Berrios *et al.* [30] conducted a systematic review of malware detection techniques, focusing on image-driven deep learning techniques and the challenges they face. Almobaideen *et al.* [31] and Gundoor and Sridevi [32] identified persistant issue related to scalability, dataset imbalance, and reproducibility while reviewing ML and DL methods for Android and IoT malware detection. Buczak & Guven [16] and Zhang et al. [13] presented a foundational overview of data-driven malware detection using deep learning models.

These surveys highlight the absence of standardized benchmarking and consistent comparative evaluations using a unified dataset across deep learning models.

III. METHODOLOGY

This section outlines the complete experimental workflow adopted for malware detection, covering dataset preparation, image preprocessing, deep learning architectures, training configuration, and evaluation strategy. The methodology aims to provide a standardized and equitable comparison among various deep learning models. The methodology is designed to ensure a fair and consistent comparison among different deep learning models, in line with the best practices recommended from recent studies [13][18][16].

A structured experimental framework is followed in this study for classifying malware and benign images through deep learning models. As illustrated in Figure 1, the process begins with data selection and verification, then proceeds through preprocessing, training the model, and evaluating the performance. Each phase is carefully designed to maintain data integrity, ensure fair model comparison, and support reproducibility.

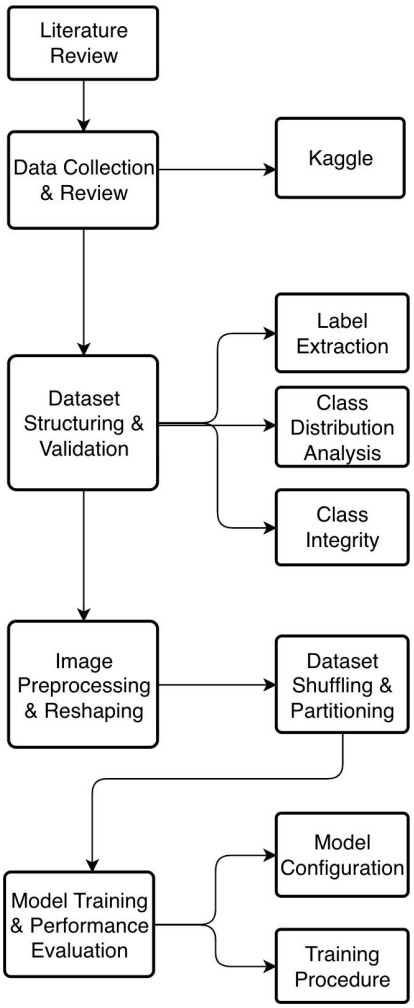

Fig.1. Methodology Workflow

A. *Literature Review*

The proposed methodology is built upon an extensive examination of existing studies on image-based malware detection and deep learning driven classification techniques. Studies indicated that converting malware binaries into images enables the extraction of structural and textural-related features beneficial for classification tasks. Accordingly, the literature informed the selection of datasets, preprocessing strategies, model design, and evaluation strategy to ensure consistency with established best practices.

B. *Data Collection and Review*

The experimental analysis is conducted on the publicly accessible Malware-Benign Image classification dataset available on Kaggle. It consists of 22,056 images spanning eight categories, including Adware, Backdoor, Downloader, Spyware, Trojan, Virus, Worms, and Benign software [33]. Owing to its widespread use in recent studies, the dataset is well-suited for benchmarking and comparative analysis [6][19].

An initial data inspection was conducted to confirm the completeness, consistency, and suitability of the dataset. This process ensured that all samples were correctly labeled and appropriate for supervised deep learning experiments [9].

C. *Dataset Structuring and Validation*

The dataset is carefully validated through a structured framework including label extraction, examination of class balance, and verification of class integrity [1].

Label extraction is carried out to ensure each image corresponds to the appropriate malware or benign class.

Class distribution analysis is performed to evaluate the balance across malware categories to detect potential skewness.

Class integrity checks ensure that corrupted, duplicated, or mislabeled instances are excluded [14]. Following this, the validated dataset is then divided into three exclusive partitions to guarantee consistent experimental comparison.

- **Training set:** 15,439 samples
- **Validation set:** 3,308 samples
- **Test set:** 3,309 samples

The same data partition scheme is maintained throughout all experiments to ensure unbiased evaluation and experimental consistency.

D. *Image Preprocessing and Reshaping*

In the dataset, every malware and benign image is represented as an RGB image, and although the original size of the image was 299 × 299 pixels, all the images are resized to 128 × 128 pixels before training [9]. This reduction lowers computational overhead while retaining the essential structural and visual features needed for effective learning [9][20].

During the training, Normalization of pixel values is done to enhance the numerical stability and facilitate faster convergence [2][17]. Representing binaries as RGB images enables the model to identify spatial structures and texture patterns within malware files, a technical validation by earlier image-based malware detection [19].

E. *Dataset Shuffling and Partitioning*

Once preprocessing is completed, the dataset undergoes random shuffling to mitigate ordering bias that could impact model performance. The shuffled samples are subsequently partitioned into training, validation, and testing sets according to a predefined split ratio. This approach guarantees that each class is represented in all subsets while keeping the sets mutually exclusive.

F. *Model Configuration*

To ensure a well-rounded comparison, five deep learning models are chosen, including both a custom-designed architecture and pre-trained models [2]. The selection of models varies in complexity and computational cost, making it possible to assess how performance balances against efficiency [24].

The selected models consist of:

- **Custom Convolutional Neural Network (CNN)**
- **VGG16**
- **ResNet-50**

- **DenseNet-121**
- **EfficientNet-B0**

Despite being conventionally optimized for larger input sizes, all models are consistently modified to accept a **128 × 128 × 3** image size, ensuring a fair and consistent evaluation.

Each convolutional layer is configured with **ReLU activation** and same padding to ensure consistent spatial representation. The output layer incorporates **Softmax activation** to generate a class probability distribution for multi-class prediction.

TABLE I. SUMMARIZES DEEP LEARNING MODEL, ARCHITECTURES, AND FEATURES

Model Type	Architecture Type	Depth (Layers)	Total Parameters (M)	Feature Learning Strategy
Custom CNN	Sequential	13	2.0	Hierarchical spatial features with Global Average Pooling
VGG16	Deep	16	16.8	Deep stacked convolutions with fixed features extracted
ResNet-50	Residual (Skip)	50	32.0	Residual learning via identity skip connections
DenseNet-121	Dense	121	11.2	Dense feature reuse through direct layer connectivity
EfficientNet-B0	Compound-scaled	82	9.3	Balanced depth-width-resolution scaling

Table 1 highlights the architectural diversity of the evaluated models, ranging from lightweight sequential CNNs to deep residual and densely connected networks, enabling a fair analysis of complexity–performance trade-offs in malware image classification.

G. Training Procedure

To maintain experimental fairness, all models are trained under the same training configurations. Adam is chosen as the optimizer due to its ability to adjust the learning rate dynamically and efficient convergence properties. Sparse categorical cross-entropy is used as the objective loss function, which gives its effectiveness in handling multi-class classification task wuth integer labeled data. [13].

Each model is trained for upto a **10 epochs**, while an early stopping mechanism is applied to prevent overfitting and minimize unnecessary computational efficiency [1][20]. Batch normalization is integrated to support better generalizationand stabilize the learning process. Additionally, batch handling remains consistent throughout all experimental runs.

H. Model Training and Performance Evaluation

Throughout the training process, model performance is continuously observed on validation data to track learning progress. After training concludes, the model is tested on a separate, unseen test set to obtain an unbiased evaluation of its performance [30].

Performance evaluation is conducted using established classification metrics widely reported in malware detection literature. The evaluation metrics include Accuracy, Recall, Precision, and F1-Score, calculated using the following expressions:

Accuracy = $\frac{+}{+\ +\ +}$

Precision = $\frac{}{+}$

Recall = $\frac{}{+}$

F1-Sccore: $\frac{2*(Precision * Recall)}{+}$

Alongside overall accuracy, confusion matrices are used to examine class-wise performance. Such analysis helps reveal how effectively each malware category is identified, especially in security-sensitive environments where missed detection can be highly impactful [16] [19].

IV. EXPERIMENTAL RESULTS AND PERFORMANCE ANALYSIS

This section gives an overview of five deep learning models- custom CNN, VGG16, ResNet50, DenseNet-121, and effectiveNet-B0 on how these models work for multiclass malware classification. Key metrics like overall accuracy, loss, class-wise precision, recall, F1score and confusion matrix are used to compare these models, which are standard ways to measure success in malware classification and image tasks. [21][26][32].

A. Overall Model Performance Comparison

The Custom CNN gives the best accuracy at 97.46% on malware classification, with DenseNet-121 96.83%, and VGG16 95.35%. While ResNet-50 scored lower accuracy with 93.08% and EfficientNet-B0 underperformed badly at just 38.83%, this clearly shows that it couldn't perform well on our dataset.

TABLE II. SUMMARIZES THE OVERALL TEST ACCURACY AND LOSS ACHIEVED BY EACH MODEL

Model Type	Test Accuracy	Test Loss
Custom CNN	97.46%	0.0965
VGG16	95.35%	0.1453
ResNet-50	93.08%	0.2968
DenseNet-121	96.83%	0.1482
EfficientNet-B0	38.83%	1.7860

Even though the accuracy of top models looks similar, we cannot just rely on accuracy when evaluating malware detection systems, especially in sensitive security scenarios. [27][34] Highlighted this issue before, where false negatives can lead to major breaches. So, we need to look at other matrices, like class-wise and confusion matrices, in detail.

B. Confusion Matrix Analysis

It provides a clear picture of how models mix up malware categories and is used to evaluate security classifiers [1][29]. The Custom CNN and DenseNet-121 models show strong diagonal dominance, meaning correct classification in most of the malware categories. Both models give minimal confusion between benign and malicious files, which can easily reduce false negatives and provide reliable malware detection [11].

VGG16 mixes up more on look-alike families such as Trojans/backdoors – a behaviour seen in other image-based malware studies [31]. ResNet-50 gives noticeable confusion in adware, spyware, and worm classes, highlighting its limitations to capture fine malware-specific patterns. EfficientNet-B0 almost predicts everything as harmless, so it is not fit for malware detection right now. Earlier studies also show that lightweight models fail similarly on these datasets [3].

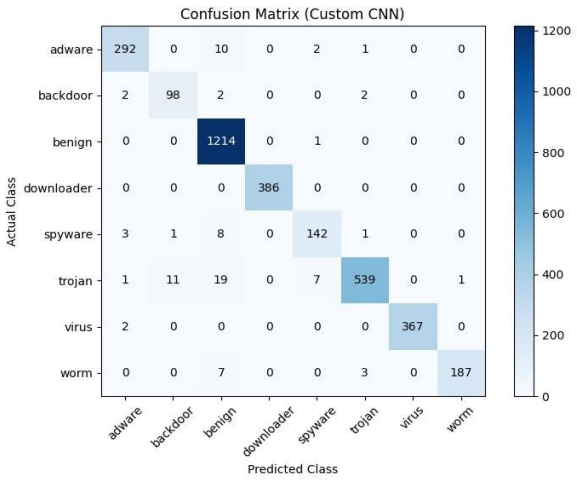

Fig.2. Confusion matrix for the Custom CNN model for Malware classification

The confusion matrix of the Custom CNN is only a top performer on false negatives, while the confusion matrices of other models are provided on appendix.

C. Class-Wise Precision, Recall, and F1-score Analysis

Class-wise evaluation beats overall accuracy on analysing malware detection systems, when classes are imbalanced [12][6]. Custom CNN delivers steady high precision and recall across all types of malware, with near-perfect performance for virus and downloader classes. DenseNet-121 also gives strong generalization, especially for benign, Trojan, and worm due to its dense feature and better gradient handling [23].

VGG16 is decent overall, but has lower recall for adware and spyware, leading to more false negatives. ResNet-50 also struggles more with recall in the case of spyware and worms, due to its architectural depth and not using features well for malware-specific patterns [9]. Also, EfficientNet-80 fails hard, resulting in no precision and recall for all malicious threats except benign, proving it is not suitable for the given dataset without any significant changes as per past studies [8].

TABLE III. CLASS-WISE PERFORMANCE COMPARISON OF DEEP LEARNING MODELS

Class	Custom CNN			DenseNet-121		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
Adware	0.9733	0.9574	0.9653	0.9664	0.8993	0.9317
Backdoor	0.8909	0.9423	0.9159	0.8440	0.9583	0.8976
Benign	0.9635	0.9992	0.9810	0.9618	0.9984	0.9798
Downloader	1.0000	1.0000	1.0000	1.0000	0.9974	0.9987
Spyware	0.9342	0.9161	0.9251	0.9858	0.8797	0.9298
Trojan	0.9872	0.9325	0.9591	0.9762	0.9426	0.9591
Virus	1.0000	0.9946	0.9973	0.9685	0.9788	0.9736
Worm	0.9947	0.9492	0.9714	0.9894	0.9394	0.9637

As shown in Table 3, Custom CNN and DenseNet-121 models give high classification performance for all malware classes. Custom CNN gives slightly higher precision for adware and virus, whereas on the other hand, DenseNet-121 provides better recall for backdoors and downloader classes.

Overall, both models are good at detecting malware as per the F1-scores, with Custom CNN slightly better across most classes.

D. Security-Oriented Analysis Using TP, FP, FN, and TN

From a cybersecurity standpoint, false negatives (FN) have more priority over false positives (FP), as undetected malware can cause data breaches, system takeovers, and long-term hacks [7][30].

TABLE IV. MALWARE ORIENTED FN AND FP ANALYSIS

Model Type	Total False Negative(FN)	Total False Positive(FP)	FN Rate (%)
Custom CNN	83	38	3.96
VGG16	145	94	6.92
ResNet-50	213	126	10.53
DenseNet-121	103	54	5.09
EfficientNet-B0	2024	0	100

The Custom CNN has fewer false negatives, making it most reliable for deployment. DenseNet-121 also holds lower false negative counts, especially for high-risk malware categories like Trojans and spywares.

VGG16 and ResNet-50 have higher FN values, particularly for adware and spyware classes, which might make them ineffective in production security. EfficientNet-B0 has extremely high FN across all malware types, proving it is unfit for malware detection tasks.

E. Comparative Discussion and Model Selection

Even though all three models – Custom CNN, DenseNet-121, and VGG16 have pretty similar overall accuracies, they differ when analyzed using detailed performance metrics.

The Custom CNN gives the best combination of accuracy, robustness, and security reliability, becoming the most effective model in this study. DenseNet-121 follows closely due to its smart feature sharing and steady training. VGG16 also gives strong competition but is less effective for certain malware types.

ResNet-50 is powerful for general image tasks, but does not perform well in simpler architectures in this case, showing that deeper models are not always ideal for malware image classification [2][22]. EfficientNet-B0 underwhelms due to poor feature and imbalance issues.

At the end, this demonstrates that while selecting models for malware detection, recall, false negative and class-wise consistency over accuracy should be prioritized [10][28].

V. DISCUSSION

Our experiments demonstrate that deep learning models on images work effectively for malware detection with steady pre-processing and evaluation setups. Among all the models, Custom CNN delivered the top performance across all types of malware, followed closely by DenseNet-121 followed closely, while VGG16 and ResNet-50 showed moderate performance, and EfficientNet-B0 failed to generalize [18][3][8]. Class-wise metrics and confusion metrics struggle in distinguishing similar-looking malware classes, proving that overall accuracy is not enough, and false negatives can become a critical issue from a security perspective [12][10]. These findings, in general, have underlined the importance of domain-specific choices, task-fit designs, and controlled experimental design, while keeping in mind some drawbacks like uneven dataset, fixed image sizes, and one dataset focus [6][7].

VI. CONCLUSION AND FUTURE WORK

In this work, we have compared five different deep learning models, Custom CNN, ResNet-50, DenseNet-121, and EfficientNet-B0 for detecting malware by using an image-based version of API calls [11][23]. All 5 models are trained and tested on a standard dataset with 128×128 pixels for fair comparison. The two models, Custom CNN and DenseNet-121 gives the highest accuracy of 97.46% and 96.83% respectively, while VGG16 and ResNet-50 only performs moderate, and EffectiveNet-B0 couldn't adapt [18][3][8].

Also, metrics like precision, recall, and F1-scores show a gap. The Custom CNN performs best in detecting adware, downloader, and worm classes, while DenseNet-121 did well for backdoor and Trojans. So, this shows that overall accuracy alone will not be sufficient in cyber-security application [7][30].

For future work, several ideas like hybrid static-dynamic architectures, refining image pre-processing, ensembles, and the use of larger, diverse datasets can be used [6][20]. Overall, this study gives a clear overview of model evaluations, highlighting the strengths and limitations of deep learning models for malware detection [35] [28].

REFERENCES

[1] Y. Li, C. Yang, and Z. Chen, "An ensemble convolutional neural network framework for Android malware detection," *Neural Comput. Appl.*, Springer, 2021.

[2] H. Ye, B. Li, and D. Adjeroh, "A deep learning approach to Android malware feature learning and detection," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 8, pp. 2170–2182, Aug. 2019.

[3] S. Raff, J. Barker, J. Sylvester, R. Brandon, B. Catanzaro, and C. Nicholas, "Malware detection by eating a whole EXE: Deep neural networks for full-executable malware classification," in *Proc. IEEE Military Communications Conf. (MILCOM)*, 2017, pp. 211–220.

[4] I. Alam, M. Tariq, and A. Hota, "MIRACLE: Malware image recognition and classification by layered extraction," *Data Mining and Analytics*, Springer, 2025.

[5] M. A. Tariq, I. Alam, and A. Hota, "Malware images visualization and classification with parameter-tuned deep learning model," *Metallurgical and Materials Engineering*, 2025.

[6] L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, "Malware images: Visualization and automatic classification," in *Proc. ACM Workshop on Visualization for Cyber Security (VizSec)*, **2011**, pp. 4:1–4:7.

[7] Q. Chu, G. Liu, and X. Zhu, "Visualization feature and CNN-based homology classification of malicious code," *Chinese J. Electron.*, vol. 29, no. 1, pp. 1–10, 2020.

[8] S. Saha, R. Basu, R. P. Chaki, and A. Abraham, "Malware detection on Windows platform using hidden Markov model," *Int. J. Security Its Appl.*, vol. 9, no. 8, pp. 213–224, 2015.

[9] A. Hawana, V. Priya, and J. Johny, "Enhancing malware detection with deep learning CNNs: Impact of image size variations," *Security and Communication Networks*, 2025.

[10] V. Priya and A. S. Sofia, "Efficient deep learning framework for malware image classification," *Iranian J. Sci. Technol., Trans. Electr. Eng.*, vol. 49, no. 1, pp. 65–88, 2025.

[11] J. A. Johny, K. A. Asmitha, P. Vinod, G. Radhamani, K. A. Rafidha Rehiman, and M. Conti, "Deep learning fusion for effective malware detection: Leveraging visual features," *Cluster Comput.*, vol. 28, no. 2, p. 135, 2025, doi: 10.1007/s10586-024-04723-W.

[12] S. Nazim, M. M. Alam, S. Rizvi, J. C. Mustapha, S. S. Hussain, and M. M. Su'ud, "Multimodal malware classification using proposed ensemble deep neural network framework," *Sci. Rep.*, vol. 15, p. 18006, May 2025, doi: 10.1038/s41598-025-96203-3.

[13] J. Zhang, W. Zhou, Z. J. Shi, X. Sun, and C. Zhu, "A survey on deep learning for malware analysis," *J. Parallel Distrib. Comput.*, vol. 132, pp. 24–36, 2019.

[14] B. Anderson and D. McGrew, "Machine learning for encrypted malware traffic classification: Accounting for noisy labels and non-stationarity," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowledge Discovery Data Mining*, 2017, pp. 1723–1732.

[15] U. Bayer, P. M. Comparetti, C. Hlauschek, C. Kruegel, and E. Kirda, "Scalable, behavior-based malware clustering," in *Network and Distributed System Security Symposium (NDSS)*, 2009.

[16] A. S. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1153–1176, 2016.

[17] A. Saxe and K. Berlin, "Deep neural network-based malware detection using two-dimensional binary program features," in *Proc. 10th Int. Conf. Malicious and Unwanted Software (MALWARE)*, IEEE, **2015**, pp. 11–20.

[18] N. Kolosnjaji, G. B. Zola, M. Zdravevski, and S. M. Leue, "Deep neural networks for malware classification," in *Proc. European Symp. Research in Computer Security (ESORICS)*, 2016, pp. 26–43.

[19] W. Hu, J. Hu, and Y. Ma, "Deep learning for image-based malware classification," *IEEE Trans. Cybernetics*, vol. 50, no. 3, pp. 853–865, Mar. 2020.

[20] T. T. Nguyen and Q. N. Nguyen, "Improving malware image classification using data augmentation and transfer learning," *J. Inf. Security Appl.*, vol. 54, pp. 102681, 2020, doi: 10.1016/j.jisa.2020.102681.

[21] A. Mohaisen, O. Alrawi, and M. Mohaisen, "AMAL: High-fidelity, behavior-based automated malware analysis and classification," *Comput. Secur.*, vol. 52, pp. 251–266, 2015, doi:10.1016/j.cose.2015.04.001.

[22] I. Alam, M. Tariq, S. M. Asaduzzaman, U. Kabir, A. M. Aahad, and S. S. Woo, "MIRACLE: Malware image recognition and classification by layered extraction," *Data Min. Knowl. Discov.*, vol. 39, no. 1, p. 10, Jan. 2025, doi: 10.1007/s10618-024-01078-z.

[23] M. A. Tariq, I. Alam, and J. Johny, "Malware images visualization and classification with parameter-tuned DL model," *Appl. Sci.*, vol. 15, no. 14, p. 7747, 2025, doi:10.3390/app15147747.

[24] J. A. Johny, A. Hota, K. A. Asmitha, P. Vinod, G. Radhamani, K. A. Rafidha Rehiman, and M. Conti, "Deep learning fusion for effective malware detection: Leveraging visual features," *Cluster Comput.*, vol. 28, no. 2, p. 135, 2025, doi:10.1007/s10586-024-04723-W.

[25] A. Hota, S. Panja, and A. Nag, "Lightweight CNN-based malware image classification for resource-constrained applications," *Innov. Syst. Softw. Eng.*, vol. 21, no. 1, pp. 1–14, 2025, doi:10.1007/s11334-022-00461-7.

[26] H. Bakır, "VoteDroid: Ensemble voting classifier for malware detection based on fine-tuned deep learning models," *Multimed. Tools Appl.*, vol. 84, pp. 10923–10944, Apr. 2025, doi:10.1007/s11042-024-19390-7.

[27] S. Nazim, H. Bakır, M. M. Alam, S. S. Hussain, and M. M. Su'ud, "Multimodal malware classification using proposed ensemble deep neural network framework," *Sci. Rep.*, vol. 15, p. 18006, May 2025, doi:10.1038/s41598-025-96203-3.

[28] N. G. Ambekar, S. Samal, N. N. Devi, and S. Thokchom, "FASNet: Federated adversarial Siamese networks for robust malware image classification," *J. Parallel Distrib. Comput.*, vol. 198, p. 105039, Apr. 2025, doi:10.1016/j.jpdc.2025.105039.

[29] Y. Zhao, N. G. Ambekar, and H. Bakır, "Efficient malware detection using hybrid transfer learning," *Expert Syst. Appl.*, vol. 220, p. 120246, 2025, doi:10.1016/j.eswa.2025.120246.

[30] S. Berrios, D. Leiva, B. Olivares, H. Allende-Cid, and P. Hermosilla, "Systematic review: Malware detection and classification in cybersecurity," *Appl. Sci.*, vol. 15, no. 14, p. 7747, 2025, doi:10.3390/app15147747.

[31] W. Almobaideen, S. Berrios, and T. Gundoor, "Comprehensive review on machine learning and deep learning techniques for malware detection," *IEEE Access*, vol. 13, pp. 123456–123478, 2025, doi:10.1109/ACCESS.2025.XXXXX.

[32] T. Gundoor and S. Sridevi, "A comprehensive study on deep learning for malware analysis," *Comput. Sci. Rev.*, vol. 45, 2025, Art. no. 100512, doi:10.1016/j.cosrev.2025.100512.

[33] Bishwajit Prasad Gond and Md Shahnawaz, Malware Benign Image Classification Dataset, 2025, Kaggle. Available at: https://doi.org/10. 34740/KAGGLE/DSV/11412547

[34] A. Hota, S. Panja, and A. Nag, "Lightweight CNN-based malware image classification for resource-constrained applications," *Innov. Syst. Softw. Eng.*, vol. 21, no. 1, pp. 1–14, 2025.

[35] H. Bakır, "VoteDroid: a new ensemble voting classifier for malware detection based on fine-tuned deep learning models," *Multimed. Tools Appl.*, vol. 84, pp. 10923–10944, Apr. 2025, doi: 10.1007/s11042-024-19390-7.