University of Piraeus
School of Information and Communication Technologies
Department of Digital Systems

Postgraduate Program of Studies
MSc Cybersecurity and AI Technologies

Master's Thesis

Title: **ARTIFICIAL INTELLIGENCE FOR MALWARE DETECTION IN THE MEDICAL INTERNET OF THINGS.**

Supervisor Professor: **Christos Xenakis**

Name-Surname	E-mail	Student ID.
Stylianos Batzakas	Stelios_mpatzakas@ssl-unipi.gr	mte24021

Piraeus
30/03/2026

Dedication

In memory of Andreas and Takis.

Abstract

The rapid proliferation of Internet of Things (IoT) devices in healthcare environments has introduced new security challenges, particularly in the detection of malware targeting medical IoT systems. Traditional signature-based antivirus solutions are often ineffective against evolving and obfuscated malware, necessitating the adoption of intelligent detection mechanisms. This thesis presents a deep learning-based framework for malware detection in Linux-based medical IoT devices by transforming binary files into image representations. Both grayscale and RGB image formats are investigated, enabling a comparative analysis of their effectiveness in malware classification tasks. Convolutional Neural Networks (CNNs), including ResNet-18 and EfficientNet-B0, are employed to automatically extract discriminative features from the generated images. Experimental results demonstrate that grayscale representations achieve competitive and, in some cases, superior performance compared to RGB images, while also offering reduced computational complexity. Furthermore, the robustness of the proposed models was evaluated under adversarial conditions using FGSM perturbations, with GAN-based adversarial attacks mentioned as a potential future avenue for more complex robustness testing.

Overall, this work contributes to the growing field of AI-driven cybersecurity for medical IoT by providing a scalable and effective malware detection approach, along with insights into representation choices and model robustness.

Keywords: Medical IoT, malware detection, deep learning, grayscale vs RGB, adversarial robustness, edge/embedded efficiency.

Περίληψη

Η ραγδαία εξάπλωση των συσκευών Internet of Things (IoT) σε περιβάλλοντα Medical IoT έχει εισαγάγει νέες προκλήσεις ασφάλειας, ιδιαίτερα στον εντοπισμό κακόβουλου λογισμικού που στοχεύει ιατρικά συστήματα IoT. Οι παραδοσιακές λύσεις antivirus που βασίζονται σε υπογραφές αποδεικνύονται συχνά ανεπαρκείς έναντι εξελισσόμενων και συγκαλυμμένων μορφών κακόβουλου λογισμικού, καθιστώντας αναγκαία την υιοθέτηση ευφυών μηχανισμών ανίχνευσης.

Η παρούσα διπλωματική εργασία παρουσιάζει ένα πλαίσιο ανίχνευσης κακόβουλου λογισμικού βασισμένο σε τεχνικές βαθιάς μάθησης (Deep Learning) για Linux-based ιατρικές συσκευές IoT, μέσω μετασχηματισμού δυαδικών αρχείων σε αναπαραστάσεις εικόνας. Μελετώνται τόσο εικόνες κλίμακας του γκρι όσο και έγχρωμες (RGB), επιτρέποντας συγκριτική ανάλυση της αποτελεσματικότητάς τους σε εργασίες ταξινόμησης κακόβουλου λογισμικού. Συνελικτικά Νευρωνικά Δίκτυα (CNNs), όπως τα ResNet-18 και EfficientNet-B0, χρησιμοποιούνται για την αυτόματη εξαγωγή διακριτικών χαρακτηριστικών από τις παραγόμενες εικόνες.

Τα πειραματικά αποτελέσματα δείχνουν ότι οι αναπαραστάσεις κλίμακας του γκρι επιτυγχάνουν ανταγωνιστική και, σε ορισμένες περιπτώσεις, ανώτερη απόδοση σε σύγκριση με τις RGB εικόνες, ενώ παράλληλα προσφέρουν μειωμένη υπολογιστική πολυπλοκότητα. Επιπλέον, η ανθεκτικότητα των προτεινόμενων μοντέλων αξιολογήθηκε υπό επιθέσεις αντιπαραθετικών παραδειγμάτων (adversarial attacks) με τη χρήση της μεθόδου FGSM, ενώ οι επιθέσεις βασισμένες σε GAN αναφέρονται ως μελλοντική κατεύθυνση για πιο σύνθετη αξιολόγηση ανθεκτικότητας.

Συνολικά, η εργασία αυτή συμβάλλει στον αναπτυσσόμενο τομέα της κυβερνοασφάλειας με χρήση τεχνητής νοημοσύνης για ιατρικά συστήματα IoT, προτείνοντας μια επεκτάσιμη και αποτελεσματική προσέγγιση ανίχνευσης κακόβουλου λογισμικού, καθώς και παρέχοντας χρήσιμες επισημάνσεις σχετικά με την επιλογή αναπαράστασης και την ανθεκτικότητα των μοντέλων.

List of Abbreviations

AI — Artificial Intelligence

AV — Antivirus

CNN — Convolutional Neural Network

RNN — Recurrent Neural Network

IoT — Internet of Things

IIoT — Industrial Internet of Things

DL — Deep Learning

ML — Machine Learning

GAN — Generative Adversarial Network

FGSM — Fast Gradient Sign Method

RGB — Red, Green, Blue (color image representation)

CPU — Central Processing Unit

GPU — Graphics Processing Unit

API — Application Programming Interface

OS — Operating System

F1-score — Harmonic Mean of Precision and Recall

TP — True Positive

TN — True Negative

FP — False Positive

FN — False Negative

ReLU — Rectified Linear Unit

SGD — Stochastic Gradient Descent

Table of Contents

1 Introduction

The rapid advancement of the Internet of Things (IoT) has led to its widespread adoption in healthcare environments, giving rise to the concept of medical IoT systems. These systems include interconnected medical devices such as patient monitors, infusion pumps, and wearable sensors, which enable real-time monitoring and improved patient care. However, this increased connectivity also introduces significant security vulnerabilities, making medical IoT devices attractive targets for cyberattacks [1].

One of the most critical threats in this domain is malware specifically designed to exploit the limited computational resources and security mechanisms of IoT devices. Traditional antivirus (AV) solutions, which rely primarily on signature-based detection, struggle to identify new and evolving malware variants, especially those employing obfuscation and polymorphic techniques [2].

To address these limitations, recent research has explored the application of Artificial Intelligence (AI) and Deep Learning (DL) techniques for malware detection. In particular, transforming binary malware samples into image representations has emerged as a promising approach, allowing the use of powerful Convolutional Neural Networks (CNNs) for automated feature extraction and classification [6][16].

This thesis presents a deep learning-based framework for detecting malware in Linux-based medical IoT systems, using the Malimg dataset, which is typically associated with Windows malware binaries. The analysis conducted focuses on translating malware into image representations to evaluate performance across grayscale and RGB formats. Furthermore, the study evaluates the performance of modern CNN architectures, including ResNet-18 and EfficientNet-B0, and examines model robustness under adversarial conditions.

The main contributions of this work can be summarized as follows:

1. A comprehensive evaluation of malware classification using grayscale and RGB image representations.

2. A comparative analysis of deep learning architectures for medical IoT malware detection.
3. An assessment of model robustness against adversarial perturbations.
4. Insights into the trade-offs between classification performance and computational cost.

The remainder of this thesis is organized as follows. Chapter 2 reviews related work in IoT security and deep learning-based malware detection. Chapter 3 describes the proposed methodology and experimental setup. Chapter 4 presents the experimental results and analysis. Finally, Chapter 5 concludes the thesis and outlines directions for future work.

1.1 Background and Motivation

Medical IoT context.

MIoT devices differ significantly from traditional computing systems in terms of architecture, operational constraints, and lifecycle management. These systems often operate on trimmed Linux kernels with vendor-specific drivers and limited computational resources, including restricted memory capacity and storage [2]. Moreover, MIoT devices are typically deployed for long operational lifetimes and receive infrequent security updates due to regulatory and operational constraints [1]. As a result, vulnerabilities may persist for extended periods, increasing the risk of compromise in critical healthcare infrastructures.

Devices frequently lack hardware isolation mechanisms common in general-purpose computing systems and must maintain high availability, making security interventions more challenging. Compromise can therefore persist undetected and propagate laterally across interconnected systems, with consequences ranging from privacy violations to disruption of clinical services [6][16]. Recent studies further highlight that similar security challenges extend to smart environments, where interconnected devices introduce additional risks and attack surfaces [25].

Threat landscape.

IoT-targeting malware families have demonstrated rapid propagation through weak

credentials, unprotected interfaces, supply-chain tampering, and exploitation of unpatched vulnerabilities [6][16]. Once deployed, malware can enroll devices into botnets, deploy cryptominers, or serve as an entry point for broader network intrusions. In healthcare environments, such attacks may disrupt clinical operations or be leveraged for extortion [1]. Recent research confirms that Linux-based IoT malware is an increasingly prevalent threat, with new variants continuously emerging to exploit these systemic weaknesses [9]. Furthermore, the expansion of AI-driven cybersecurity solutions into critical infrastructures underscores the growing need for resilient and secure system architectures across diverse application domains [26].

Limitations of conventional detection.

Signature-based antivirus solutions provide high precision for known threats but are ineffective against polymorphic malware, packing techniques, and adversarial obfuscation [4]. Static feature engineering approaches, such as opcode frequency analysis or imported symbol inspection, require domain expertise and often lack generalizability across architectures and compiler variations [4]. Dynamic analysis offers deeper behavioral insights but remains computationally expensive, difficult to scale, and potentially unsafe to deploy within sensitive healthcare environments [5]. These limitations motivate the exploration of more scalable and adaptive malware detection methodologies [10]. Recent studies emphasize the importance of integrated adversarial defense frameworks and real-time detection mechanisms [22][23][27].

Motivation for image-based deep learning.

Transforming binary files into image representations enables the application of computer vision techniques to malware detection. By mapping byte values to pixel intensities, this approach preserves local byte relationships and structural patterns within executables. This enables:

- End-to-end feature learning using convolutional neural networks (CNNs) and related architectures that excel at spatial pattern recognition [7][8][16].
- Format-agnostic processing, eliminating the need for manual parsing of diverse executable formats and firmware structures [16].
- Efficient training and deployment, as models can operate on standardized image inputs and be optimized for resource-constrained environments [7].

Despite these advantages, several challenges remain. The choice of representation is still under investigation: grayscale images provide compact and direct encoding, while RGB representations offer increased expressiveness at the cost of higher computational requirements [7][8][16]. Model selection must also balance classification performance with deployability in constrained MIoT environments [2][9]. Additionally, robustness is a critical concern, as deep learning models are vulnerable to adversarial manipulation. Prior work has demonstrated that GAN-based perturbations can successfully evade even advanced malware classifiers [11][12][13], while recent studies emphasize the importance of integrated adversarial defense frameworks and real-time detection mechanisms [22][27][28].

Research motivation.

Against this backdrop, healthcare environments require malware detection systems that are (i) accurate for Linux-based threats prevalent in MIoT ecosystems [9], (ii) efficient enough for deployment on edge devices or near-device gateways [2], and (iii) resilient to adversarial evasion techniques [11][12][13]. Additionally, emerging approaches explore the integration of cybersecurity with distributed technologies such as blockchain to enhance trust, transparency, and risk management in complex digital ecosystems. Recent advancements highlight the integration of blockchain-based solutions to improve transparency and trust in cybersecurity [21][26]

This thesis addresses these challenges by establishing a unified pipeline for grayscale and RGB malware image representations, benchmarking multiple deep learning architectures, evaluating computational efficiency, and assessing robustness under adversarial conditions[11][12][13]. The findings aim to support the design of practical and secure AI-based malware detection systems tailored to real-world MIoT constraints.

Paper	Approach	Algorithm/Model	Tool/Framework	Accuracy
Nataraj et al. 2011 (Malimg) [16]	Grayscale image conversion	GIST descriptors + kNN	Custom feature extraction	97.18% in 25 malware families
Bozkir et al. 2019 (MaleVis) [17]	RGB image conversion	CNNs: AlexNet, VGG, ResNet, Inception, DenseNet	PyTorch, Caffe, bin2png (Python script), OpenCV	DenseNet 97.48%, ResNet18 97.18%
Aslan & Yilmaz 2021 [33]	Grayscale images + transfer learning	Hybrid CNN with two pre-trained models	Transfer learning	94.9–97.8% (Malimg, MaleVis, Microsoft BIG)
Awan et al. 2021 Cnn [37]	Malimg grayscale	CNN + VGG19 (feature selection + frozen layers)	PyTorch/TensorFlow	97.68%
O'Shaughnessy & Sheridan 2022[38]	Obfuscated malware → grayscale	Space-filling curves + CNN	Custom converter	97.6%
Seneviratne et al. 2022 (SHERLOCK) [39]	Grayscale → Vision Transformer	ViT	Self-supervised learning	97%
Vasan et al. 2024 (IMCBL)[35]	Lightweight alt. to DL	IMCBL (non-DL)	CPU-only, no GPU	97.64%
This thesis (Batzakas) 2026	**MIoT-motivated malware-image benchmark on Malimg**	**ResNet-18, EfficientNet-B0, DeiT-Tiny, Autoencoder, Row-LSTM; FGSM robustness evaluation**	**PyTorch**	**EfficientNet-B0 99.43% (grayscale) grayscale outperformed replicated 3-channel input. DeiT-Tiny showed the strongest FGSM robustness.**

Overall, prior work shows that malware-as-images methods can achieve high classification accuracy, but most evaluations are Windows-centric and focus mainly on clean-set performance. In this thesis, experiments on the Malimg benchmark showed that single-channel grayscale inputs consistently matched or outperformed replicated three-channel inputs. EfficientNet-B0 achieved the best clean performance, while DeiT-Tiny showed better resistance under FGSM perturbations, highlighting a trade-off between clean accuracy, robustness, and efficiency.

Deep Learning Algorithms for Malware-as-Images

Several families of deep learning architectures have been applied to malware-as-image classification. Convolutional Neural Networks (CNNs) are the most widely used, achieving high accuracy by leveraging spatial texture patterns of malware images [7][8][16]. Transfer learning and ensemble techniques reduce training time and further improve accuracy by combining the outputs of multiple pre-trained models [15]. More recently, Vision Transformers (ViT) have been introduced as scalable alternatives for large malware image corpora [14]. Autoencoder-based models have also been explored to compress high-dimensional features and to support anomaly detection.

These approaches provide the methodological foundation for this thesis, which benchmarks CNNs, ensembles, autoencoders, and transformers across grayscale and RGB representations.

1.2 Problem Statement

Despite significant advances in deep learning–based malware detection, several challenges remain unresolved in the context of medical IoT environments. Existing research has primarily focused on general-purpose computing platforms or Windows-based malware datasets, leaving critical gaps in the understanding of Linux-targeted malware that dominates IoT and MIoT ecosystems..

Lack of Linux-targeted research. The majority of prior studies have focused on **Windows Portable Executable (PE) malware** or on generic IoT datasets [7][8][16]. However, MIoT devices predominantly run Linux-based firmware, making Windows-oriented solutions less directly applicable. Although recent work has begun to classify Linux IoT malware variants [9], the literature in this area remains sparse compared to research on Windows malware. This creates a gap between existing studies and the real-world security requirements of MIoT systems.

Unresolved image representation trade-offs. Malware-as-image approaches have gained attention due to their ability to transform binaries into visual structures that are amenable to convolutional and related models [7][8][16]. Yet, a systematic analysis of **grayscale versus RGB image encodings** is still missing. Grayscale images are compact and computationally efficient, whereas RGB encodings may capture richer byte-level relationships but require more memory and compute power. In resource-constrained MIoT environments, where devices and gateways operate with limited hardware capacity, this trade-off is critical [2][9].

Adversarial robustness is underexplored. Deep-learning models, while powerful, are susceptible to **adversarial examples**: carefully perturbed inputs that preserve malware functionality but deceive classifiers. Techniques such as **Generative Adversarial Networks (GANs)** have already demonstrated the ability to generate malware samples capable of bypassing detection [11][12][13]. However, few malware detection frameworks include systematic evaluation of robustness under such conditions, leaving real deployments vulnerable to evasion. Recent studies also demonstrate that adversarial attacks extend beyond image-based malware classification to broader AI-driven cybersecurity systems, highlighting the need for adaptive and platform-agnostic defense mechanisms [22][23][29].

Efficiency and practical constraints. Beyond accuracy, malware detection methods must be evaluated against **memory, latency, and computational constraints**. These considerations are especially relevant in MIoT deployments where edge or near-device analysis is preferred to reduce cloud dependency and latency [2][9][10]. Nevertheless, most existing studies emphasize accuracy metrics while neglecting performance trade-offs that determine feasibility in practice.

Threat Model and System Requirements

The considered threat model assumes that adversaries can deliver malware binaries to MIoT devices through various attack vectors, including exploitation of unpatched vulnerabilities, weak or default credentials, and supply-chain tampering [3][9]. Once executed, such malware can compromise device integrity, exfiltrate sensitive health information, or enroll devices into botnets. The defense goal of this thesis is to detect malicious binaries before they are executed on MIoT devices or edge gateways.

To achieve this, system requirements must be explicitly considered. Malware detection solutions for MIoT must be lightweight, with low memory footprint, reduced inference latency, and minimal computational overhead, in order to remain compatible with embedded Linux-based platforms [2][9]. At the same time, they must ensure sufficiently high detection accuracy to meet healthcare security and privacy standards. These requirements guide the evaluation of proposed image-based deep learning approaches throughout this thesis.

Summary of open issues. The current body of work leaves several questions unanswered:

1. Limited focus on **Linux-based malware** relevant to MIoT environments.

2. Lack of a **systematic comparison of grayscale versus RGB image encodings**.
3. Insufficient evaluation of **adversarial robustness** in malware image classification.
4. Minimal discussion of **efficiency trade-offs** in resource-constrained deployments.

This thesis addresses these challenges by conducting a comprehensive study of malware image detection tailored for medical IoT. Specifically, it investigates representation trade-offs, benchmarks multiple deep-learning models, evaluates adversarial robustness, and analyzes cost–benefit aspects related to accuracy, efficiency, and practicality.

1.3 Research Objectives

The overarching goal of this thesis is to design and evaluate deep learning–based malware detection methods tailored to Linux-based medical IoT environments using the malware-as-images paradigm.

Dataset pipeline. A primary objective is to establish a reproducible pipeline for malware dataset preparation. This includes the transformation of binaries into grayscale and RGB images using benchmark datasets such as Malimg [16] and MaleVis [17], while also incorporating more recent resources such as Dumpware10 [18]. The pipeline ensures comparability across different models and modalities, addressing reproducibility gaps noted in recent surveys [19][20].

Grayscale versus RGB trade-offs. A second objective is to conduct a systematic comparison between grayscale and RGB malware image encodings. Previous works demonstrated strong results for both representations but typically restricted evaluation to one modality [7][8][16]. Grayscale images are compact and efficient, while RGB encodings offer richer structural information at the cost of higher computational and memory demands [2][9][17]. This thesis aims to quantify these trade-offs in terms of accuracy, latency, and resource requirements, with a focus on their impact in MIoT contexts.

Model benchmarking. A third objective is to benchmark state-of-the-art learning architectures. Convolutional neural networks (CNNs) have shown strong performance in malware classification [8], but emerging alternatives such as hybrid CNN–autoencoder models, transformer-based architectures [14], and ensemble or transfer learning methods [15] remain underexplored in the MIoT setting. The benchmarking will compare these models under uniform protocols to assess their strengths and weaknesses.

Adversarial robustness. A fourth objective is to evaluate the resilience of malware detection models against adversarial perturbations. Deep-learning classifiers are vulnerable to attacks that preserve malware functionality while inducing misclassification [11][12]. GAN-based and visualization-aware adversarial generation techniques have already proven effective in evading detection [13]. The thesis will therefore assess robustness across grayscale and RGB modalities under such

conditions [19][20].This aligns with recent research directions focusing on adversarially robust AI systems and real-time defensive frameworks in cybersecurity applications [27][28].

Cost–benefit framework. A final objective is to develop a cost–benefit framework that links detection performance to computational efficiency. Prior research has shown that high-accuracy models often demand extensive resources and training times, which is unsuitable for MIoT deployment, where inference may need to occur at the edge [2][9]. By mapping trade-offs between model accuracy, robustness, and efficiency, the thesis aims to provide actionable guidance for deploying lightweight versus resource-intensive models in real-world MIoT environments.

1.4 Research Questions

Based on the background, problem statement, and objectives, this thesis seeks to address the following research questions, which are directly linked to the identified challenges and requirements:

1. **RQ1: How effective are grayscale versus RGB malware images for classification?**
   This tackles the unresolved trade-off in malware representation [16][17].

2. **RQ2: Which deep learning models achieve the best accuracy and robustness for Linux-based malware detection in MIoT contexts?**
   This addresses the lack of systematic evaluation of different models under MIoT constraints [8][14][15].

3. **RQ3: How vulnerable are malware image classifiers to adversarial attacks generated by GANs and other perturbation methods?**
   This corresponds to the underexplored problem of resilience against evasion attempts [11][12][13].

4. **RQ4: What are the trade-offs between detection accuracy and computational/memory efficiency?**
   This links directly to the system requirements of lightweight and resource-aware detection for embedded MIoT platforms [2][9].

By aligning the research questions with the open challenges and the system-level requirements, the thesis ensures that the proposed methodology is both scientifically rigorous and practically relevant for MIoT malware detection.

1.5 Contributions

This thesis makes the following contributions:

- **Comparative study of malware image representations.** By systematically evaluating grayscale and RGB encodings, the thesis provides insight into their relative strengths, weaknesses, and deployment suitability [16][17].
- **Benchmarking of multiple deep learning architectures.** CNNs, autoencoder-based models, transformers, and ensemble approaches are tested under consistent protocols to identify optimal designs for MIoT malware detection [8][14][15].
- **Evaluation of adversarial robustness.** Models are evaluated under FGSM-based perturbations, while GAN-based and visualization-aware attacks are discussed as future work. [11][12][13].
- **Cost–benefit analysis for MIoT deployments.** By quantifying accuracy, latency, and resource usage, the thesis delivers practical guidelines for selecting models that balance performance with feasibility in constrained environments [2][9][10].

1.6 Limitations

Despite the contributions of this study, several limitations must be acknowledged.

First, the evaluation relies mainly on publicly available benchmark datasets such as Malimg [16] and MaleVis [17]. While widely adopted in the literature, these datasets may not fully capture the diversity of Linux-based malware that targets Medical IoT devices.

Second, the adversarial robustness analysis is restricted to FGSM-based gradient perturbations [11][12][13]. Other evasion strategies, such as advanced obfuscation or polymorphic engines beyond section packing, are out of scope.

Finally, efficiency is evaluated through benchmark experiments on general-purpose hardware, and not through deployment on actual MIoT devices. As a result, performance estimates may differ in real-world embedded environments.

1.7 Thesis Structure

The remainder of this thesis is organized as follows:

- **Section 1: Introduction** – Provides the background and motivation, defines the problem, outlines research objectives, poses research questions, and states the contributions.

- **Section 2: Related Work** – Surveys the state of the art in malware detection, with emphasis on malware-as-images approaches, existing datasets, and related challenges.
- **Section 3: Methodology** – Describes the dataset preparation, grayscale and RGB conversion techniques, chosen models, adversarial evaluation setup, and metrics.
- **Section 4: Evaluation**– Presents the experimental design, training and testing procedures, and technical implementation details.
- **Section 5: Discussion** – Analyzes the experimental results, compares modalities and models, and discusses adversarial robustness and efficiency trade-offs.
- **Section 6: Conclusion and Future Work** – Summarizes the findings, highlights the contributions, and outlines future directions for research in MIoT malware detection.

2 Related Work

This chapter reviews the state of the art in malware detection research, focusing on techniques relevant to malware-as-images classification and security challenges in Medical Internet-of-Things (MIoT) environments. The review begins with traditional malware analysis approaches and gradually moves toward modern deep learning–based detection methods. Particular emphasis is placed on image-based malware representations, deep learning architectures, available datasets, and adversarial machine learning threats that affect the reliability of AI-based malware detectors.

2.1 Overview of Malware Analysis Approaches

Malware analysis encompasses the techniques and methodologies used to understand the structure and behavior of malicious software. Traditional approaches are generally categorized into static analysis, dynamic analysis, and machine learning–based methods, each offering distinct advantages and limitations. These trade-offs become particularly significant in resource-constrained environments such as Medical Internet-of-Things (MIoT) devices, where computational efficiency, safety, and robustness are critical operational requirements.

Malware analysis techniques can be broadly categorized into static, dynamic, and machine learning–based approaches. The general taxonomy of malware analysis methods is illustrated in Figure 2.1.

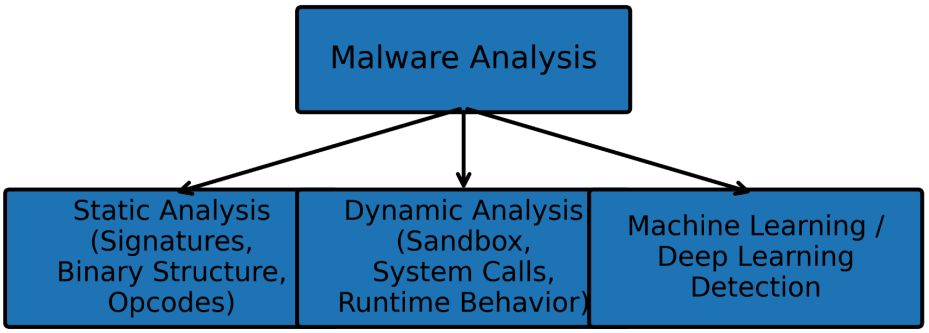

Figure 2.1: Taxonomy of malware analysis approaches including static analysis, dynamic analysis, and machine learning–based detection techniques.

Static analysis examines software binaries without executing them. Traditional static techniques analyze structural characteristics such as opcode sequences, import tables, header metadata, strings, and embedded resources. Signature-based antivirus systems have historically relied on static analysis by matching known byte patterns against previously identified malware families. While static approaches are computationally efficient and accurate for known threats, their effectiveness decreases significantly when malware employs obfuscation, packing techniques, or polymorphic transformations designed to conceal malicious functionality [4].

Dynamic analysis involves executing suspicious binaries in controlled environments such as sandboxes, virtual machines, or honeypots while monitoring runtime behavior. Behavioral indicators such as system calls, network activity, file system changes, and process interactions are recorded during execution. Dynamic analysis can reveal hidden malicious behaviors and detect previously unseen threats that static analysis may fail to identify [5]. However, this approach introduces substantial computational overhead and operational risk, particularly in embedded environments where virtualization and instrumentation capabilities may be limited.

Machine Learning and Deep Learning: As malware has grown increasingly modular, evasive, and obfuscated, machine learning (ML) and deep learning (DL) approaches have emerged as powerful alternatives. Early ML-based systems relied on handcrafted features, such as opcode frequencies, API call statistics, or n-gram byte patterns [4] . Although these models achieved reasonable performance, they were sensitive to feature selection and struggled with generalization across diverse architectures and compiler settings.

Deep learning, by contrast, enables end-to-end feature extraction directly from raw or transformed binary data. Convolutional neural networks (CNNs) have demonstrated strong performance in treating malware binaries as images, identifying spatial patterns that correlate with malware families without manual feature engineering [7],[8]. Recent surveys confirm that DL techniques now dominate the landscape of malware detection research, owing to their ability to learn expressive representations from large datasets [10] .

Nevertheless, DL methods introduce their own challenges since they require substantial computational resources for training and inference, their robustness is limited, as they are vulnerable to adversarial perturbations and finally their deployment on MIoT devices must consider memory footprint, latency, and energy consumption. The vulnerability of deep learning models to adversarial manipulations remains an open issue for MIoT deployment [11][12][13][29].

These limitations motivate a shift toward lightweight, efficient, and resilient DL-based solutions, particularly those based on image representations of malware, which form the core focus of this thesis.

Given the limitations of traditional analysis techniques, recent research has increasingly explored machine learning and deep learning methods that automatically learn discriminative patterns from malware artifacts. These approaches enable scalable detection without relying on handcrafted features and form the foundation of modern malware detection systems.

2.2 Malware-as-Images: Foundational Work

Transforming malware binaries into images has emerged as a promising direction in modern malware detection research. Instead of relying on manually engineered features, malware-as-images approaches encode raw binary data as visual representations that can be processed using computer vision models. In this paradigm, byte values are mapped directly to pixel intensities, enabling deep learning architectures to automatically extract structural patterns from malware samples.

Binary-to-Image Conversion Paradigm

The conceptual foundation of malware image analysis was introduced by Nataraj et al. (2011), who proposed converting executable binaries into grayscale images for texture-based malware classification [16]. In this approach, the binary file is interpreted as a sequence of byte values ranging from 0 to 255, which are then reshaped into a two-dimensional matrix where each byte corresponds to a pixel intensity. This representation preserves local structural relationships within the binary and allows machine learning models to identify recurring visual patterns associated with specific malware families.

By framing malware detection as an image recognition task, this paradigm eliminates the need for handcrafted features and enables the use of convolutional architectures that excel at identifying hierarchical spatial patterns. Importantly, it is **format-agnostic**, making it suitable for diverse executable formats, including Linux-based binaries that dominate MIoT ecosystems.

The original Malimg dataset [16] became a widely used benchmark in subsequent research, cementing the role of malware images in the literature.

In malware-as-images approaches, executable binaries are transformed into visual representations that can be processed by computer vision models. The general workflow of this process is illustrated in Figure 2.2.

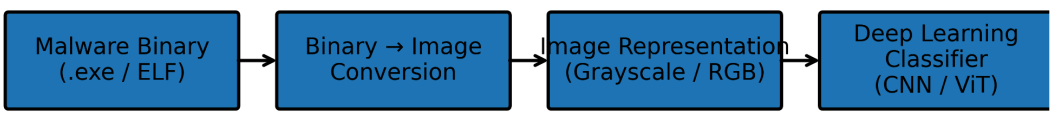

Figure 2.2: Malware-as-images detection pipeline. Executable binaries are converted into image representations and then classified using deep learning models.

Grayscale Image Representations

Grayscale representations treat the binary file as an 8-bit image with a single channel. This approach remains the most commonly used method in the field due to its simplicity and efficiency. Studies such as Kalash et al. (2018) [7] and Gibert et al. (2020) [8] demonstrate that CNNs operating on grayscale malware images achieve high accuracy across diverse datasets.

Key advantages include:

1. **Low memory footprint:** 1 channel vs. 3 channels in RGB
2. **Reduced computational cost:** fewer parameters and smaller input tensors
3. **Strong performance for structural malware patterns**
4. **Suitability for embedded and MIoT scenarios**, where resource constraints are a major concern.

Grayscale encoding remains a strong baseline and is frequently preferred in edge-oriented detection pipelines where model size and inference time are critical considerations.

RGB Image Representations

To enhance representational richness, later works explored **RGB encodings**, which map the byte stream into three color channels instead of one. Bozkir et al. (2019), through the MaleVis dataset [17] , systematically evaluated CNN architectures on RGB malware images and demonstrated that deeper models such as DenseNet and ResNet benefit from the additional channel information.

RGB images may implicitly capture:

A. multi-scale correlations across bytes,
B. high-frequency artifacts from packing or encryption,
C. more diverse visual textures.

Because RGB encodings increase dimensionality, they typically yield stronger performance in large-scale CNNs, though at the cost of computational overhead. As a result, many studies treat RGB methods as a **high-accuracy, high-cost** alternative to grayscale images.

In the context of MIoT, this trade-off is crucial: while RGB representations may improve classification accuracy, their higher resource demands may render them unsuitable for lightweight embedded deployments. This motivates the comparative grayscale–RGB analysis undertaken in this thesis.

Extensions and Variants

Beyond standard image encoding, several studies have proposed enhancements aimed at improving discrimination:

1. **Histogram equalization and contrast adjustments** to amplify structural differences[30].
2. **Space-filling curves** such as Hilbert or Z-order curves to preserve byte-adjacency relationships more effectively[30].
3. **Memory forensics–based imaging** such as Dumpware10 [18] , which converts memory dumps into images for detecting live infections.

These variants highlight the growing interest in exploring richer or more robust image mappings beyond the classical 2D grayscale representation.

Significance for Deep Learning

The malware-as-images paradigm aligns naturally with the strengths of modern deep networks:

1. CNNs can automatically learn multi-scale, texture-based signatures.
2. Transformers can process malware as sequences of image patches.
3. Autoencoders can capture latent structural patterns for anomaly detection.

By circumventing brittle handcrafted features and leveraging mature computer vision models, image-based malware analysis provides a scalable and expressive framework, one particularly suitable for analyzing heterogeneous Linux-based MIoT binaries.

Following this foundational work, subsequent research explored alternative image representations, including grayscale and RGB encodings. These representations differ in computational cost, memory requirements, and representational capacity, leading to ongoing discussions regarding their suitability for malware classification tasks.

2.3 Deep Learning Models for Malware Images

The emergence of deep learning has significantly advanced malware classification research by enabling models to learn complex feature representations directly from raw or image-based malware inputs. Within the malware-as-images paradigm, several deep learning architectures have been applied successfully, including convolutional neural networks (CNNs), transfer learning models, vision transformers, and autoencoder-based architectures.

Several deep learning architectures have been explored for malware image classification. These include convolutional neural networks, transformer-based models, and autoencoder architectures. An overview of these model families is illustrated in Figure 2.3.

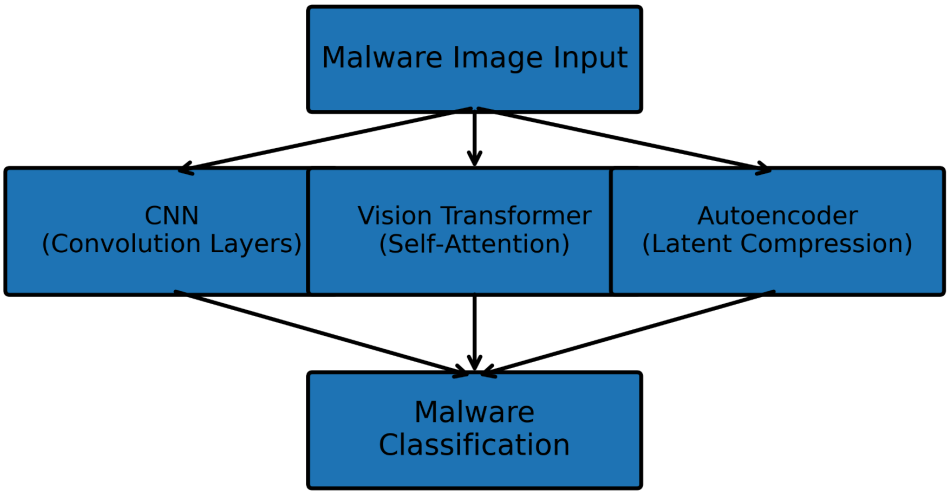

Figure 2.3: Overview of deep learning architectures used in malware image classification, including convolutional neural networks, vision transformers, and autoencoder-based models.

Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) represent the foundational architecture in malware image classification. Their ability to extract spatially localized features, combined with hierarchical representations across multiple convolutional layers, has made them the dominant model family for this task.

Early works demonstrated that even simple CNNs operating on grayscale images could achieve strong performance. Kalash et al. (2018) showed that CNNs automatically extract texture patterns indicative of malware families [7] , while Gibert et al. (2020) confirmed the effectiveness of CNNs across multiple datasets and image encodings [8] .

With the introduction of the MaleVis RGB dataset, deeper architectures such as **VGG**, **ResNet**, **Inception**, and **DenseNet** were evaluated extensively. Bozkir et al. (2019) reported that DenseNet achieved up to 97.48% accuracy on RGB samples, outperforming classical CNN configurations [17] .

Key advantages of CNNs:

- High accuracy on both grayscale and RGB images
- Ability to capture local textures and repeated patterns
- Mature ecosystem (PyTorch, TensorFlow)
- Straightforward training and deployment

Limitations:

- Growth of parameters with increasing depth
- Computation-heavy for MIoT edge devices
- Vulnerability to adversarial perturbations

Despite these challenges, CNNs remain the most widely adopted architecture and serve as a strong baseline in this thesis.

Transfer Learning

Transfer learning leverages pretrained image models (typically trained on ImageNet) by fine-tuning them on malware image datasets. This approach reduces training time,

improves generalization, and enables the use of highly expressive architectures without requiring large malware datasets.

Representative examples include:

- Hybrid CNN models combining pretrained backbones with custom layers, achieving high classification performance (>99 %) across several malware datasets [32].
- Feature extraction using VGG19 or similar networks followed by shallow classifiers such as SVMs or ANN variants[33].
- Lightweight fine-tuning approaches, such as transfer learning with pretrained CNN models, have shown high accuracy on malware image datasets, with minimal computational cost [34].

Transfer learning is especially valuable for MIoT research because pretrained networks can be adapted to small malware datasets, mitigating the scarcity of Linux-targeted samples.

Advantages:

1. Reduced training requirements
2. High accuracy with modest dataset sizes
3. Potential for lightweight deployment (e.g., MobileNet, EfficientNet variants)

Disadvantages:

1. Pretrained models may encode features irrelevant to binary data
2. Larger architectures require optimization for edge platforms
3. Still susceptible to adversarial manipulation

Vision Transformers (ViT)

Vision Transformers (ViTs) represent a more recent architectural paradigm in computer vision and have begun to attract attention in malware classification research. Unlike CNNs, transformers rely on self-attention mechanisms to capture global relationships between image regions. Studies have shown that transformer-based

architectures can achieve competitive performance in malware image classification, particularly when large datasets are available [14].

Benefits of ViTs:

A. Ability to model long-range relationships between distant image regions
B. Potentially superior performance on large and diverse datasets
C. Architecture-agnostic training pipeline

Challenges:

A. Higher data requirements compared to CNNs
B. Increased computational cost
C. Less explored in resource-constrained settings such as MIoT devices

Nonetheless, ViTs offer a promising alternative to CNN-dominated approaches and form one of the model families benchmarked in this thesis.

Autoencoders and Hybrid Architectures

Autoencoders (AEs) compress high-dimensional images into compact latent vectors, enabling unsupervised representation learning and anomaly detection. In malware image research, autoencoder-based systems serve two main purposes:

1. **Dimensionality reduction and feature extraction**
2. **Combining reconstruction errors with classification for enhanced robustness**

Hybrid architectures e.g., CNN encoder + fully connected classifier, can improve classification by leveraging compressed intermediate features. Additionally, AEs have been explored for detecting previously unseen malware families (open-set detection), which is particularly relevant for MIoT environments where zero-day malware variants may emerge.

Advantages:

1. Compact latent embeddings suitable for lightweight classifiers
2. Potential resilience to obfuscation through higher-level feature extraction

3. Flexibility in hybrid designs

Limitations:

1. Often weaker accuracy than supervised CNN or ViT models
2. Reconstruction-based detection remains vulnerable to adversarial perturbations

Autoencoder variants remain an active complementary direction and are included in this thesis for comparative benchmarking.

Ensemble and Lightweight Methods

Recent research has explored ensembles of CNNs, transformers, and statistical models to improve robustness and accuracy. Stacked ensemble techniques have demonstrated strong performance by combining diverse feature spaces and decision boundaries [15]. At the same time, lightweight non–deep-learning methods such as IMCBL [30] achieve near–deep-learning-level accuracy while requiring significantly fewer computational resources. Such models are highly relevant for MIoT deployments, where inference often occurs at the edge.

Ensemble models generally offer robustness and improved performance, but their computational overhead limits their applicability in embedded contexts. Conversely, lightweight alternatives strike a balance between efficiency and accuracy, making them promising candidates for MIoT systems.

2.4 Datasets in Malware Image Research

Datasets play a fundamental role in evaluating malware detection systems. In the malware-as-images domain, datasets typically consist of executable binaries that have been converted into visual representations. Several publicly available datasets have become standard benchmarks for malware image classification research.

Malimg Dataset

The Malimg dataset remains one of the most widely used datasets in malware image classification research [16]. It contains 9,339 malware samples belonging to 25 malware families, each converted into grayscale images. Despite its popularity, Malimg primarily contains Windows-based malware samples and therefore does not fully represent the characteristics of Linux-based malware commonly found in IoT environments.

Characteristics:

- **Format:** Grayscale images (single channel)
- **Platform:** Windows PE malware
- **Class distribution:** Highly imbalanced; several families dominate the dataset
- **Image generation:** Fixed-width reshaping based on file size

Strengths:

- Historical importance and widespread adoption
- Standardized grayscale encoding
- Suitable for benchmarking CNN architectures

Limitations:

- Exclusively Windows malware
- Lack of diversity in file types and architectures
- Not representative of Linux-based MIoT threats

Despite these limitations, Malimg remains the most frequently used dataset for evaluating baseline performance across grayscale models, and many studies report accuracies above 95% when using deep CNNs.

MaleVis Dataset

The MaleVis dataset addresses limitations in earlier datasets and evaluates malware classification with **RGB image encodings** [17] . MaleVis contains **13,000 samples** represented as 3-channel images and includes both benign and malicious executables.

Characteristics:

Format: RGB images

Platform: Primarily Windows executables

Purpose: Benchmarking deep CNNs (ResNet, DenseNet, Inception, VGG)

Advantages: Richer representation due to multi-channel encoding

RGB encoding enables models to learn cross-channel correlations and enhanced texture patterns, which often improves classification accuracy. Prior studies, including DenseNet- and ResNet-based evaluations, demonstrate that MaleVis supports performance levels equal to or exceeding those observed with Malimg.

However, the dataset remains Windows-focused and does not fully capture the characteristics of Linux-based malware relevant to MIoT systems.

Memory-Forensics-Based Datasets (Dumpware10)

To broaden the scope beyond executable files, memory-forensics-based malware datasets were introduced using Dumpware10 framework [18]. Instead of transforming binaries, this approach converts **memory dumps** of running infectious processes into images, enabling the detection of malware during execution.

Characteristics:

Format: Grayscale or RGB images extracted from process memory

Platform: Cross-platform malware samples captured at runtime

Motivation: Detect malware that is obfuscated or dynamically unpacked

Techniques: Manifold learning and computer vision methods

This approach captures real behavioral traces that may not be visible in static binaries, making Dumpware-like datasets especially useful for detecting polymorphic or runtime-evolving malware scenarios that frequently arise in IoT and MIoT environments.

Despite their promise, memory-forensics datasets remain underutilized in large-scale deep-learning research, partly due to the challenges of acquiring representative runtime samples and the sensitivity of forensic data.

Other Emerging Datasets

Recent works have explored additional datasets spanning different platforms and visualization strategies:

- **PCAP-to-image datasets** converting network traffic into RGB images
- **Histogram-transformed malware images** enhancing contrast and structural patterns
- **Space-filling curve–based encodings** for obfuscated or packed malware

While not all of these datasets align directly with image-encoded binaries, they demonstrate the diversity of representations within the broader malware image research landscape.

Limitations of Current Datasets

Across the literature, several recurring limitations are evident:

(a) Overrepresentation of Windows Malware

The vast majority of publicly available datasets contain Windows PE samples. The scarcity of Linux-based malware datasets creates significant challenges in accurate malware detection for MIoT [9][26].

(b) Limited Dataset Diversity

Many datasets include only a handful of malware families or exhibit severe class imbalance, reducing robustness and hindering open-set generalization.

(c) Lack of Real MIoT Samples

Due to regulatory and privacy constraints, authentic MIoT malware datasets are scarce, forcing researchers to rely on generic IoT or PC-based malware corpora.

(d) Few Resources Supporting Adversarial Robustness Studies

Datasets designed specifically to evaluate adversarial perturbations in malware images are rare, limiting research on robustness and secure deployment.

These limitations motivate the dataset strategy of this thesis, which combines multiple publicly available corpora and applies consistent conversion pipelines to enable systematic grayscale–RGB comparisons and robust model benchmarking.

2.5 Linux-Based IoT and MIoT Malware Literature

The rapid expansion of Internet-of-Things ecosystems has introduced new cybersecurity challenges, particularly for embedded devices operating on Linux-based firmware. Medical Internet-of-Things (MIoT) devices frequently rely on lightweight operating systems and remain deployed for extended lifecycles with limited security updates, making them attractive targets for malware attacks [1], [2].

Threat Landscape in IoT and MIoT Systems

IoT-targeted malware has evolved rapidly over the past decade, exploiting weak authentication, insecure network interfaces, vulnerable third-party libraries, and outdated firmware. Foundational studies, such as IoTPOT [3] , revealed the scale and speed with which IoT devices can be compromised through automated brute-force attacks and opportunistic scanning. Mirai and its numerous variants demonstrated how massive botnets can be constructed using compromised Linux-based IoT devices, enabling distributed denial-of-service (DDoS) attacks, credential harvesting, cryptomining, and lateral movement into larger networks.

In MIoT environments, the consequences of malware compromise are significantly more severe:

- **Device manipulation** can disrupt clinical workflows or affect the accuracy of vital-sign monitoring.
- **Data exfiltration** threatens patient privacy and regulatory compliance.
- **Service disruption** can directly impact patient safety.

Surveys emphasize that the security posture of MIoT systems is weakened by long device lifecycles, inconsistent patching practices, and the absence of hardware isolation features standard in general-purpose computers [1], [2] .

As healthcare networks become increasingly interconnected, MIoT devices provide potential footholds for attackers to bypass perimeter defenses, making effective malware detection indispensable.

Characteristics of Linux-Based Malware

Linux malware differs fundamentally from Windows malware both structurally and behaviorally. Executables may be compiled for diverse architectures (ARM, MIPS, PowerPC, x86), often without the rich metadata present in Windows PE headers. Furthermore:

- Linux malware frequently incorporates **lightweight payloads**, reflecting the limited resources of IoT devices.
- **Static linking**, custom toolchains, and stripped binaries complicate static analysis.
- Malware families often include **numerous variants** generated through small code mutations, evasion techniques, and altered propagation scripts.

These characteristics underscore the need for detection methods that generalize across architectures and obfuscation strategies objectives well aligned with image-based deep learning approaches.

Linux-Based Malware Classification Studies

Historically, relatively few works have focused specifically on Linux IoT malware classification. Relevant studies, analyzed Linux IoT malware variants using a combination of binary-level and hybrid approaches, demonstrating that Linux-focused datasets and detection pipelines significantly improve classification fidelity [9] . Their results highlight the insufficiency of Windows-oriented malware detectors when applied to Linux-based environments, confirming the need for platform-specific analysis. Other surveys also note a critical gap: most machine-learning–based studies rely on datasets dominated by Windows PE binaries and rarely evaluate performance

on IoT-targeted malware samples [10] . As a result, there is limited understanding of how well deep-learning models generalize to executables compiled for embedded Linux systems. Memory-forensics approaches such as Dumpware10 [18] offer an alternative perspective by analyzing runtime behavior in a platform-agnostic manner, but these methods are not commonly applied at scale and may not be suitable for resource-constrained MIoT environments.

Security Challenges in MIoT Malware Detection

MIoT devices impose unique operational and architectural constraints:

- **Limited CPU/GPU capabilities** restrict the deployment of large deep-learning models.
- **Low memory capacity** constrains both model size and preprocessing pipelines.
- **Real-time processing requirements** demand low-latency inference.
- **Regulatory barriers** complicate device modification, patching, and firmware instrumentation.

Because of these constraints, traditional malware defenses, including complex dynamic analysis pipelines or heavyweight ensemble models are largely impractical. MIoT contexts require **lightweight, accurate, and robust detection mechanisms**, ideally implemented at edge gateways or within device firmware.

Image-based deep-learning models offer a promising avenue, but their applicability to MIoT systems depends on resolving trade-offs between accuracy, latency, resource consumption, and robustness against adversarial attacks.

Relevance to This Thesis

The gaps identified in the literature reveal a clear opportunity:

- Existing image-based malware detectors focus almost exclusively on Windows PE malware.
- Research targeting Linux IoT malware remains limited in scope and scale.
- MIoT-specific constraints are rarely considered in model design.

- Adversarial robustness in Linux-based malware detection is severely underexplored.

This thesis addresses these shortcomings by:

1. Systematically evaluating grayscale and RGB representations for malware images.
2. Benchmarking CNNs, autoencoders, transformers, and ensemble-type architectures,
3. Focusing on Linux-targeted malware relevant to MIoT deployments.
4. Assessing adversarial robustness under GAN-based perturbations.

In addition, recent research highlights that adversarial machine learning attacks can also target IoT network traffic classification systems, further expanding the attack surface in IoT ecosystems [23]. This indicates that both binary-level and network-level AI models are vulnerable to adversarial manipulation. Through this perspective, the present work contributes to bridging the gap between general malware image research and the practical requirements of real-world MIoT cybersecurity.

2.6 Adversarial Attacks Against Malware Image Classifiers

Despite their strong classification performance, deep learning models are vulnerable to adversarial manipulation. Adversarial attacks involve carefully crafted perturbations applied to input samples with the goal of inducing misclassification while preserving the functionality of the underlying malware. In malware-as-images systems, these perturbations can be applied directly to the image representation derived from executable binaries.

Adversarial Machine Learning in the Malware Domain

Adversarial machine learning (AML) in malware detection differs from classical image-based AML because the underlying data represents **executable code**, not

natural images. Perturbations must preserve operational semantics, meaning the manipulated malware must remain functional. Consequently, early research introduced attacks directly on binary structures, such as modifying unused header fields, injecting no-op instructions, or appending adversarial bytes to the end of a file.

Deep-learning models trained on raw binary features can be evaded through carefully designed modifications that preserve malware behavior while altering its learned representation [11] . Their work shows that even state-of-the-art neural detectors lack robustness when confronted with adversarially altered executables.

These findings underline an important limitation: deep-learning–based malware classifiers often fail to generalize under adversarial conditions, making robustness evaluation essential for practical deployment in sensitive environments like MIoT.

Visualization-Aware Attacks

With the rise of malware image classification, visualization-aware attacks emerged to specifically target models trained on grayscale or RGB malware images. Unlike adversarial manipulations on natural images, visualization-based perturbations must maintain the functional integrity of the binary, the structural layout of executable code and the validity of image-based encodings.

COPYCAT represents one of the most influential frameworks in this domain [12] . It generates adversarial malware images by applying subtle modifications to binary regions that heavily influence CNN classification, while preserving the executable's runtime behavior. COPYCAT demonstrated that CNNs trained on malware images could be misled into misclassification with minimal perturbations, often requiring changes to less than 1% of image pixels.

These attacks expose a fundamental weakness: image-based classifiers tend to rely on superficial visual textures rather than semantically meaningful features. This vulnerability becomes particularly concerning in MIoT contexts, where edge devices may rely on lightweight image classifiers that attackers could bypass with visualization-aware adversarial variants.

GAN-Based Malware Perturbation Techniques

Generative Adversarial Networks (GANs) have further expanded the scope of adversarial malware generation by enabling the creation of sophisticated perturbations or even fully synthetic samples. In the malware image domain, GANs can be trained to generate images that mimic real malware texture patterns, craft perturbations that maximize classifier confusion and produce “camouflaged” images that appear benign to detectors.

Studies have highlighted the application of DCGAN and related architectures to malware detection, showing that GAN-generated samples can bypass even advanced classifiers while retaining visual similarity to genuine malware [13] . GANs effectively learn the distribution of malware images, allowing them to manipulate samples in ways that exploit the decision boundaries of deep-learning models. For MIoT environments, GAN-based attacks pose a realistic threat scenario, as attackers may generate adversarial variants offline and deploy them at scale against vulnerable medical gateways or devices.

Recent work extends adversarial machine learning beyond malware binaries to system-level AI security. Platforms such as AIAS provide integrated defenses against adversarial AI attacks in real-world environments, combining detection, mitigation, and monitoring capabilities [22]. Similarly, recent studies investigate adversarial training and real-time detection pipelines to enhance resilience of AI models under attack conditions [27][28].

Furthermore, adversarial techniques have been explored in broader AI contexts, demonstrating how model vulnerabilities can be systematically analyzed and exploited across different domains [29]. These findings reinforce the need for robust evaluation of malware detection systems beyond standard accuracy metrics.

Implications for Linux-Based MIoT Malware Detection

Adversarial vulnerabilities are particularly problematic in MIoT systems for several reasons:

- ➔ **Resource constraints** prevent the deployment of large, robust, adversarially trained models.

- ➔ **High-stakes environments** require extremely low false-negative rates, as undetected malware may threaten patient safety.
- ➔ **Widespread use of Linux executables** enables attackers to craft adversarial variants tailored to MIoT device architectures.
- ➔ **Limited forensic visibility** on embedded devices makes post-compromise detection difficult.

Despite these risks, adversarial robustness is underexplored in malware image research, especially for Linux-based malware. Most studies evaluate accuracy under clean conditions but seldom consider adversarial perturbations, creating a gap between academic performance metrics and real-world resilience.

This thesis addresses this gap by incorporating a robustness evaluation using FGSM-based perturbations, while more advanced attack scenarios are discussed from the literature.

2.7 Summary and Research Gaps

The literature reviewed in this chapter highlights significant progress in malware detection using deep learning and image-based representations. However, several important research gaps remain, particularly in the context of Medical Internet-of-Things environments.

First, the majority of existing studies focus on Windows-based malware datasets, leaving Linux-oriented IoT malware significantly underexplored. Second, relatively few works provide systematic comparisons between grayscale and RGB malware image representations. Third, adversarial robustness remains insufficiently investigated despite the growing threat of adversarial machine learning attacks against deep learning models. These gaps motivate the research objectives of this thesis, which aim to evaluate malware image representations, benchmark multiple deep learning architectures, analyze adversarial robustness, and assess deployment feasibility under MIoT resource constraints.

Malware Analysis Approaches:

Traditional static and dynamic analysis techniques provide valuable insights but face

well-documented challenges in handling obfuscation, runtime packing, and semantic-preserving mutations [4], [5] . Their resource demands and operational risks further limit their applicability to MIoT devices.

Malware-as-Images Paradigm:

The transformation of binaries into grayscale or RGB images enables end-to-end representation learning and simplifies cross-platform analysis [16], [17] . Image-based methods benefit from the maturity of computer vision architectures but depend heavily on the chosen encoding strategy.

Deep Learning Models:

CNNs, transformers, autoencoders, and transfer learning models have demonstrated high accuracy on malware image datasets [7], [8], [14], [15] . However, most evaluations focus solely on accuracy, often using Windows datasets that do not reflect MIoT environments.

Datasets and Data Limitations:

Publicly available datasets (e.g., Malimg, MaleVis) are predominantly Windows-based [16], [17] and class-imbalanced, with limited representation of Linux or IoT-specific threats. Memory-forensics datasets such as Dumpware10 [18] provide alternatives but remain underused.

IoT and MIoT Malware Landscape:

MIoT devices face increasing threats from Linux-based malware variants that exploit weak authentication, outdated firmware, and resource constraints [1], [2], [3], [9] . Existing malware detectors rarely account for the limited computational capabilities or regulatory constraints of medical devices.

Adversarial Attacks:

Deep-learning based malware classifiers are highly vulnerable to adversarial perturbations both binary-level and visualization-based, capable of inducing misclassification while preserving malware functionality [11], [12], [13] . Robustness is seldom evaluated in prior work, despite its practical importance.

Key Research Gaps

Despite substantial progress, several critical gaps remain unaddressed in the literature:

(1) Limited Focus on Linux-Based Malware for IoT/MIoT

Most image-based malware studies rely on Windows PE datasets.
There is a scarcity of research targeting **Linux executables**, which dominate MIoT ecosystems. The lack of representative datasets leads to detectors that may perform well in academic benchmarks but poorly in real MIoT deployments.

(2) Absence of Systematic Grayscale vs. RGB Comparisons

While both image representations have been explored, no comprehensive study has quantified:

A. accuracy differences,
B. efficiency trade-offs,
C. robustness impact,
D. computational implications for MIoT hardware.

Such a comparison is essential for understanding which encoding is more suitable for edge deployment.

(3) Minimal Consideration of Resource Constraints

Few studies evaluate:

A. model size,
B. inference latency,
C. memory footprint,
D. suitability for embedded or edge systems.

For MIoT devices, where CPU, RAM, and energy budget are restricted, these factors are as important as accuracy.

(4) Underexplored Adversarial Robustness in Malware Images

Despite clear evidence that DL-based detectors are vulnerable to adversarial manipulation, only a handful of works evaluate robustness under:

A. GAN-based perturbations,
B. visualization-aware attacks (e.g., COPYCAT),
C. semantic-preserving binary modifications.

This gap is critical, as MIoT devices are high-value targets in adversarial threat environments.

(5) Lack of Unified Pipelines and Reproducibility

Current literature exhibits inconsistencies in:

- preprocessing pipelines,
- model evaluation protocols,
- dataset splits,
- image generation procedures.

These inconsistencies hinder cross-study comparisons and undermine reproducibility.

Contribution of This Thesis in Addressing These Gaps

The present thesis responds directly to the gaps identified above by:

1. **Focusing on Linux-based malware relevant to MIoT systems**, incorporating datasets beyond traditional Windows corpora.
2. **Conducting the first systematic comparison of grayscale and RGB image encodings** in terms of accuracy, robustness, and computational efficiency.
3. **Benchmarking multiple deep-learning architectures** CNNs, transformers and autoencoders under a unified and reproducible evaluation pipeline.
4. **Evaluating adversarial robustness** against FGSM and visualization-aware perturbations.
5. **Analyzing cost–benefit trade-offs** to identify model and representation combinations suitable for MIoT hardware constraints.

Through this analysis, the thesis bridges the gap between academic malware image research and the practical cybersecurity needs of medical IoT environments.

3 Methodology

3.1 System Overview

This chapter presents the methodological framework used in this thesis for malware classification using deep learning and adversarial robustness evaluation. The overall system pipeline consists of several stages including malware dataset preparation, malware-to-image conversion, deep learning model training, adversarial attack generation, and performance evaluation.[3]

Computer vision-based malware analysis is an established paradigm[6][7][8][16].

The general pipeline of the proposed system is illustrated in Figure 3.1.

Figure 3.1: Overview of the proposed malware classification pipeline. Malware binaries are converted into images, processed by deep learning models, evaluated under adversarial attacks, and assessed using classification metrics.

The workflow of the system can be summarized as follows:

Malware binaries → Image conversion → Deep learning classification → Adversarial attack generation → Evaluation

This pipeline enables the investigation of both classification performance and robustness of deep learning models against adversarial perturbations.

3.2 Dataset

The experiments conducted in this thesis utilize the MalImg dataset, which is one of the most widely used benchmark datasets for malware image classification research.

While MalImg is a popular benchmark dataset, the lack of Linux-based malware representation remains a gap in MIoT detection research [9][10].

The Malimg dataset contains grayscale images generated from malware binaries belonging to multiple malware families. Each malware binary is transformed into a visual representation by mapping each byte value (0–255) to a corresponding pixel intensity. The dataset contains 25 malware families and thousands of malware samples. These samples exhibit structural patterns that can be captured by computer vision models for classification purposes.

To ensure proper evaluation, the dataset was divided into three subsets:

• Training set – used for model learning
• Validation set – used for hyperparameter tuning
• Test set – used for final evaluation

The dataset split used in the experiments follows the configuration defined in the project dataset split file.

3.3 Malware-to-Image Conversion

A key component of malware image classification is the transformation of executable binaries into visual representations. In this process, malware binaries are interpreted as sequences of bytes.Each byte value (ranging from 0 to 255) is mapped to a corresponding pixel intensity, forming a visual representation of the binary file. This transformation produces grayscale images that capture structural patterns of the binary file.[3]

The malware image generation pipeline is illustrated in Figure 3.2.

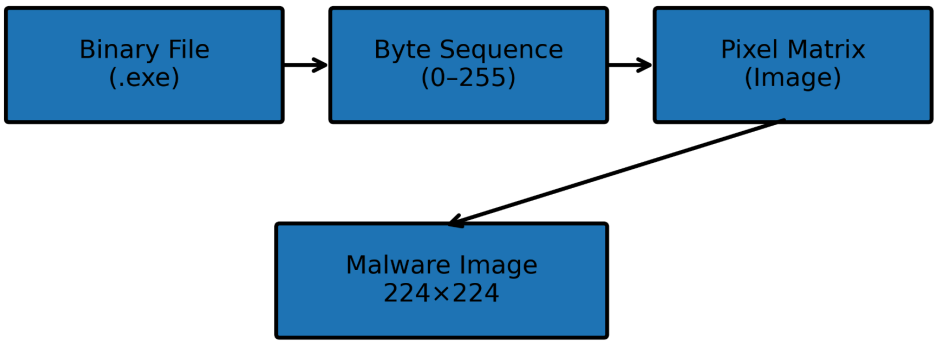

Figure 3.2: Malware binary to image conversion process. Binary files are interpreted as byte sequences that are mapped to pixel intensities to form image representations suitable for deep learning models.

Two types of image representations were used in the experiments:

• Grayscale representation (single channel)
• RGB representation (three channels)

In the RGB representation, grayscale images are replicated into three channels to match the input expectations of deep learning models designed for RGB images. This allows us to evaluate whether richer representations improve classification performance, despite the added computational cost. This method follows previous work where grayscale images were converted to RGB representations to facilitate model compatibility [17][31].

All images were resized to a fixed resolution of 224 × 224 pixels in order to ensure compatibility with the deep learning models used in this study.

3.4 Deep Learning Models

In this work, **three deep learning architectures** were initially evaluated for malware image classification:

- **ResNet-18**
- **EfficientNet-B0**

- **DeiT-Tiny Vision Transformer**

Each of these models represents a distinct deep learning paradigm, enabling a comparative analysis of convolutional and transformer-based models for malware image classification. They were selected based on their proven performance in image classification tasks and their ability to operate within the constraints of resource-limited environments, such as those commonly found in Medical IoT (MIoT) systems.

1. **ResNet-18**:

   ResNet-18 is a convolutional neural network (CNN) that utilizes **residual connections** to address the vanishing gradient problem and enable the training of deeper architectures. Its key strength lies in its ability to learn efficient features from input data through its layered convolutional operations, making it a strong candidate for spatial pattern recognition in malware images. ResNet-18 has demonstrated strong performance in various computer vision tasks and serves as a solid baseline for malware image classification.

2. **EfficientNet-B0**:

   EfficientNet-B0 is another CNN that optimizes the scaling of **network depth**, **width**, and **resolution** through a compound scaling method. This approach enables high performance while maintaining computational efficiency, making it particularly suited for embedded or edge devices, which often have strict computational and memory constraints. EfficientNet-B0 has been shown to outperform traditional CNN architectures in terms of accuracy and efficiency, making it ideal for malware classification tasks where both high accuracy and efficiency are essential.

3. **DeiT-Tiny Vision Transformer**:

   The DeiT-Tiny model is a **vision transformer (ViT)**, which uses **self-attention mechanisms** to process image patches. Unlike CNNs, which operate locally, vision transformers can capture **long-range dependencies** across the entire image, potentially offering better performance for complex datasets. While transformer models are traditionally used in natural language processing tasks, they have recently gained attention in computer vision due to

their ability to capture global context, which is important for identifying patterns in malware images.

These architectures were selected because they represent a blend of traditional convolutional approaches and more recent transformer-based methods, offering a broad view of the potential for deep learning in malware detection. They are evaluated based on their classification performance, generalizability, and computational efficiency, crucial factors for real-world deployment in MIoT contexts.

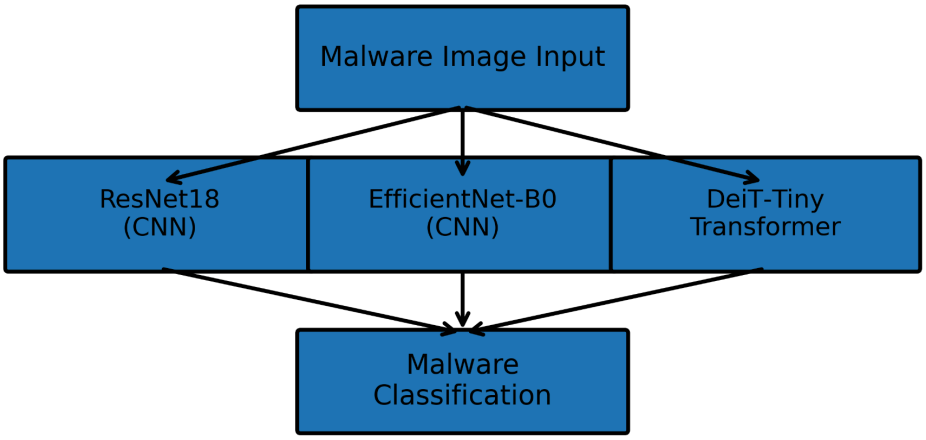

Figure 3.3: Deep learning architectures evaluated for malware image classification, including ResNet18, EfficientNet-B0, and the DeiT-Tiny Vision Transformer.

Autoencoders and Row-LSTM

As the next step, two additional models will be introduced and explained for malware image classification evaluation:

- **Autoencoder**
- **Row-LSTM**

These models will be discussed in further detail in the upcoming sections, as they provide alternative approaches to feature learning and sequence modeling for malware detection tasks. The **Autoencoder** will be considered for its ability to learn compact representations of input data, while the **Row-LSTM** will be evaluated for its sequential approach to processing image rows.

3.5 Training Configuration

All models were trained using the PyTorch deep learning framework. Training was performed on a GPU-enabled environment to accelerate computation.

The following training configuration was used:

• Number of epochs: 10
• Batch size: 128
• Optimizer: Adam
• Image size: 224 × 224 pixels

During training, the models were optimized to minimize the cross-entropy loss function. The validation dataset was used to monitor training progress and to select the best-performing model checkpoint.

Model checkpoints were saved after each epoch to allow performance comparison across training iterations.

3.6 Adversarial Attack Generation

To evaluate the robustness of the trained models, adversarial examples were generated using the Fast Gradient Sign Method (FGSM)[6].

FGSM is a gradient-based adversarial attack that perturbs the input image by adding a small perturbation in the direction of the gradient of the loss function with respect to the input.

The adversarial perturbation is computed as: $\mathbf{x\_adv = x + \varepsilon \cdot sign(\nabla\_x L(x, y))}$, where x represents the original input image, ε controls the magnitude of the perturbation, and L represents the loss function.

In this study, multiple perturbation levels were tested in order to evaluate the sensitivity of the models to adversarial noise.

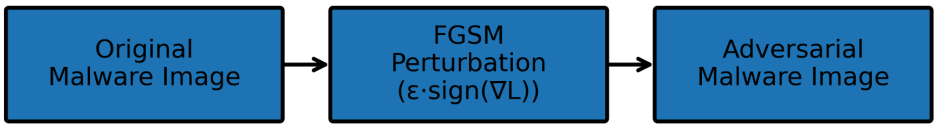

Figure 3.4: Conceptual illustration of the FGSM adversarial attack. A small perturbation is added to the original malware image to generate an adversarial example designed to mislead the classifier.

In addition to gradient-based attacks, more advanced adversarial techniques based on Generative Adversarial Networks (GANs) have been proposed in the literature, demonstrating the ability to generate highly realistic and evasive malware variants [12][13].GAN-based techniques for generating adversarial samples in malware detection have been explored in several works [12][13][22].However, the implementation of GAN-based attacks involves significant complexity, including the need for stable generative training and domain-specific constraints.

Due to these challenges, GAN-based adversarial attack generation is not implemented in this study. Instead, adversarial examples were generated using the Fast Gradient Sign Method (FGSM), a gradient-based attack.

3.7 Evaluation Metrics

To evaluate the performance of the malware classification models, two primary evaluation metrics were used:

- Classification Accuracy
- Macro F1-score

Accuracy measures the proportion of correctly classified malware samples over the total number of samples. The Macro F1-score provides a balanced evaluation metric that accounts for class imbalance by computing the F1-score independently for each class and then averaging the results. These metrics provide a comprehensive

evaluation of both classification performance and model robustness when evaluated under adversarial attack scenarios.

3.8 Experimental Environments and Reproducibility

Two execution environments were used: a local CPU environment for initial validation and a GPU server for full experiments.

Local CPU Validation Environment

A local run was performed with ResNet-18 on CPU to validate correctness of dataset loading, splitting, training loop stability, checkpointing, and basic convergence behavior. This stage ensures that failures encountered later on the cluster are not due to application logic errors.

GPU Training and Evaluation Environment (DGX1)

The main experiments were executed on a shared GPU server (DGX-class system) using the following reproducibility controls:

- execution inside a **Singularity container**
- GPU selection using CUDA_VISIBLE_DEVICES=3
- explicit bind-mount of the project directory into /work
- explicit PYTHONPATH export to ensure correct module resolution

The server status snapshot observed during login included a temperature reading of **66.4°C** and memory usage **67%**, reflecting a busy shared environment; this motivates the containerized approach for stable dependencies and repeatable execution contexts.

Containerization Strategy

Instead of relying on a fragile host Python installation, training and evaluation were executed inside a CUDA-enabled PyTorch Singularity image (mounted project under /work). This approach reduces “works-on-my-machine” inconsistencies and isolates model dependencies (torch, torchvision, timm, torchmetrics, sklearn, matplotlib) from the host environment.

3.9 Mapping Methodology to Thesis Research Questions

This methodology supports the thesis questions as follows:

RQ1 (grayscale vs RGB) is addressed by establishing a grayscale baseline pipeline first (Malimg), then extending the same controlled split/training/evaluation framework to RGB datasets and RGB encodings.

RQ2 (best models under MIoT constraints) is addressed by benchmarking diverse architectures (CNN and transformer families) under the same preprocessing protocol and recording both predictive performance and operational artifacts (checkpoints, evaluation outputs) for later efficiency analysis.

RQ3 (adversarial vulnerability) is enabled by the strict separation of training and evaluation artifacts and by using deterministic image inputs; the resulting trained classifiers become targets for adversarial perturbation generation in later experimental stages.

RQ4 (trade-offs) is supported by the reproducible pipeline and standardized artifact generation, allowing Chapter 4 and Chapter 5 to quantify accuracy and macro F1 jointly with practical measures such as runtime settings, model size, and hardware constraints.

4 Experimental Evaluation and Results

4.1 Experimental Setup

This section describes the experimental setup used to evaluate the proposed AI-powered malware detection framework for Medical Internet-of-Things (MIoT) environments. The experimental design is focusing on deep learning–based malware detection using image representations, comparison between grayscale and RGB encodings, and robustness evaluation under adversarial conditions.

Image-based malware analysis has emerged as an effective approach for capturing structural patterns in binary files and enabling the application of computer vision techniques for classification tasks [6][8][16]. In the context of MIoT environments, such approaches are particularly relevant due to their ability to operate without dynamic execution, which is often infeasible on resource-constrained devices [1][2].

All experiments were conducted under a unified and reproducible protocol to ensure fair comparison across architectures and input representations. Reproducible experimental protocols are key for consistent evaluation of AI models [22]. The evaluation pipeline includes dataset preparation, preprocessing, model training, performance evaluation, and robustness analysis.

Dataset and Splits

The experimental evaluation is based on the **Malimg dataset** [16], which contains 9,339 malware samples categorized into 25 malware families. This dataset is widely adopted in malware image classification research and provides a standardized benchmark for evaluating deep learning models [7][8].

To ensure reproducibility and consistency across all experiments, a fixed train/validation/test split was generated and reused across all configurations (grayscale and RGB). All reported performance results correspond exclusively to the held-out test set.

Table 4.1 – Malimg Dataset Split

Split	**Samples**
Training	6,540
Validation	1,398

Test	1,401
Total	**9,339**

The Malimg dataset exhibits class imbalance across malware families, which is a common challenge in malware classification tasks and motivates the use of macro-averaged evaluation metrics [10][20].

Image Preprocessing

All malware binaries were transformed into image representations and resized to a fixed resolution of **224 × 224 pixels**, ensuring compatibility with standard convolutional and transformer-based architectures.

Two input modalities were considered:

- **Grayscale representation:**
  Malware binaries are mapped directly to pixel intensities (0–255), preserving raw byte-level structural information. This representation has been widely used in prior work due to its simplicity and effectiveness [6][16].
- **RGB representation:**
  Grayscale images were converted into three-channel RGB format to evaluate whether additional channel capacity improves classification performance. Similar approaches have been explored in recent studies comparing representation modalities in malware image classification [17].

A separate dataset root and corresponding split file were created for RGB experiments, ensuring that label assignments and data splits remained identical across representations. No data augmentation techniques were applied in this study. This design choice ensures that the observed performance differences are attributable to model architecture and input representation, rather than augmentation-induced variability.

Experimental Pipeline

The overall experimental workflow consists of the following stages:

1. Conversion of malware binaries into image representations (grayscale and RGB)
2. Dataset splitting and preprocessing
3. Model training using deep learning architectures
4. Performance evaluation on a held-out test set
5. Adversarial robustness testing
6. Computational and resource profiling

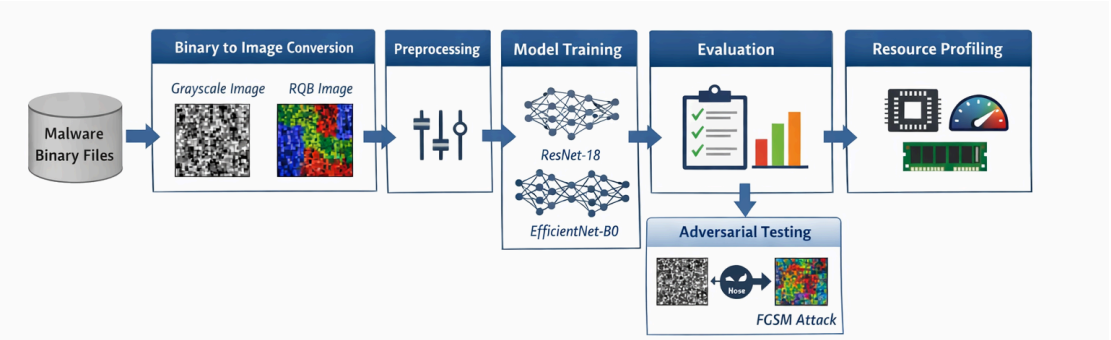

Figure 4.1 – Experimental Pipeline

End-to-end workflow of the proposed malware detection framework, including binary-to-image conversion, preprocessing, model training, evaluation, adversarial testing, and resource profiling.

This pipeline reflects a typical malware-as-images framework and enables systematic evaluation of both performance and deployment feasibility [8][10].

Reproducibility Considerations

To ensure reproducibility, all experiments were conducted using:

- A fixed dataset split across all models and representations
- Consistent preprocessing procedures
- Identical evaluation protocols

Model checkpoints were saved during training, and all evaluations were performed on the same test set under controlled conditions.

This controlled experimental design enables a reliable comparison of:

1. Grayscale versus RGB representations
2. Different deep learning architectures (CNNs, transformers, RNNs, autoencoders)
3. Performance under adversarial perturbations
4. Computational and deployment trade-offs in MIoT environments

Such reproducibility is essential in malware detection research, where inconsistent evaluation setups often lead to misleading performance comparisons [10][20].

4.2 Models and Training Configuration

This section describes the deep learning architectures and training configuration used for malware classification. The selected models represent different learning paradigms, including convolutional neural networks (CNNs), transformer-based architectures, unsupervised representation learning, and sequential modeling. This diversity enables a comprehensive evaluation of architectural suitability for malware-as-images classification in MIoT environments.

Evaluated Architectures

Five models were selected to cover a broad spectrum of architectural approaches. Deep learning architectures, including CNNs and transformers, have been extensively benchmarked in malware image classification research [8][14][15]:

ResNet-18 (CNN baseline):
A widely used convolutional neural network that employs residual connections to enable stable training of deep architectures. CNN-based models have consistently demonstrated strong performance in malware image classification tasks due to their ability to capture local spatial patterns [7][8].

EfficientNet-B0 (parameter-efficient CNN):
EfficientNet introduces compound scaling to balance network depth, width, and resolution, achieving high performance with relatively low parameter count. EfficientNet-based approaches have shown superior performance in image-based malware classification tasks [8][10].

DeiT-Tiny (Vision Transformer):

Transformer-based models process images as sequences of patches and are capable of capturing long-range dependencies. Recent studies have explored Vision Transformers for malware classification, demonstrating competitive performance compared to CNNs [14].

Convolutional Autoencoder (unsupervised learning):

Autoencoders learn compressed latent representations by minimizing reconstruction error. In malware analysis, they are commonly used for anomaly detection and feature extraction rather than direct classification [10][20].

Row-LSTM (RNN-based sequential model):

This model treats each image row as a time step in a sequence, allowing evaluation of whether sequential dependencies can capture malware structure. Recurrent neural networks have been applied in malware analysis, particularly for sequential data such as system call traces [4].

Table 4.2 – Evaluated Architectures Overview

Model	**Type**	**Learning Paradigm**	**Role in Study**
ResNet-18	CNN	Supervised	Baseline model
EfficientNet-B0	CNN	Supervised	High-performance model
DeiT-Tiny	Transformer	Supervised	Attention-based model
Autoencoder	CNN	Unsupervised	Representation learning

Row-LSTM	RNN	Supervised	Sequential modeling

Training Configuration

All supervised models were trained as multiclass classifiers with **25 output classes**, corresponding to the malware families in the Malimg dataset.

Loss Function:

Cross-entropy loss was used for all classification models, as it is standard for multiclass classification tasks [8].

Autoencoder Loss:

The autoencoder was trained using **Mean Squared Error (MSE)** loss to minimize reconstruction error between input and output images.

Batch Size:

A batch size of **128** was used across all experiments to ensure consistency and efficient GPU utilization.

Training Duration:

Models were trained for up to **10 epochs**, with checkpointing performed at each epoch to ensure reproducibility and allow model selection.

Optimization:

Standard stochastic gradient-based optimization was applied (e.g., Adam or SGD), which is commonly used in deep learning–based malware detection systems [10].

Input Configuration

All models were trained using input images of size **224 × 224 pixels**, consistent with standard ImageNet-based architectures.

- For **grayscale experiments**, images were loaded as single-channel inputs and internally expanded to three channels when required by pretrained architectures.
- For **RGB experiments**, images were explicitly represented as three-channel tensors.

This setup ensures compatibility across architectures while enabling fair comparison between grayscale and RGB modalities.

Training Consistency and Fair Comparison

To ensure a fair and controlled comparison:

- All models were trained using the **same dataset split**
- Identical preprocessing pipelines were applied
- The same evaluation metrics were used across all experiments
- No data augmentation was applied

This controlled setup isolates the impact of:

A. Model architecture
B. Input representation (grayscale vs RGB)

Such consistency is critical in malware classification research, where differences in preprocessing or data splits can significantly affect reported performance [10][20].

Design Rationale

The selection of architectures reflects the requirements of MIoT malware detection:

- **CNNs** are expected to perform well due to their ability to capture spatial patterns in malware images [8]
- **Transformers** are evaluated for their ability to model global dependencies [14]
- **Autoencoders** provide insight into representation learning and anomaly detection potential
- **RNN-based models** test whether sequential modeling is suitable for image-based malware representations

This combination enables a comprehensive evaluation of both performance and architectural suitability under realistic deployment constraints.

4.3 Evaluation Metrics

The performance of the evaluated models was assessed using a set of standard classification and robustness metrics, selected to reflect both predictive accuracy and reliability under class imbalance and adversarial conditions.[7]

Given the characteristics of malware classification datasets, particularly the presence of class imbalance across malware families, multiple complementary metrics were employed to provide a comprehensive evaluation.

Classification Metrics

The primary evaluation metrics used in this study are:

- **Accuracy**
- **Macro-averaged F1-score**

Accuracy

Accuracy measures the proportion of correctly classified samples over the total number of test samples. It provides a general indication of model performance:

$$Accuracy = \frac{Correct\ Predictions}{Total\ Predictions}$$

While accuracy is widely used in malware classification studies, it can be misleading in imbalanced datasets, where dominant classes may disproportionately influence the overall score [10][20].

Macro-Averaged F1-Score

To address class imbalance, the **macro-averaged F1-score** was used as a primary evaluation metric. Macro F1 computes the F1-score independently for each class and then averages the results, giving equal importance to all malware families.

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

Macro-averaging is particularly important in malware classification, where minority families may otherwise be underrepresented in performance evaluation [10]. This metric penalizes models that perform poorly on less frequent malware classes, making it more suitable for realistic threat detection scenarios.

Key Insight:
A high accuracy combined with a significantly lower macro F1-score typically indicates poor performance on minority classes, which is a common issue in malware datasets.

Adversarial Robustness Metrics

To evaluate model robustness under adversarial conditions, classification performance was measured on perturbed inputs generated using adversarial attack methods.[6][7]. Adversarial robustness is critical for ensuring real-world deployment [11][27].

Adversarial Accuracy

Adversarial accuracy measures the classification performance of a model when inputs are modified by adversarial perturbations:

$$Adversarial\ Accuracy = \frac{Correct\ Predictions\ (Adversarial)}{Total\ Samples}$$

This metric captures the degradation in model performance under attack conditions and is essential for assessing deployment reliability in adversarial environments [11][12].

Attack Success Rate (ASR)

The **Attack Success Rate (ASR)** quantifies the effectiveness of adversarial attacks by measuring the proportion of originally correct predictions that become misclassified after perturbation:

$$ASR = \frac{Successful\ Attacks}{Originally\ Correct\ Predictions}$$

ASR provides a complementary perspective to adversarial accuracy, highlighting the vulnerability of models to targeted perturbations [11].

Metric Selection Rationale

The selected evaluation metrics reflect the specific requirements of malware detection in MIoT environments:

- **Accuracy** provides a general performance baseline
- **Macro F1-score** ensures fair evaluation under class imbalance
- **Adversarial accuracy and ASR** capture robustness against evasion attacks

This combination enables a balanced evaluation of:

- Classification performance
- Generalization across malware families
- Robustness under adversarial conditions

Such multi-dimensional evaluation is essential in cybersecurity applications, where both accuracy and resilience are critical for real-world deployment [10][11].

4.4 Baseline Results (Grayscale vs RGB)

This section presents the baseline classification performance of the evaluated models using malware images derived from binary files. The analysis focuses on comparing grayscale and RGB representations under identical training and evaluation conditions, directly addressing **RQ1**.

Image-based malware classification has been widely studied using grayscale representations, where raw byte values are mapped directly to pixel intensities, preserving structural patterns within executable files [6][16]. More recent approaches have explored RGB encodings to potentially enhance representational capacity by introducing additional channels [17]. All models were trained and evaluated using the fixed dataset split described in Section 4.1. Performance is reported exclusively on the held-out test set.

Grayscale Baseline Performance

The grayscale representation serves as the primary baseline, as it directly encodes byte-level information without artificial transformation.

Table 4.3 – Grayscale Classification Performance

Model	Test Accuracy (%)	Macro F1-score
ResNet-18	89.35	0.9068
EfficientNet-B0	**99.43**	**0.9864**
DeiT-Tiny	96.43	0.8897

These results indicate that grayscale representations are sufficient for capturing discriminative malware patterns, achieving high classification performance without the overhead of multi-channel inputs. Prior work suggests that grayscale encodings often outperform RGB in terms of computational efficiency and performance [7][17][10].

Analysis of Grayscale Results

EfficientNet-B0 achieves near-perfect classification performance, reaching **99.43% accuracy** and **0.9864 macro F1-score**, indicating highly effective feature extraction from grayscale malware images. This result confirms prior findings that modern convolutional architectures can successfully capture structural and texture-based patterns in malware visualizations [8][10].

ResNet-18, while significantly simpler, achieves strong baseline performance but is clearly outperformed by EfficientNet-B0. This highlights the importance of architectural scaling and feature representation capacity in malware classification tasks.

DeiT-Tiny achieves high overall accuracy (96.43%) but a lower macro F1-score (0.8897), suggesting sensitivity to class imbalance. This discrepancy indicates that while the model performs well on dominant malware families, it struggles to generalize across minority classes, a known challenge in transformer-based approaches under limited data conditions [14].

RGB Representation Results

To evaluate whether additional channel capacity improves classification performance, grayscale images were converted into RGB format and used as input to the models.

Table 4.4 – RGB Classification Performance

Model	Test Accuracy (%)	Macro F1-score
EfficientNet-B0	98.07	0.9526
DeiT-Tiny	95.22	0.8789

While RGB representations provide a richer encoding of the input data, the performance gains are not consistently significant compared to grayscale, especially when considering the increased computational cost.

Analysis of RGB Results

The results indicate that RGB encoding does not lead to improved classification performance. EfficientNet-B0 shows a decrease in both accuracy and macro F1-score compared to its grayscale counterpart, suggesting that the additional channels do not introduce meaningful discriminative information. Similarly, DeiT-Tiny achieves slightly lower performance in RGB format, reinforcing the observation that increased input dimensionality does not necessarily translate to improved learning in malware image classification tasks. These findings are consistent with prior work suggesting that malware images do not exhibit natural color semantics, and thus additional channels may introduce redundancy rather than useful features [6][17].

Grayscale vs RGB Comparison

We compare the grayscale representation, which uses a single channel, with the 3-channel replicated RGB representation where grayscale images are expanded into three channels for compatibility with models designed for color inputs. To directly assess the impact of representation, Table 4.5 presents a side-by-side comparison.

Table 4.5 – Grayscale vs RGB Comparison

Model	**Grayscale Acc (%)**	**Grayscale F1**	**RGB Acc (%)**	**RGB F1**
EfficientNet-B0	**99.43**	**0.9864**	98.07	0.9526
DeiT-Tiny	96.43	0.8897	95.22	0.8789

Comparative Interpretation

Across all evaluated architectures, grayscale representation consistently matches or outperforms RGB encoding. The observed performance gap suggests that:

- Grayscale images preserve essential structural information directly derived from binary data
- RGB conversion introduces additional complexity without increasing semantic richness
- Model capacity is better utilized when learning from compact and information-dense inputs

From a computational perspective, RGB encoding also increases memory consumption and input dimensionality, further reducing its practical utility in resource-constrained environments.

Implications for MIoT Malware Detection

In MIoT environments, where computational resources are limited and real-time detection is critical, the choice of input representation plays a significant role.

The experimental results indicate that:

- Grayscale encoding provides **superior or equivalent performance**
- RGB encoding introduces **additional computational overhead**
- No consistent performance gains justify the increased cost

Answer to RQ1

Under the controlled experimental conditions of this study:

1. Grayscale representation is **more efficient and equally or more effective** than RGB
2. RGB encoding does **not provide measurable performance benefits**
3. The increased computational cost of RGB inputs is **not justified**

Therefore, grayscale malware images emerge as the **preferred representation** for MIoT-oriented malware detection pipelines.

4.5 Model Comparison

This section provides a consolidated comparison of all evaluated architectures, integrating results from previous experiments in order to identify the most effective models for malware classification in MIoT environments. The comparison focuses on classification performance, generalization capability, and architectural suitability, directly addressing **RQ2**.

Deep learning–based malware classification has been extensively studied using convolutional architectures, while more recent approaches explore transformers and hybrid models [8][10][14]. However, performance alone is not sufficient; models must also be evaluated in terms of their ability to generalize across malware families and operate under realistic constraints.

Table 4.6 – Overall Model Comparison (Grayscale Baseline)

Model	Type	Accuracy (%)	Macro F1	Key Characteristics
EfficientNet-B0	CNN	**99.43**	**0.9864**	High performance, efficient scaling
DeiT-Tiny	Transformer	96.43	0.8897	Global attention, imbalance sensitivity
ResNet-18	CNN	89.35	0.9068	Lightweight baseline
Autoencoder	CNN (unsup.)	—	—	Stable representation learning
Row-LSTM	RNN	39.11	0.1130	Sequential modeling (ineffective)

Performance Ranking

Based on classification performance under grayscale inputs:

1. **EfficientNet-B0** – highest accuracy and macro F1-score
2. **DeiT-Tiny** – strong accuracy but reduced macro performance
3. **ResNet-18** – solid baseline but lower overall performance
4. **Row-LSTM** – significantly underperforms
5. **Autoencoder** – not directly comparable (unsupervised)

EfficientNet-B0 clearly outperforms all other models, confirming the effectiveness of modern CNN architectures in malware image classification tasks [8][10].

CNN vs Transformer Performance

The comparison between CNN-based and transformer-based architectures reveals important differences:

1) **CNNs (EfficientNet, ResNet)**
    a) Strong local feature extraction
    b) High accuracy on structured image patterns
    c) Well-suited for malware texture representations
2) **Transformers (DeiT-Tiny)**
    a) Capture global dependencies
    b) Competitive accuracy
    c) More sensitive to class imbalance and dataset size

The results suggest that malware images primarily encode **local spatial patterns**, which are more effectively captured by convolutional operations. This observation aligns with prior research in malware visualization–based classification [6][8].

Sequential and Unsupervised Models

The evaluation of alternative architectures highlights their limitations:

- **Row-LSTM:**
  Treating images as sequences of rows leads to a significant loss of spatial information. The model fails to capture discriminative features, resulting in extremely low accuracy and macro F1-score. This confirms that malware images cannot be effectively modeled as purely sequential data.
- **Autoencoder:**
  The autoencoder demonstrates stable reconstruction performance, indicating successful representation learning. However, it does not provide direct classification capability and must be combined with a downstream classifier for practical use [10][20].

Generalization and Class Imbalance

The discrepancy between accuracy and macro F1-score in some models (e.g., DeiT-Tiny) highlights the impact of class imbalance.

- High accuracy does not necessarily indicate strong generalization
- Macro F1-score reveals performance on minority malware families
- CNNs demonstrate better balance between overall accuracy and class-wise performance

This reinforces the importance of using macro-averaged metrics in malware classification tasks [10][20].

Implications for Model Selection

From a practical perspective, model selection for MIoT malware detection should consider:

- **Accuracy:** EfficientNet-B0 provides the best classification performance
- **Generalization:** CNNs outperform transformers under class imbalance
- **Architectural suitability:** Spatial inductive bias is critical
- **Deployment constraints:** Lightweight models may still be required

Answer to RQ2

The experimental results demonstrate that:

1. CNN-based architectures provide the most effective solution for malware image classification
2. EfficientNet-B0 achieves the best trade-off between accuracy and model efficiency
3. Transformer-based models offer competitive performance but are more sensitive to data imbalance
4. Sequential models (RNNs) are not suitable for image-based malware classification
5. Unsupervised models (autoencoders) are useful for representation learning but not standalone classification

Overall, convolutional architectures emerge as the **most suitable and reliable choice** for MIoT malware detection in the evaluated setting.

4.6 Adversarial Robustness Results

While high classification accuracy is essential, malware detection systems deployed in adversarial environments must remain robust against evasion attempts. In real-world scenarios, attackers may intentionally modify malware samples to bypass detection systems without altering their functionality. Adversarial machine learning has demonstrated that deep learning models are vulnerable to such carefully crafted perturbations [11][12].

This section evaluates the robustness of the examined models under adversarial conditions, directly addressing **RQ3**.

Adversarial Attack Method

Adversarial samples were generated using the **Fast Gradient Sign Method (FGSM)**, a widely used gradient-based attack that perturbs input data in the direction of the loss gradient [11].

The perturbation is defined as:

$$x_{adv} = x + \epsilon \cdot sign(\nabla_x J(x, y))$$

where:

x is the original input image

ϵ controls the perturbation magnitude

$J(x, y)$ is the loss function

∇_x denotes the gradient with respect to the input

FGSM is particularly relevant in malware detection, as it simulates realistic evasion strategies where minimal changes can significantly impact model predictions [12].

Experimental Setup

Adversarial evaluation was conducted on the **RGB Malimg test set**, ensuring consistency with the representation used during attack generation.

Three perturbation magnitudes were considered:

- **ε = 0.005** (low perturbation)
- **ε = 0.01** (moderate perturbation)
- **ε = 0.2** (strong perturbation)

Model performance was evaluated using classification accuracy under each perturbation level.

Table 4.7 – Classification Accuracy under FGSM Attack

Model	**Clean Acc**	**ε = 0.005**	**ε = 0.01**	**ε = 0.2**
ResNet-18	0.9779	0.5182	0.2270	0.0000
EfficientNet-B0	0.9807	0.2391	0.1777	0.0714
DeiT-Tiny	0.9522	**0.7623**	**0.5253**	**0.3084**

The observed performance degradation under FGSM perturbations highlights the vulnerability of deep learning models to adversarial manipulation, even when perturbations are visually imperceptible.

Robustness Analysis

The results reveal significant performance degradation across all models under adversarial perturbations, highlighting the vulnerability of deep learning–based malware classifiers.

CNN Vulnerability

CNN-based models have been shown to degrade rapidly under adversarial perturbations [11]. In this thesis, the results show that both **ResNet-18** and **EfficientNet-B0** exhibit **rapid degradation** even under low perturbation levels:

- EfficientNet-B0 drops from **98.07% to 23.91%** at $\varepsilon = 0.005$
- ResNet-18 drops from **97.79% to 51.82%** at $\varepsilon = 0.005$

Under strong perturbation ($\varepsilon = 0.2$):

- ResNet-18 collapses completely (**0% accuracy**)
- EfficientNet-B0 retains only **7.14% accuracy**

This behavior confirms that CNN-based models are highly sensitive to gradient-based adversarial noise, as also observed in prior work on adversarial malware evasion [11][12].

Transformer Robustness

In contrast, **DeiT-Tiny** demonstrates significantly **stronger** robustness:

- Retains **76.23% accuracy** at $\varepsilon = 0.005$
- Retains **52.53% accuracy** at $\varepsilon = 0.01$
- Maintains **30.84% accuracy** even at $\varepsilon = 0.2$

The degradation curve is notably smoother compared to CNNs, indicating improved resistance to small and moderate perturbations.

This behavior is consistent with recent findings suggesting that transformer-based models may exhibit increased robustness due to their global attention mechanisms [14].

Clean Accuracy vs Robustness Trade-off

A clear trade-off emerges:

- **EfficientNet-B0:** highest clean accuracy, lowest robustness
- **DeiT-Tiny:** slightly lower clean accuracy, highest robustness
- **ResNet-18:** moderate accuracy, severe degradation under attack

This demonstrates that optimizing for clean performance does not guarantee resilience in adversarial environments.

Implications for MIoT Security

In MIoT environments, adversarial robustness is critical due to the high-stakes nature of healthcare systems.[7]

The experimental findings suggest that:

- Deep learning models can be **easily bypassed** using small perturbations
- CNN-based detectors are particularly vulnerable to adversarial manipulation
- Transformer-based models provide improved robustness under low-intensity attacks
- No evaluated model remains reliable under strong adversarial conditions

Given that malware authors may actively attempt to evade detection systems, robustness must be treated as a primary design requirement rather than an optional evaluation criterion.

Answer to RQ3

The adversarial evaluation demonstrates that:

1. Malware image classifiers are highly sensitive to adversarial perturbations
2. CNN-based architectures degrade rapidly under gradient-based attacks
3. Transformer-based models exhibit comparatively stronger robustness at low perturbation levels
4. Clean accuracy alone is not sufficient for evaluating deployment readiness

These findings highlight the necessity of incorporating adversarial defense mechanisms, such as adversarial training or hybrid detection pipelines, in future MIoT malware detection systems.

4.7 Efficiency and Resource Analysis

Beyond classification accuracy and robustness, malware detection systems deployed in Medical Internet-of-Things (MIoT) environments must satisfy strict computational and resource constraints. Devices operating in such environments often have limited memory, processing power, and energy availability, making efficiency a critical factor in model selection [1][2].

This section evaluates the computational footprint of the examined architectures, addressing **RQ4** by analyzing memory usage, parameter count, and inference efficiency.

Computational Profiling Setup

Resource profiling was performed under GPU execution using the validation split to ensure consistent measurement conditions across models.

The following metrics were collected:

- **Number of parameters (model size)**
- **Peak GPU memory usage (VRAM)**
- **Inference throughput (images per second)**

These metrics provide insight into both **computational cost** and **deployment feasibility**.

Table 4.8 – Computational Resource Profiling

Model	**Input**	**Parameters (M)**	**Peak VRAM (MB)**	**Throughput (img/sec)**

ResNet-18	Grayscale	11.18	859	**2457**
ResNet-18	RGB	11.19	909	2185
EfficientNet-B0	Grayscale	4.04	1322	1085
EfficientNet-B0	RGB	4.04	1372	1099
DeiT-Tiny	Grayscale	5.43	**275**	1146
DeiT-Tiny	RGB	5.53	325	1088

Memory Footprint Analysis

Significant differences in memory usage are observed across architectures:

- **DeiT-Tiny** exhibits the lowest memory footprint (~275 MB), making it highly suitable for memory-constrained environments
- **EfficientNet-B0**, despite having fewer parameters than ResNet-18, consumes more memory due to intermediate feature maps and compound scaling
- **RGB inputs** consistently increase memory usage across all models

These results indicate that **memory consumption is influenced not only by parameter count but also by architectural design and input dimensionality**.

Inference Throughput

Throughput analysis reveals:

- **ResNet-18** achieves the highest inference speed, exceeding **2400 images/sec**, making it suitable for real-time applications
- **EfficientNet-B0** provides higher accuracy but at significantly lower throughput
- **DeiT-Tiny** offers a balanced trade-off between speed and memory efficiency

This highlights a fundamental trade-off between **performance and inference speed**, which is critical in real-time malware detection systems.

Grayscale vs RGB Cost–Benefit Analysis

The comparison between grayscale and RGB representations reveals:

- RGB increases **input dimensionality by 3×**
- RGB leads to higher **VRAM consumption**
- RGB increases **data storage and loading overhead**
- RGB does **not provide consistent accuracy improvements** (Section 4.4)

These findings confirm that RGB encoding introduces additional computational cost without proportional performance benefits, making it less suitable for MIoT deployment scenarios.

Deployment Implications for MIoT

In practical MIoT environments:

A. Edge devices often operate under **limited memory and compute resources**
B. Real-time detection requires **low-latency inference**
C. Continuous monitoring demands **efficient and stable models**

Based on the experimental results:

- **EfficientNet-B0 (grayscale)** provides the highest accuracy but requires higher memory
- **ResNet-18 (grayscale)** offers excellent throughput and moderate resource usage
- **DeiT-Tiny** provides superior memory efficiency with acceptable performance

A practical deployment strategy may involve:

- CNN-based models for high-accuracy detection
- Lightweight models for edge deployment
- Hybrid approaches combining efficiency and robustness

Answer to RQ4

The experimental results demonstrate that:

1. Model selection must balance **accuracy, memory usage, and inference speed**
2. Grayscale representations are more **resource-efficient** than RGB
3. CNN-based architectures provide the best **accuracy-efficiency trade-off**
4. Transformer-based models offer **memory advantages** but slightly lower performance

Overall, grayscale CNN models emerge as the most suitable choice for MIoT malware detection systems, offering a strong balance between detection performance and computational feasibility.

4.8 Summary of Findings

This section summarizes the experimental results presented in Chapter 4 and provides consolidated answers to the research questions. The evaluation followed a structured pipeline, including baseline establishment, representation analysis, architectural comparison, adversarial robustness evaluation, and resource profiling.

Representation-Level Findings (RQ1)

The comparison between grayscale and RGB malware image representations demonstrates that grayscale encoding consistently achieves equal or superior performance.

Grayscale images preserve the original byte-level structure of malware binaries, enabling effective feature extraction by deep learning models [6][16]. In contrast, the computational costs of RGB images have been reported to be significantly higher than

grayscale images in malware detection [10][17], resulting in increased computational cost without performance gains.

Therefore, grayscale representation is both **more efficient and more suitable** for malware classification in MIoT environments.

Architecture-Level Findings (RQ2)

The evaluation of different architectures reveals a clear performance hierarchy:

A. **EfficientNet-B0** achieves the highest classification accuracy, confirming the effectiveness of modern CNN architectures for malware image analysis [8][10]
B. **DeiT-Tiny** provides competitive performance but shows sensitivity to class imbalance
C. **ResNet-18** offers a solid baseline with lower overall performance
D. **Row-LSTM** fails to capture discriminative spatial features
   Autoencoders provide stable representations but do not perform classification independently

These findings indicate that malware images primarily encode **local spatial patterns**, which are best captured by convolutional neural networks [8].

Adversarial Robustness Findings (RQ3)

The adversarial evaluation highlights significant vulnerabilities in deep learning–based malware classifiers.

CNN-based models exhibit rapid performance degradation under gradient-based perturbations, consistent with prior research on adversarial attacks in malware detection [11][12]. Transformer-based models demonstrate improved robustness under low perturbation levels, likely due to their global attention mechanisms [14].

However, no evaluated model remains stable under strong adversarial conditions, indicating that **clean accuracy alone is not sufficient** for assessing deployment reliability.

Efficiency and Deployment Findings (RQ4)

Resource profiling reveals important trade-offs between performance and computational efficiency.

- **EfficientNet-B0** provides the highest accuracy but requires higher memory
- **ResNet-18** achieves the highest inference speed
- **DeiT-Tiny** offers the lowest memory footprint

RGB representations increase computational cost without improving performance, making them unsuitable for resource-constrained environments. These findings align with the requirements of MIoT systems, where efficient and reliable detection must operate under limited computational resources [1][2].

Overall Conclusions

The experimental results support the following key conclusions:

1. Grayscale malware images provide the best balance between performance and efficiency
2. CNN-based architectures are the most effective for malware image classification
3. Transformer-based models offer improved robustness but slightly lower performance
4. Sequential modeling approaches are ineffective for this task
5. Deep learning models are highly vulnerable to adversarial attacks
6. Efficiency considerations are critical for real-world MIoT deployment

Final Insight

The findings of this chapter establish a **cost–benefit–robustness trade-off framework** for malware detection in MIoT environments. While high-performance models such as EfficientNet-B0 achieve near-perfect accuracy under clean conditions, their vulnerability to adversarial attacks and higher resource requirements highlight the need for balanced model selection.

Future MIoT malware detection systems should therefore integrate:

A. Efficient grayscale-based representations
B. Robust model architectures
C. Adversarial defense mechanisms
D. Resource-aware deployment strategies

This holistic perspective is essential for developing reliable and scalable AI-powered security solutions in modern IoT healthcare ecosystems.

5 Discussion

5.1 Impact of Image Representation on Malware Classification

This study investigated the role of image representation in malware-as-images classification pipelines for Medical Internet-of-Things (MIoT) environments. Malware binaries were converted into visual representations and evaluated using both grayscale and RGB encodings. The experimental results show that grayscale representations provide strong classification performance while maintaining lower computational overhead.

In grayscale encoding, each byte value of the executable is mapped directly to a pixel intensity.[16] This representation preserves structural relationships within the binary and enables convolutional networks to detect repeated byte patterns corresponding to malware families.[7][8][16] Because grayscale images contain only a single channel, they also reduce input dimensionality and computational cost.

In contrast, RGB representations introduce three-channel inputs that increase data dimensionality. Although RGB encodings can sometimes improve classification performance in computer vision tasks, the experiments performed in this thesis demonstrate that malware images do not benefit significantly from this additional representational capacity. Malware images originate from byte intensity mappings rather than natural color distributions, and therefore the extra channels introduce redundancy rather than meaningful semantic information.[16][17]

The comparative experiments show that grayscale models achieve equal or slightly superior accuracy while requiring fewer computational resources. These findings are particularly important for MIoT deployments, where processing power and memory availability are limited.[1][2][9] Consequently, grayscale representations appear to offer the most practical trade-off between performance and efficiency for AI-powered malware detection in embedded environments.

5.2 Architectural Trade-offs Between CNNs, Transformers, RNNs and Autoencoders

The experimental evaluation compared several deep learning architectures, including convolutional neural networks (CNNs), vision transformers (ViTs), recurrent neural networks (RNNs), and autoencoder-based models. Each architecture introduces different inductive biases and computational requirements, which influence both classification performance and deployability.

CNN architectures demonstrated the strongest classification performance across the evaluated experiments. EfficientNet-B0 achieved the highest clean accuracy, indicating that convolutional filters effectively capture spatial patterns present in malware image representations. These patterns often correspond to code sections, padding regions, and repeated structural structures within executable binaries.[16]

Vision Transformers achieved competitive classification accuracy but exhibited slightly weaker performance on minority classes due to class imbalance in the dataset.[14] However, transformer-based models demonstrated improved robustness against adversarial perturbations compared to CNN-based architectures, particularly under low perturbation magnitudes.

Sequential architectures based on recurrent neural networks were evaluated using a Row-LSTM design in which malware images were interpreted as sequences of image rows. This formulation produced significantly lower classification accuracy compared to CNN and transformer models. The results suggest that malware images are dominated by spatial texture patterns rather than sequential dependencies, making convolutional processing more appropriate for this domain.[7][8][16]

Autoencoder-based models were evaluated from a representation-learning perspective. The convolutional autoencoder successfully learned compressed latent representations of malware images while maintaining low reconstruction error across training, validation, and test splits. Although the autoencoder does not directly perform classification, its compact representation capability makes it useful for anomaly detection and feature extraction.[10][20] More broadly, these findings reinforce the need for AI-based security pipelines that balance predictive performance, robustness, and deployment feasibility in real-world environments [22][27][29].

5.3 Adversarial Robustness and Security Implications

A critical requirement for malware detection systems operating in real-world environments is robustness against adversarial manipulation. Attackers may attempt to modify malware binaries or their visual representations in order to evade machine learning detectors while preserving malicious functionality.[11][12]

This thesis evaluated model robustness using gradient-based adversarial perturbations and examined the impact of adversarial noise on classification accuracy. The experiments reveal that convolutional neural networks experience significant performance degradation when subjected to adversarial perturbations. Even small perturbation magnitudes can substantially reduce classification accuracy.

In comparison, transformer-based architectures demonstrated improved stability under low perturbation levels, maintaining higher accuracy than CNN-based models in adversarial conditions. This suggests that self-attention mechanisms may capture broader contextual relationships that are less sensitive to localized perturbations.

Beyond gradient-based attacks, generative adversarial networks (GANs) represent a powerful method for generating adversarial malware samples capable of bypassing detection systems. GAN-generated samples can preserve malware functionality while manipulating visual representations to confuse classifiers.[13] This highlights the importance of incorporating robustness evaluation into AI-powered malware detection pipelines.

Recent advancements in adversarial AI defense frameworks suggest that future systems should integrate adaptive and platform-agnostic protection mechanisms capable of detecting and mitigating attacks in real time [22][27][28][29]. Additionally, adversarial training approaches have been proposed to improve resilience by exposing models to adversarial examples during training [27][29]. For MIoT security, these findings emphasize that high classification accuracy alone is insufficient. Malware detection systems must also be resilient to adversarial manipulation in order to remain reliable in hostile environments.[22][27][28]

5.4 Efficiency and Deployment in Medical IoT Environments

Medical IoT systems impose strict constraints on computational resources, including limited memory capacity, restricted processing power, and strict latency requirements.[1][2][9] As a result, malware detection solutions must balance detection performance with computational efficiency.[2][9][10]

The resource profiling experiments show that different architectures exhibit significant differences in memory consumption, parameter count, and inference throughput. EfficientNet-B0 provides the highest classification accuracy but requires more memory compared to simpler convolutional architectures.

Transformer-based models such as DeiT-Tiny demonstrate lower memory usage but slightly reduced classification accuracy. This trade-off may still be acceptable in scenarios where memory constraints are the primary limitation.

The autoencoder model represents the most lightweight component in the evaluated system. With a very small model size, the autoencoder can serve as a lightweight anomaly detection module deployed directly on edge devices or gateways. Suspicious binaries identified by the autoencoder can then be forwarded to more complex classifiers for detailed analysis.

These findings suggest that layered detection architectures combining lightweight anomaly detection with high-accuracy classifiers may provide the most practical approach for MIoT malware defense.

5.5 Limitations and Research Directions

While the results presented in this thesis demonstrate the potential of image-based deep learning for malware detection, several limitations must be acknowledged.

First, the evaluation relies primarily on publicly available datasets. These datasets may not fully represent the diversity of malware targeting Linux-based medical IoT environments. Future research should explore larger datasets containing real-world IoT malware samples.[9][10][19][20]

Second, adversarial robustness evaluation focused on gradient-based perturbations. Additional adversarial scenarios, including GAN-based malware generation and binary-level evasion techniques, should be investigated in future work. [12][13][22][27][29]

Finally, the experiments were conducted on GPU-enabled research infrastructure rather than directly on embedded MIoT hardware. Future studies should evaluate the performance of these models on real IoT devices and edge gateways to better understand deployment constraints.[1][2][9]

Despite these limitations, the findings of this thesis demonstrate that deep learning based malware image classification can provide a promising foundation for AI-powered antivirus systems in medical IoT environments.

6 Conclusion and Future Work

6.1 Summary of Findings

This thesis investigated the feasibility of image-based deep learning approaches for malware detection in Medical Internet-of-Things (MIoT) environments. By transforming malware binaries into image representations, the study leveraged modern computer vision architectures to perform classification without relying on handcrafted features.

The experimental evaluation demonstrated that convolutional neural networks remain highly effective for malware image classification. In particular, EfficientNet-B0 achieved the highest clean accuracy across the evaluated models, highlighting the ability of CNN architectures to capture structural patterns embedded in malware binaries. Vision Transformer models such as DeiT-Tiny achieved competitive accuracy while demonstrating comparatively stronger resilience under low adversarial perturbations.

A systematic comparison between grayscale and RGB representations revealed that grayscale inputs provide equal or superior classification performance while significantly reducing computational overhead. Since malware images originate from byte intensity mappings rather than natural color distributions, additional RGB channels do not necessarily introduce meaningful discriminative information. As a result, grayscale representations offer a more efficient and practical solution for MIoT-oriented malware detection pipelines.

Additional architectural experiments further highlighted that sequential row-based models such as Row-LSTM are ineffective for capturing the spatial structures present in malware images. Conversely, convolutional autoencoders demonstrated stable reconstruction behavior and learned compact latent representations of malware images, suggesting their usefulness for anomaly detection or feature compression tasks.

Resource profiling experiments emphasized the importance of balancing classification performance with computational efficiency. While EfficientNet-B0 achieved the strongest predictive performance, lighter architectures such as ResNet-18 and DeiT-Tiny offer advantages in terms of inference throughput and memory consumption. These trade-offs are particularly relevant in MIoT environments, where malware detection must operate within strict resource constraints.

Finally, adversarial robustness evaluation revealed that deep learning malware classifiers remain vulnerable to gradient-based perturbations. Convolutional models exhibited rapid degradation under adversarial noise, whereas transformer-based models retained higher accuracy under low perturbation magnitudes. These results

highlight the necessity of incorporating robustness evaluation when designing AI-powered malware detection systems.[22][27][29]

6.2 Contributions of the Thesis

The primary contributions of this thesis can be summarized as follows:

- **a)** A reproducible malware-as-images pipeline for converting executable binaries into grayscale and RGB image representations suitable for deep learning models.
- **b)** A systematic comparison of grayscale and RGB malware image encodings in terms of classification accuracy, computational efficiency, and deployment suitability.
- **c)** Benchmarking of multiple deep learning architectures, including CNNs, Vision Transformers, autoencoders, and sequential models, under a unified experimental protocol.
- **d)** Adversarial robustness evaluation using gradient-based perturbation methods to assess the resilience of malware image classifiers.
- **e)** A cost–benefit analysis connecting detection accuracy, robustness, and resource efficiency within the context of MIoT malware detection.

6.3 Limitations

Despite the contributions of this work, several limitations must be acknowledged. First, the experiments rely primarily on publicly available datasets such as Malimg, which consist mainly of Windows-based malware samples. While these datasets provide valuable benchmarks, they may not fully represent the diversity of Linux-based malware commonly found in IoT and MIoT ecosystems.

Second, the adversarial robustness evaluation focuses on gradient-based perturbations applied to image representations. Other evasion techniques, including advanced binary-level obfuscation strategies or adaptive adversarial attacks, were not investigated within the scope of this thesis.[12][13][27][29]

Finally, efficiency measurements were conducted on GPU-enabled research infrastructure rather than directly on embedded MIoT hardware. Consequently, real-world deployment performance may differ depending on the computational capabilities of the target platform.

6.4 Future Research Directions

The findings of this thesis open several promising research directions that can further advance AI-based malware detection in Medical Internet-of-Things (MIoT) environments.

First, future work should focus on the development and utilization of larger, more diverse, and Linux-oriented malware datasets.[9][10][19][20] The current reliance on predominantly Windows-based datasets limits the generalizability of experimental findings. Expanding dataset diversity to include real-world IoT and MIoT malware samples would enable more realistic and representative evaluation of detection systems, particularly for embedded Linux platforms.[9][10]

Second, enhancing adversarial robustness remains a critical research priority. While this study evaluates vulnerability under gradient-based perturbations, future approaches should incorporate adversarial training strategies, robust optimization techniques, and hybrid defense mechanisms.[22][27][28][29] Combining deep learning classifiers with anomaly detection modules or real-time monitoring frameworks could significantly improve resilience against adaptive and evolving adversarial threats.[22][27][28]

Third, the design of lightweight and resource-efficient deep learning models is essential for practical deployment in MIoT environments. Given the strict computational constraints of embedded devices, future research should explore model compression techniques such as pruning, quantization, and knowledge distillation. These approaches can reduce model size and inference latency while preserving high detection performance, enabling deployment at the edge or gateway level.

Fourth, an additional promising direction is the integration of explainable artificial intelligence (XAI) techniques to improve transparency and trust in AI-based malware detection systems.

Beyond technical challenges, the deployment of AI-based cybersecurity systems must also consider regulatory and ethical dimensions. This is particularly important in healthcare contexts, where data privacy, system accountability, and safety are critical. Recent work highlights the emerging European regulatory ecosystem for ethical AI and its implications for secure system design [24], emphasizing the need for alignment between technical innovation and regulatory compliance.

In summary, this thesis demonstrates that image-based deep learning constitutes a scalable and effective approach for malware detection in MIoT ecosystems. Future systems should aim to combine efficient data representations, robust and adaptive learning architectures, and resource-aware deployment strategies, while also addressing adversarial threats and regulatory requirements. Such a holistic approach is essential for strengthening the security posture of next-generation medical IoT infrastructures.

References

[1] Sun, W., Cai, Z., Li, Y., Liu, F., Fang, S., & Wang, G. (2018). Security and Privacy in the Medical Internet of Things: A Review. *Security and Communication Networks*, 2018, Article ID:5978636. https://doi.org/10.1155/2018/5978636

[2] Mohammadi, M., Al-Fuqaha, A., Sorour, S., & Guizani, M. (2018). Deep learning for IoT big data and streaming analytics: A survey. *IEEE Communications Surveys & Tutorials*, 20(4), 2923–2960. https://doi.org/10.1109/COMST.2018.2844341

[3] Pa, Y., et al. (2015). IoTPOT: Analysing the Rise of IoT Compromises. *Proceedings of the 9th USENIX Workshop on Offensive Technologies (WOOT '15)*. USENIX Association. IoTPOT: Analysing the Rise of IoT Compromises

[4] Kolosnjaji, B., et al. (2016). Deep Learning for Classification of Malware System Call Sequences. *29th Australasian Joint Conference on Artificial Intelligence(AI 2016)*,Springer Verlag. https://users.ics.forth.gr/~zarras/files/AI_2016_Deep.pdf

[5 Egele, M., Scholte, T., Kirda, E., & Kruegel, C. (2012). A Survey on Automated Dynamic Malware-Analysis Techniques and Tools. *ACM Computing Surveys*, 44(2), 1–42. https://doi.org/10.1145/2089125.2089126

[6] Ni, S., Qian, Q., & Zhang, R. (2018). Malware identification using visualization images and deep learning. *Computers & Security, 77*, 871–885. https://doi.org/10.1016/j.cose.2018.04.005

[7] Kalash, M., Rochan, M., Mohammed, N., Bruce, N. D. B., Wang, Y., & Iqbal, F. (2018). Malware Classification with Deep Convolutional Neural Networks. In *2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS 2018)* (pp. 1–5). IEEE. https://doi.org/10.1109/NTMS.2018.8328749

[8] Gibert, D., Mateu, C., Planes, J., & Vicens, R. (2019). Using convolutional neural networks for classification of malware represented as images. Journal of Computer Virology and Hacking Techniques, 15(1), 15–28. https://doi.org/10.1007/s11416-018-0323-0

[9] Ramamoorthy, J., Gupta, K., Shashidhar, N. K., & Varol, C. (2024). Linux IoT Malware Variant Classification Using Binary Lifting and Opcode Entropy. *Electronics*, 13(12), 2381. https://doi.org/10.3390/electronics13122381

[10] Qureshi, S. U., He, J., Tunio, S., Zhu, N., Nazir, A., Wajahat, A., Ullah, F., & Wadud, A. (2024). Systematic review of deep learning solutions for malware detection and forensic analysis in IoT. *Journal of King Saud University – Computer and Information Sciences*, 36(8), 102164. https://doi.org/10.1016/j.jksuci.2024.102164

[11] Kolosnjaji, B., Demontis, A., Biggio, B., Maiorca, D., Giacinto, G., Eckert, C., & Roli, F. (2018). Adversarial malware binaries: Evading deep learning for malware detection in executables. *Proceedings of the 2018 26th European Signal Processing Conference (EUSIPCO 2018)*, 533–537. https://doi.org/10.23919/EUSIPCO.2018.8553214

[12] Khormali, A., Abusnaina, A., Chen, S., Nyang, D., & Mohaisen, A. (2019). COPYCAT: Practical adversarial attacks on visualization-based malware detection. *arXiv preprint arXiv:1909.09735*. https://doi.org/10.48550/arXiv.1909.09735

[13] Mercaldo, F., Martinelli, F., & Santone, A. (2024). Deep Convolutional Generative Adversarial Networks in Image-Based Android Malware Detection. *Computers*, 13(6), 154. https://doi.org/10.3390/computers13060154

[14] Bavishi, S., & Modi, S. (2024). Accelerating Malware Classification: A Vision Transformer Solution. *arXiv preprint arXiv:2409.19461*. https://doi.org/10.48550/arXiv.2409.19461

[15] Kumar, K., Mandoria, H. L., & Singh, R. (2025). Efficient malware classification using transfer learning and stacked ensemble techniques. *International Journal of Mathematical, Engineering and Management Sciences*, 10(4), 913–930. https://doi.org/10.33889/IJMEMS.2025.10.4.044

[16]L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath. 2011. Malware images: visualization and automatic classification. In Proceedings of the 8th International Symposium on Visualization for Cyber Security (VizSec '11). Association for Computing Machinery, New York, NY, USA, Article 4, 1–7. https://doi.org/10.1145/2016904.2016908

[17] A. S. Bozkir, A. O. Cankaya, and M. Aydos (2019). Utilization and comparison of convolutional neural networks in malware recognition. In *Proceedings of the 2019 27th Signal Processing and Communications Applications Conference (SIU)*, Sivas, Turkey, 24–26 April 2019, pp. 1–4. https://doi.org/10.1109/SIU.2019.8806511

[18] A. S. Bozkir, E. Tahillioglu, M. Aydos, and I. Kara, "Catch them alive: A malware detection approach through memory forensics, manifold learning and computer vision," Computers & Security, vol. 103, p. 102166, 2021, doi:10.1016/j.cose.2020.102166.

[19] Abdelouahab Amira, Abdelouahid Derhab, Elmouatez Billah Karbab, and Omar Nouali. 2023. A Survey of Malware Analysis Using Community Detection Algorithms. ACM Comput. Surv. 56, 2, Article 40 (February 2024), 29 pages. https://doi.org/10.1145/3610223

[20] Aboaoja, F. A., Zainal, A., Ghaleb, F. A., Al-rimy, B. A. S., Eisa, T. A. E., & Elnour, A. A. H. (2022). Malware detection issues, challenges, and future directions: A survey. Applied Sciences, 12(17), 8482. https://doi.org/10.3390/app12178482

[21] Farao, A., Paparis, G., Panda, S., Panaousis, E., Zarras, A., & Xenakis, C. (2023). INCHAIN: A Cyber Insurance Architecture with Smart Contracts and Self-Sovereign Identity on Top of Blockchain. *International Journal of Information Security*.

[22] Petihakis, G., Farao, A., Bountakas, P., Sabazioti, A., Polley, J., & Xenakis, C. (2024). AIAS: AI-Assisted Cybersecurity Platform to Defend Against Adversarial AI Attacks. *Proceedings of the 19th International Conference on Availability, Reliability and Security (ARES)*, 1–7.

[23] Pantelakis, V., Bountakas, P., Farao, A., & Xenakis, C. (2023). Adversarial Machine Learning Attacks on Multiclass Classification of IoT Network Traffic. *Proceedings of the 18th International Conference on Availability, Reliability and Security (ARES)*, 1–8.

[24] Bolgouras, V., Zarras, A., Leka, C., Stylianou, I., Farao, A., & Xenakis, C. (2025). EU Regulatory Ecosystem for Ethical AI. *AI and Ethics, 5*(5), 5063–5080.

[25] Paparis, G., Zarras, A., Farao, A., & Xenakis, C. (2025). CRASHED: Cyber Risk Assessment for Smart Home Electronic Devices. *Journal of Information Security and Applications, 91*, 104054.

[26] Farao, A., Zarras, A., Voudouris, A., Paparis, G., & Xenakis, C. (2025). B2SAPP: Blockchain-Based Solution for Maritime Security Applications. *Frontiers in Computer Science, 7*, 1572009.

[27] Ziras, G., Farao, A., Zarras, A., & Xenakis, C. (2025). From Vulnerability to Resilience: Adversarial Training and Real-Time Detection for AI Security. *Array*, 100546.

[28] Petihakis, G., Farao, A., Bolgouras, V., Bountakas, P., Panou, A., Floros, E., et al. (2025). Cloud-Based Platform-Agnostic Adversarial AI Defense Framework. *Proceedings of the IEEE Conference on Network Function Virtualization and Software-Defined Networking (NFV-SDN)*, 1–6.

[29] Zarras, A., Kollarou, A., Farao, A., Bountakas, P., & Xenakis, C. (2025). Testing the Limits: Exploring Adversarial Techniques in AI Models. *PeerJ Computer Science, 11*, e3330.

[30] Ni, S., Qian, Q., & Zhang, R. (2018). Malware identification using visualization images and deep learning. *Computers & Security*, 77, 871–885. https://doi.org/10.1016/j.cose.2018.04.005

[31] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *Proceedings of the 38th International Conference on Machine Learning (ICML)*, vol. 139, pp. 10347–10357, 2021. Available at: https://proceedings.mlr.press/v139/touvron21a.html

[32] Ashawa, M., Owoh, N., Hosseinzadeh, S., & Osamor, J. (2024). Enhanced Image-Based Malware Classification Using Transformer-Based Convolutional Neural Networks. *Electronics*, 13(20), 4081. https://doi.org/10.3390/electronics13204081

[33] Ö. Aslan and A. A. Yilmaz, "A New Malware Classification Framework Based on Deep Learning Algorithms," in IEEE Access, vol. 9, pp. 87936-87951, 2021, doi: 10.1109/ACCESS.2021.3089586.

[34] El-Shafai, W., Almomani, I., & AlKhayer, A. (2021). Visualized Malware Multi-Classification Framework Using Fine-Tuned CNN-Based Transfer

Learning Models. *Applied Sciences*, 11(14), 6446. https://doi.org/10.3390/app11146446

[35] Vasan, D., Hammoudeh, M., & Alazab, M. (2024). *Broad learning: A GPU-free image-based malware classification (IMCBL)*. Applied Soft Computing, 154, 111401. https://doi.org/10.1016/j.asoc.2024.111401

[36]

[37] M. J. Awan, et al., "Image-based malware classification using VGG19 network and spatial convolutional attention,"

Electronics, vol. 10, no. 19, p. 2444, 2021. https://doi.org/10.3390/electronics10192444

[38] O'Shaughnessy, S., & Sheridan, S. (2022). Image-based malware classification hybrid framework based on space-filling curves. Computers & Security, 116, 102660. https://doi.org/10.1016/j.cose.2022.102660

[39] S. Seneviratne, R. Shariffdeen, S. Rasnayaka, and N. Kasthuriarachchi, "Self-Supervised Vision Transformers for Malware Detection," arXiv:2208.07049, Aug. 2022. https://arxiv.org/abs/2208.07049