Dijlah Journal of Engineering Science (DJES) Vol. 3, No. 3, August, 2026, pp. 296-315
ISSN: Printed: **3078-9656, Online: 3078-9664**, paper ID:206

A Comparative Analysis of Malware Classification Based on Visualization Techniques

Wisam K. Jummar[1], Ismail Taha Ahmed[2], Khattab M Ali Alheeti [3]
[1]Center for Continuing Education, University of Anbar, Ramadi, Iraq
[23] College of Computer Sciences and Information Technology, University of Anbar, Anbar, Iraq

Article Info

Article history:

Received: May,09,2026
Revised: Jul.,20,2026
Accepted: Aug.,19,2026

Keywords:

Malware Detection and Classification
Malware Visualization
Obfuscation Techniques
Hand Crafted Features
Cybersecurity

ABSTRACT

Malware identification and classification is an important research direction due to the increasing number of electronic devices, which has led to the proliferation of malware that makes it difficult to contain. This is a significant risk in the digital world. Malware identification and classification using traditional methods suffer from several limitations. Obfuscation techniques in static methods are a problem that hinders feature extraction, while the time required for dynamic methods is a barrier, in addition to the problems posed by evasion techniques. In this paper, we examine methods that use a new approach based on malware visualization, which helps overcome the problems faced by traditional methods. Research in this area is divided into three groups. The first group relies on hand crafted features, the second group relies on deep learning algorithms, and the third group adopts hybrid methods that combine both techniques. Through a careful analysis of these works, the methods and algorithms used, the strengths and weaknesses of each method were identified, and a set of solutions were proposed for developing robust classification systems.

Corresponding Author:

Wisam K. Jummar
Center for Continuing Education, University of Anbar
Ramadi, Iraq
Email: wisamkhalid6@uoanbar.edu.iq

1. INTRODUCTION

Malware is truncated for malicious software; it is a software code that is exploited to achieve certain gains by some people with different skills. Examples of this include making financial gains by stealing banking data for victims, or stealing important data such as political, commercial, or other data. Due to the recent advancements in technology and its penetration into most fields, such as the military, business, and medical fields, hacking operations have become increasingly widespread. To overcome attacks if they occur or prevent them from occurring in the first place, they must be thoroughly studied and analyzed to determine their type and calculate their potential impact, by understanding their basic function.

Malware analysis includes mainly two mechanisms: static analysis and dynamic analysis. Static analysis is a method by which code or the structure of a program is examined without executing it. This type of analysis is used to detect and confirm whether a file is malicious, find information about its function, and create basic collection of signatures[1]. The detection patterns utilized in static analysis include byte-sequence ngrams, control flow graph, string signature, syntactic library call and operational code (opcode) frequency distribution [2]. Dynamic analysis uses a different technique to analyze how malware works. These programs are executed in a controlled environment, such as a sandbox or virtual machine [3]. As with static analysis, dynamic analysis includes specific detection patterns, such as: tracking the information flow, function based feature, system calls and API calls [4].

Static analysis examines code without running it, so it is both safer and faster. However, it may be vulnerable to some evasion techniques. Dynamic analysis has a greater ability to detect new malware because it runs the file in a virtual environment, but this results in significant system slowdown. The speed at which malware is produced or mutated from existing types is very high, so the time and effort required to analyze these programs

using either method is a significant drawback. Moreover, once effective countermeasures are developed, the malware will use this time to mutate into other forms, making it harder to detect. For these reasons, we need new methods for analyzing and detecting malware, and these methods must be robust and fast.[5].

Recently, a new malware classification technique was proposed by[6]. Instead of focusing on invisible features for classification, this technique utilizes vision-based malware analysis. This approach relies on image processing and is known as malware visualization. This work converted the structure of packed binary samples into images that are two-dimensional grayscale. The generated images showed specific texture patterns that were used to identify the underlying malware category. This method opened up new horizons in identifying malware and classifying its families. It was observed that the texture of the resulting images is very similar to software from the same family. Since then, many studies have been published in this regard. The use of this method has many advantages, as the processes of obfuscating the binary code do not significantly change the texture of the resulting images, making this method somewhat resistant to obfuscation. In addition, it is faster than traditional methods and does not require human expertise to analyze and extract the features required by static and dynamic methods. Figure 1 shows an overview of malware detection methods.

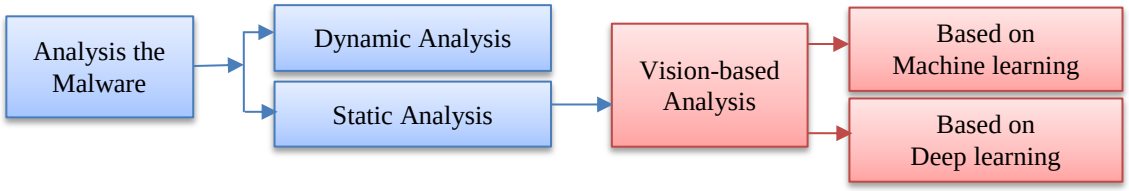


Figure 1. Overview of malware detection and classification techniques

The aim of this research was to review and categorize research papers in the field of malware identification and classification, specifically using visualization methods, and compare them with each other. This extensive reading led to the division of research in this field into three categories: The first category relied on manual methods to extract features from malware images, as these methods rely primarily on human expertise to set up the system. The second category adopted a different approach to extracting features automatically, without the intervention of experts as in the first category. Here, they used deep learning, a powerful tool for extracting features from images with minimal human intervention. To address the shortcomings or weaknesses of both categories and capitalize on their strengths, some researchers in the third category opted for feature extraction using a hybrid approach that combines both manually extracted features and those extracted using deep learning.

The main contributions of our work are as follows:

1. This work divides research in the field of visualization-based malware identification and classification into three categories. Researchers can focus on the category appropriate for their research.
2. The strengths and weaknesses of each research are extracted, allowing the reader to build a robust recognition system.
3. The research identifies the databases used in the training and testing process. This facilitates the researcher's selection of appropriate datasets for their system, and also enables them to communicate with researchers to utilize datasets specifically built for certain systems.
4. The work identifies the tools, algorithms and techniques used in each research, facilitating researchers' restructuring and development of these systems.

The rest of this paper is organized as follows: In Section 2 presents the most important standard databases used in training and evaluating systems; in Section 3, malware detection techniques are discussed through malware visualization and a comprehensive literature review. This section is divided into three main parts: malware detection by hand crafted features, using deep learning, and hybrid methods. Finally, we concludes our work and make an perspective for the future work.

2. EXISTING PUBLICLY DATASETS

There are many standard dataset used by researchers in this field, these datasets are generally available for use. Some researchers have prepared a datasets specifically for their research. Here we provide an overview of some of the most popular dataset used in evaluating scientific research.

The Malimg dataset [6] Includes 9,339 samples of malware. This dataset consists of 25 malware categories, with each sample belonging to one of these categories. It's worth noting that the number of samples in the categories varies from one category to another within this dataset. We notice from Figure 2 that the Malimg dataset is not balanced.

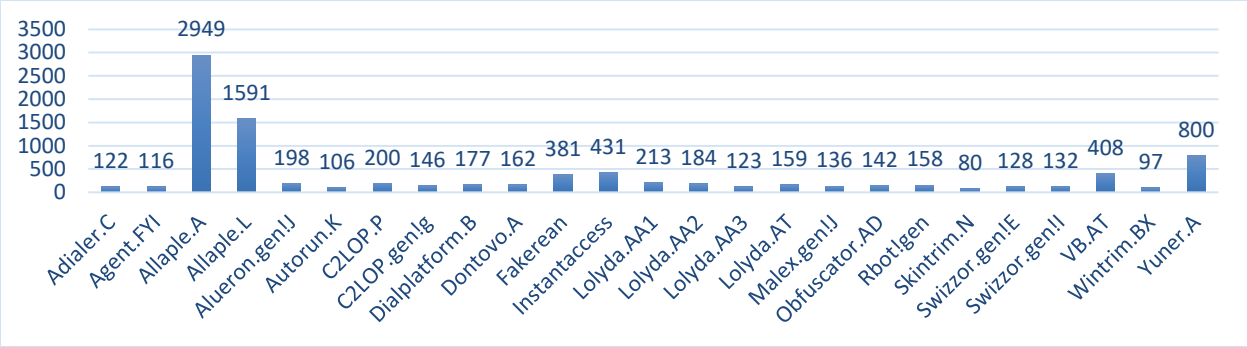


Figure 2. Distribution of Malware Classes in Malimg Dataset

The Malevis dataset [7], this dataset also includes 25 malware categories, and is divided into 9,100 training samples and 5,126 test samples. As can be seen in Figure 3, each of these classes includes 350 training samples in addition to various testing samples. It also includes a category for benign samples called (Other), used to distinguish between malware and benign samples.

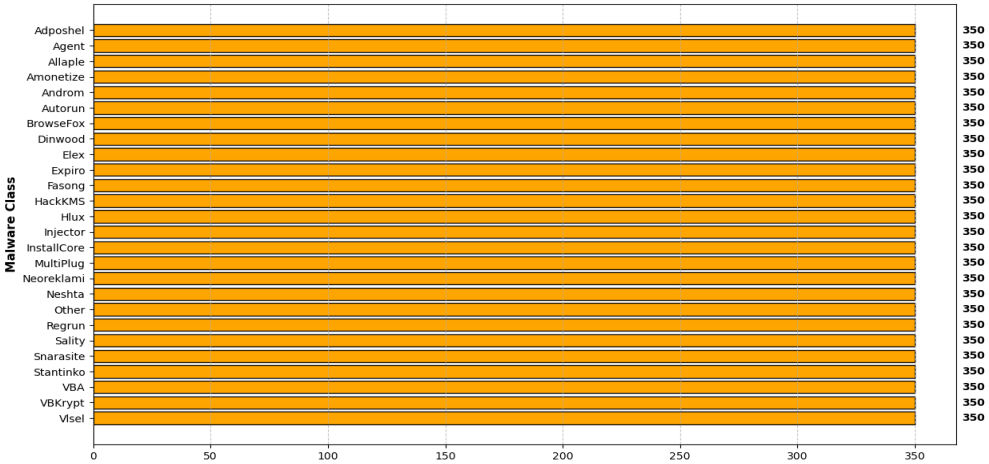


Figure 3. Distribution of Malware Classes in MalVis Dataset

In 2015, the Microsoft Malware Classification Challenge [8], was announced, releasing a dataset called (BIG 2015) containing nine malware families containing 10,868 samples. Since then, this dataset has become a standard for malware detection and behavior modeling research, it consists of a set of known malware files.[9]. As shown in Figure 4, the (Big 2015) dataset is unbalanced, and this must be addressed to avoid potential overfitting.

Although there are many public datasets used for testing malware detection systems, some researchers have gone to build private datasets to which they apply their own detection system, as we see in Tables (1, 2, 3).

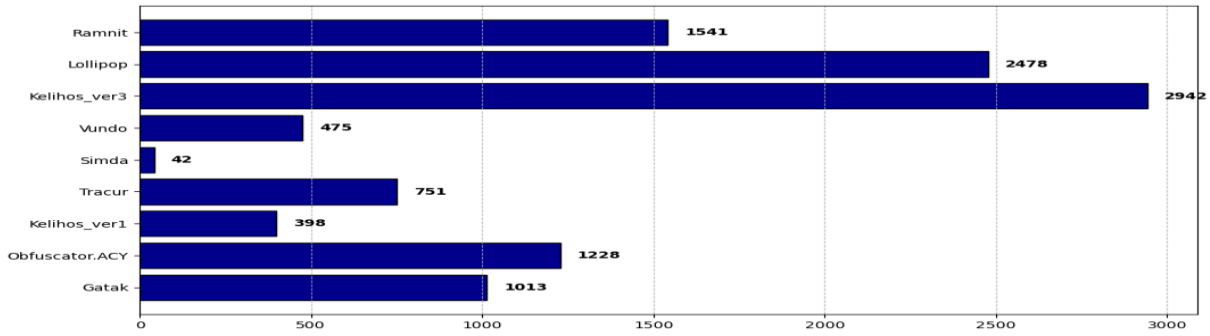


Figure 4. Distribution of Malware Classes in (BIG 2015) Dataset

3. EXISTING MALWARE DETECTION ALGORITHMS

The field of malware detection has undergone a significant shift, with many innovative techniques being used to counter the rapid and increasing sophistication of this software. Among the most important developments that have received widespread attention is the use of image visualization to detect malware. With this approach,

malware binary files are converted into visual representations, such as grayscale or color images, enabling advanced image processing or machine learning algorithms to be used for analysis.

When reviewing the literature and examining research papers in this field, will find that they fall into one of three categories: 1. Works that hand crafted features from malware images. 2. Works that use deep learning to extract features. 3. Works that adopt a hybrid approach that combines both of the above methods.

Each of these methods has advantages and disadvantages, and this will be explained in the literature review for each of these categories.

3.1. Malware detection and classification algorithms based hand-crafted features

Detection and classifying malware utilizing hand-crafted features extracted from malware images, It is considered an innovative approach, that leverages image processing and machine learning techniques to detect malicious software. First step is to converting binary files into images, then extracting meaningful features from them, that used to train a classifier.

While reading the literature in this field, it was noticed that many researchers convert malware to grayscale images, while others rely on RGB images. While most of the works are based on extracting features in the spatial domain of images, few works were found depends on the frequency domain of images.

Despite the benefits of this method, there are several challenges: the process of designing and engineering effective features requires domain expertise. Also, hand-crafted features may not generalize well to new malware. In addition, techniques like packing or encryption can used to evade detection.

[6], were the first to propose visualizing malware as grayscale images. Firstly the malware executables are represented for binary strings, after that reshaped as a matrices, that visualized as an images. Image processing techniques are apply to extract the features, this allow for categorization in effective way without the need for disassembly or code execution. The results showed that this innovative method is about 40 times faster than current techniques. In addition to, this technique also showed resilience against common obfuscation methods (eg. section encryption), which is often used by malware creators to evade detection.

The research by [10], classify malware using image processing techniques, focusing on texture patterns. The study extract the features by Discrete Wavelet Transform (DWT), and employing machine learning classifiers, like K-Nearest Neighbors (KNN) and Support Vector Machines (SVM). The results exhibit high classification accuracy.

In [11], applying DWT to decompose the malware images into four levels, and extract 11 statistical features to utilize in classification. A Support Vector Machine (SVM) used as classifier to distinguish between malware classes. This method achieve high accuracy rates on different datasets.

In [12], researchers have presented a study to identify and classify Trojan malware using image processing techniques. Emphasis was placed on images texture pattern analysis. Utilized Gabor wavelet method, to extract key features, and employe K-nearest neighbors (KNN) and Support Vector Machine (SVM) for classification. The experimental results indicate an enhancement in classification accuracy and minimize in the false positive rate.

A novel visualization based approach proposed by [13] for Precise malware classification. The complexity of training is reduced by generating low-dimensional data. This data is generated by visualizing the malware as colored images, and utilize gray-level co-occurrence matrix GLCM and color moments that capture global texture and color features. To extract local features, Simhash processes special byte sequences from malware code and data sections. Combining global and local features, this method achieves precise classification with low computational cost.

[14] proposed, a novel malware image classification system (MICS). This system designed for the IoT environment, that transform Potentially Malicious software into gray-scale images, and extracts the local texture features using D-SIFT (Dense Scale-Invariant Feature Transform), and global texture features using GIST, aimed at malware classification. MICS shows high classification accuracy. The results refer to that MICS is effective for processing IoT malware at a large scale, which providing a prospective solution for enhancing malware detection operations in connected devices.

In [15], presents a novel framework for malware classification by multi-layer learning, using a bag-of-visual-words (BoVW) model. The binary files of malware are transform into grayscale images, then extract the features by local descriptors such as LBP and Dense SIFT. The proposed method shows resilience against packed malware, but confronts challenges with samples that obfuscated or encrypted.

In [16], submit study that propose an accurate and fast model for identifying malware variants. Malware binary files are transformed into image of grayscale format, and use both local and global patterns for classification. Local features are extracted from edges, while extract global features are depends on the texture. The computational

efficiency enhancing by applying dimensionality reduction techniques. Experimental results Indicate that the presented model surpasses state-of-the-art techniques in both classification accuracy and response time.

A novel malware approach for malware detection introduced by [17], using binary visualization and incremental neural networks with self-organization feature. The method foxed on detection ransomware in .pdf and .doc files. It also performed well in detecting malicious code from other formats and file types, identify efficiently in real-time unknown malware.

Through [18], proposes a new binary texture analysis technique for classifying the malware via transforming the executable files into greyscale images, then extracting first-order and second-order texture features, that derived from the grey-level co-occurrence matrix (GLCM). The method obtains high accuracy, speed, and efficiency, with an average classification time of 0.01 seconds. This technique shows robustness facing code obfuscation and provides a cost-effective solution for malware classification in large-scale.

In [19], introduce a novel system for malware classification depends on visual representations. The system extracts handcrafted features from the images (e.g., contrast, entropy, correlation), and utilize a hierarchical ANN architecture for the classification process. The first level conducts approximate classification, while the second level improves the classification for confusing families of malware. The proposed method stands out in speed and simplicity, making it appropriate for real-time applications.

A new malware detection framework presents by [20], that merges malware visualization, feature extraction automatically, and collaborative learning. malware binaries transformed into grayscale images, the framework enables effective feature extraction using Local Binary Pattern (LBP) extractor. The framework utilizes a collaborative learning approach, and noise learning theory, to improve classification accuracy by learning from both labeled and unlabeled samples. Experimental results exhibits that the proposed method accomplishes high performance in classification, while showing resilience to data imbalances.

In [21], concentrates on classification of malware by using a new method which integrates the Segmentation-based Fractal Texture Analysis (SFTA) and Gaussian Discriminant Analysis (GDA). The study extract the features by apply image processing techniques after converts malware samples into images. The proposed method achieves a high accuracy in classification, exceeding existing advanced techniques.

Hybrid framework was introduced by [22] In order to classify malware. In this framework, use space-filling curves SFCs to converts both static executable malware files and dynamic process memory dumps into image formats. Following that analyzing these images for visual features using (Histogram of Oriented Gradients HOG, Gabor filters, and LBP), which enable strong classification without the need for reverse engineering. The results showed significant improvements in performance over standalone static or dynamic methods.

In [23], presents a new framework based on subspace representation techniques for malware visualization and classification. The proposed method leverages malware grayscale image patterns for efficient analysis. The framework employs Subspace Methods (SM), and Kernel Subspace Methods (KSM), to achieve robust classification. The approach is achieving high accuracy, however using the kernel methods, still be computationally intensive, especially with large datasets or high-dimensional feature.

The researchers in [24], proposed malware classification approach. Firstly malware conversion to grayscale images, next feature extraction by (SFTA, LBP, Haralick, Gabor, Tamura), finally classification by trying multiple classifier (GDA, KNN, Logistic, SVM, RF, Ensemble). The SFTA-KNN and Gabor-KNN methods outperforming current state-of-the-art approaches, and reduced computational demands.

Table 1. Hand-Crafted Features Extraction Algorithms

Authors/year	Spatial /Transform Domain	Methods	Dataset	Accuracy	advantages	limitations
Nataraj et al/ 2011 [6]	Spatial Domain	Image processing techniques, GIST, KNN	Malimg	98%	1. Innovative Visualization Approach. 2.High Classification Accuracy. 3.Efficiency and Speed: about 1.4 seconds.	1.Vulnerability to Countermeasures: manipulate the binary structure of malware to evade detection. 2.Global Feature Dependence. 3.Potential for Misclassification. 4.Limited Dataset Scope.
A Makandar, A Patrot/ 2017 [10]	Transform Domain	Gabor Wavelet, DWT, GIST, KNN and SVM	Malheur and Malimg	98.88% for Malimg	1.High Classification Accuracy. 2.Reducing the number of feature vectors, which enhances computational efficiency. 3.Effective	1.Dataset constraints: may affect generalizability. 2.May struggle with metamorphic malware.

					Feature Extraction.	
A Makandar, A Patrot/ 2017 [11]	Transform Domain	DWT, SVM	Malheur and Malimg	92.53% for Malimg, 91.05% for Malheur	1.Innovative Use of Wavelet Transform.2.High Classification Accuracy.	1. Dataset Limitations. 2. Feature Selection: may not capture all relevant aspects of malware behavior. 3.The computational complexity could hinder scalability.
Makandar & Patrot/ 2018 [12]	transform domain	Gabor Wavelets, KNN, SVM	Malimng 9 families of Trojan	89.11%	1.Innovative Approach. 2.Effective Feature Extraction. 3.Comparative Analysis.	1.Limited Dataset. 2.Focus on Trojan Malware.
Fu et al/2018 [13]	Spatial Domain	GLCM and color moments, Simhash, RF, K-NN, and SVM	7087 samples from 15 families	97.47%	1.Novel Visualization Method.2.Combination of Features. 3.Robustness Against Variants. 4.Low Computational Cost.	1.Requires that the structure of malware can be parsed. 2.Poor adaptability of the to packed malware.
Naeem et al/ 2018 [14]	Spatial Domain	D-SIFT, GIST, Bag of Features (BOF), PCA, SVM	9339 unpacked samples belonging to 25 windows malware families.	97.4%	1. Novel Framework: for the IoT environment. 2. High Classification Accuracy. 3. computational efficiency.	1.Dataset Limitations: Windows malware samples, which may limit the generalizability of the findings to other operating systems. 2.Lack of Real-Time Testing. 3.Scalability Concerns.
Liu et al/ 2019 [15]	Spatial Domain	LBP, Dense SIFT, Multi-layer Learning Framework, RF, KNN.	MalingA 9,458, MalingB 12,278 & malware dataset from the CNCERT has 15,000 malware binary files	99% with MalingA. And 97.4% MalingB	1.Enhanced Classification Accuracy. 2.Innovative Feature Extraction Approach.	1.Higher Computational Cost: processing times three days on the MalingB. 2.Sensitivity to Obfuscation and Encryption. 3.Parameter Sensitivity.
Naeem et al /2019 [16]	Spatial Domain	LGMP, D-SIFT and GIST, conventional bag of features, Gaussian weight, K-NN, SVM, naive Bayes (NB)	Malimg, Malheur, VirusShare, and Microsoft Big 2015	98.4% for Malimg	1.Dependence on Training Data Size. 2.Complexity of Feature Integration of local and global features.	the model is slightly slower than the other models
Baptista et al/2019 [17]	Spatial Domain	Self-Organizing Incremental Neural Networks (SOINN), SFCs.	2K benign files, 2K malicious files	91.7% ransomware in .pdf files, 94.1% for .doc	1.Innovative Approach. 2.High Detection Accuracy. 3.The algorithm is designed to be lightweight and fast.	1.Limited Dataset Diversity 4,000 samples. 2.Sensitivity to Evasive Techniques. 3.Performance Variability.
Verma et al/ 2020 [18]	Spatial Domain	histogram equalization, GLCMs, RF	DB1:Malimg, DB2:2916 Windows malware samples	98.58% for DB1 and 98.11% for DB2	1.Cost- and Time-Effective Method: average classification time of 0.01 seconds. 2.High Accuracy.	1.Evasion by Embedding: by embedded within larger legitimate binaries. 2.less effective for detecting unknown malware.
Moussas & Andreatos/ 2021 [19]	Spatial Domain	Artificial Neural Networks (ANNs), correlation matrix	Malimg	99.13%	1.Novel Two-Level ANN Architecture. 2.High Accuracy. 3.Simplicity and Speed. 4.Handling of Confusing Families.	1.Dataset size and diversity: limiting the generalizability of the findings. 2.Features may not accurately represent the most complex malware. 3.Limited comparison with other systems
Gao et al/2021 [20]	Spatial Domain	LBP, the average grid gray intensity, The Tri-training algorithm, (CCA), RF, KNN, and LR	Malimg and Microsoft Big 2015	97.95% for Malimg, 94.09% for Microsoft	1.Resistant to data imbalances. 2.Improves the classification ability. 3.Reduces the noise impact caused by the lack of labeled samples.	1.Increased Training Time. 2.Dependence on Unlabeled Samples may introduce noise and affect overall accuracy. 3.Neural networks can lead to high training costs. 4-Obfuscation techniques can distort the extracted features
Ahmed et al/ 2022 [21]	Spatial Domain	SFTA and Gabor filters, GDA, NB	MaleVis	98%	1.High Classification Accuracy. 2.Effective Feature Extraction.	1.Evaluated solely using one dataset, which limit the generalizability of the results.

					3.Reduced Computational Costs.	2.Requires considerable effort in feature extraction.
O'Shaughnessy & Sheridan/ 2022 [22]	Spatial Domain	SFCs, LBP, Gabor Filters, HOG, KNN, RF, Decision Tree (DT)	A special 13,599 samples, 23 families. without benign files	97.0%	1.Novel Hybrid Framework. 2.Effective Feature Representation. 3.Robustness Against Obfuscation.	1.There is no testing performed using a holdout dataset. 3.Dynamic analysis produce false positives, it also consumes a lot of resources and time.
Benchadi Djafer Yahia et al/ 2023 [23]	Spatial Domain	SM, KSM, Occlusion Sensitivity Analysis (OSA).	Malimg, Dumpware	98.07% Malimg, 97.21% Dumpware	1.Innovative Framework. 2.High Classification Accuracy.	1.Limited Dataset Diversity. 2.Non-linear Complexity: may introduce additional computational overhead. 3.The interpretability of the results may still be limited.
Ahmed, et al/ 2024 [24]	Spatial Domain	SFTA), (LBP), Haralick, Gabor, Tamura, GDA, KNN, Logistic Regression LR, SVM, RF, and Ensemble learning	Malimg & MaleVis	96.29% for Malimg, 98.02% for MaleVis	1.Comparative Analysis: to identify optimal candidates for malware detection. 2.Enhanced Accuracy. 3.Robustness to Dataset Imbalance.	1.High Dimensionality: processing overhead and increased resource consumption. 2.Feature Engineering Challenges. 3. Discover malware, not classifying its families. 4. Benign softwares are little through the test stage.

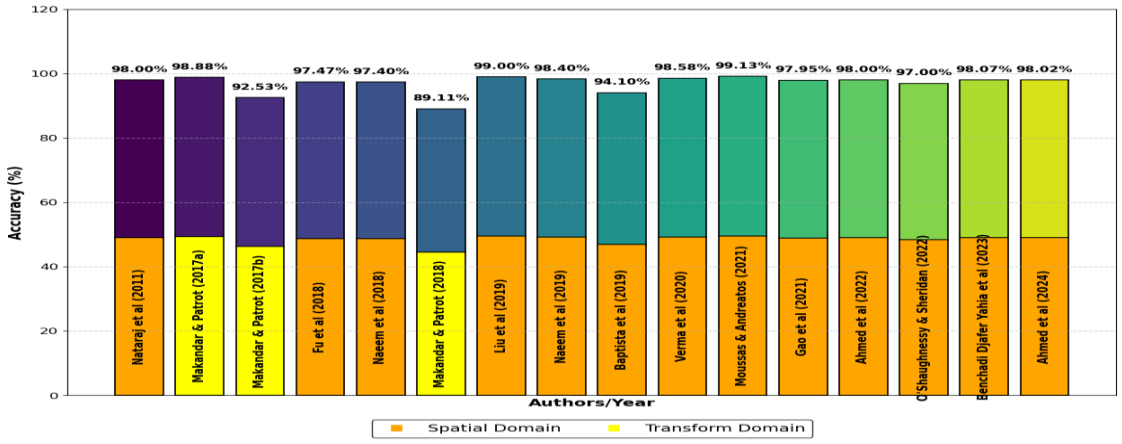


Figure 5. Comparison of accuracy between works utilizing handcrafted features

3.1.1. Comparison of malware detection with handcrafted features

Evolution the techniques of malware detection that utilizing transformation of the binary to image demonstrates considerable advancements in both computational efficiency and classification accuracy. Many studies have been reviewed and analyzed, and through the analysis of these studies, the potential that can be achieved by applying the image visualization methodology can be highlighted as an important means of extracting features in both the spatial and transform domains using various tools and algorithms. Many algorithms have demonstrated excellent performance, with some achieving over 97% accuracy when tested on benchmark datasets such as Malimg and MaleVis. On the other side, some drawbacks exist including lack of diversity in the dataset, high sensitivity to the obfuscation techniques, high computation cost and challenges in being applied to real life settings.

If we look at the works that achieved the highest accuracy results, they are as follows:

The classification rate in [10] was 98.88%: This work reduced the number of feature vectors, which enhanced computational efficiency and resulted in high classification accuracy. However, it did not use all samples in the dataset, but rather a small portion (1,610 sample) from the Malimg dataset. The research did not report any results on the Malheur dataset. Therefore, it is difficult to adopt these results generally and apply them to real-world applications.

In[15], the classification rate of 99%: A novel framework was proposed to enhance feature extraction and improve classification accuracy. Although the system was trained on three datasets, it achieved mixed results. It achieved 99% accuracy on the MalingA dataset and 97.4% on the MalingB dataset, while no results were reported

on the third dataset. However, this system can be considered to have good generalizability, as it has been tested on multiple databases.

The classification rate in[19] was 99.13%: Despite the innovative design, simplicity, and high speed of this system, its implementation relied on a single dataset, Malimg. This raises questions about the generalizability of the results to other datasets or its real-world use, as well as the limited comparison with other methods. Furthermore, the Malimg database is unbalanced, which could lead to model bias or overfitting.

Despite the achievements of research that adopted hand crafted features, there are still critical challenges that have not been addressed, such as:

1.Vulnerability to Evasive Techniques: this reduces the robustness of detection systems to some extent due to obfuscation and encryption by the adversary.

2.Scalability: real-time processing is difficult to achieve due to the large dimensions of the feature space and the computational cost.

3.Dataset imbalance and diversity: Overreliance on certain datasets can weaken the model's scope.

4. Most of the researches rely on global features of the texture, this may produce an unstable model that can only be applied to a small number of samples.

These challenges and problems are noteworthy and should be addressed systematically. Therefore, we propose the inclusion of key features to create an advanced framework, these features include:

1.Further Enhance Feature Representation: using features that combine spatial domain features with transform domain features, will help combat obfuscation techniques.

2.Enhance Robustness:

A. Use the techniques of adversarial training, this will make models more resistant to obfuscated and encrypted malware.

B. Using techniques like regularization and data augmentation, helps create a classifier that is resistant to noise and can face adversarial examples.

3.Improved and Expandable System:

A. To improve processing power, algorithms of dimensionality reduction, such as principal component analysis (PCA), can be applied.

B. Technologies like edge computing can be Implement, this will help to share the capacity of processing for real-time analysis on low-resources systems like IoT devices.

4.Comprehensive Dataset Evaluation: Expand training and validation datasets by adding real, diverse, and balanced samples from public sources and industry sources.

5.Combine local features with global features, in order to extract comprehensive features and the possibility of using the model to train on a large dataset.

The primary goal of this proposed framework is to achieve high recognition and classification accuracy, while addressing the limitations of present technologies. This framework can be considered an interesting candidate for research and application, given its ability to adapt to emerging malware threats.

3.2. Malware detection and classification algorithms based machine- crafted features

Deep Learning algorithms have attained significant advancement in the field of computer vision. Deep learning simulates the functions of the human cerebral cortex by implementing artificial neural networks with multiple hidden layers. Deep neural network layers provide multiple levels of abstraction by extracting multiple features [25]. Machine learning-based approaches are capable of utilizing the techniques of deep learning to learn complex patterns of malware and have shown significant findings in the identification of obfuscated malware [26].

Research on malware detection using deep learning is divided into two types: the first type uses convolutional neural networks for feature extraction as well as classification (end-to-end detection). While in the second type, features are extracted using convolutional neural networks and for classification, traditional machine learning classifiers such as (RF, KNN, etc.) are used (partial detection).

In [27], presented a scalable framework for malware detection called ScaleMalNet, that leverages from image processing techniques and deep learning. The framework consists of two-stage, at the first step classify malware utilizing a hybrid of static and dynamic analysis, next step classifying into corresponding malware families using image processing, using a hybrid architecture that combines convolutional neural networks (CNNs) and long short-term memory (LSTM) networks. Experimental results shows that deep learning methodologies considerably exceed classical algorithms of machine learning, achieving high accuracy in real-time malware detection.

In [28], introduce framework called the Malware Spectrogram Image Classification (MSIC). The framework is used to distinguish benign from malicious software, as well as to classify malware families, by using

spectrogram images with convolutional neural networks. The results show that the proposed framework achieves high accuracy in both detection and classification.

In [29], proposed a method for malware detection by texture visualizing of the code, based on a fast and improved R-CNN model, in addition to transfer learning. The method is based on texture features, extracted from grayscale images of the transformed code. The approach results in high accuracy, and a low false positive rate.

In [30], proposes (MalCVS), a novel framework for malware classification, that visualize section distribution information in PE files through using Colored Label boxes (CoLab). Feature extraction is done through a fine-tuned VGG16 model, and employing Support Vector Machine (SVM) for classification.

In [31], presents (RMVC) method for malware analysis. At the first visualizing malware as grayscale images, next utilizing the techniques of deep learning such as Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) to enhance classification accuracy. Incorporation of RNNs enables learning from unlabeled data, improving ability of the model to classify malware without the need for extensive labeled datasets.

In [32], produce a novel architecture, hybrid of two pre-trained network models in an optimized manner for classifying the malware. Malware binaries initially visualized as grayscale images, then classifying using Resnet and AlexNet architectures. The experimental results demonstrate this method can classify the malware effectively.

In [33], a new method proposed for malware detection through visualizing malware binaries in the frequency domain, using 'bigram-dct' images. This work employs Convolutional Neural Networks (CNNs) and enhance classification performance by takes advantage of transfer learning.

In [34], presents deep transfer learning approach for malware image classification (DTMIC), that extract features from malware images automatically by using convolutional neural networks (CNNs). The method addresses overfitting issues, also enhance classification performance by employing transfer learning and implementing the technique of early stopping. The proposed model present resilience against different malware types, inclusive packed and encrypted samples.

In [35], introduce ideal approach for devices with resource-constrained, which is a lightweight malware classification method based on image. By reducing image width (e.g., to 32x64, 16x64, or 8x64 pixels), Then use convolutional neural networks. Experiments show that the method achieves high accuracy, while significantly reducing training time.

A novel multi-task learning model was presented In [36], for malware detection, by converting malware files into image, classification accuracy has been greatly improved across different operating systems. The model leverages of Convolutional Neural Networks (CNNs), and show superior. Additionally, addressing the challenge of limited datasets by use of CycleGAN for data augmentation, and enhances the detection capabilities against obfuscation techniques though integration of dynamic analysis.

In [37], proposed framework for malware classification based on EfficientNetB1. Three image representation methods was evaluated in this study: fixed-width, file-size-based, and compressed byte stream-based. It turned out that the fixed-width method achieves the highest accuracy. The model suitable for real-world applications, as it benefits from transfer learning and compound scaling that achieve high performance using fewer parameters.

A novel architecture lightweight proposed in [38], for malware classification by integrating small Convolutional Neural Networks (CNN) and Variational Autoencoders (VAE), which enhanced by channel and spatial attention mechanisms. The method performs well even with limited data, and aims to improve classification performance while reducing computational costs.

In [39], presents an innovative system to analyze, detect, and categorize malware. It employing a combination of Deep Convolutional Neural Networks (CNN), with Deep Generative Adversarial Networks (GAN). The system generates new malware images to enhance training, by converting both malware and benign file binaries into RGB images.

In [40], presents VisMal, a novel visualized malware classification framework, which classify malware efficiently using (CNN), by converting them into grayscale images and enhancing their features leveraging a contrast-limited adaptive histogram equalization algorithm. The system improves analysis efficiency with a response time of 4.0 ms.

New Bi-model architecture for malware detection was proposed in [41], hybrid features are extracted from images of malware binaries using transfer learning with pre-trained models: VGG-16 and ResNet-50.

In [42], presents the Multi-Channel Code Visualization Method (MC-ISA) for malware detection. This method employing color enhancement, image size adaptation, and multi-channel representation, to address the limitations of existing visualization techniques. The proposed method use convolutional neural networks (CNNs),

and effectively retain byte information and original properties of spatial distribution for executable files, thus enhances feature representation aiding in the rapid detection of malware.

In [43], a novel method called MCTVD was proposed for malware classification, which utilizes visualization approach from three channel, based on assembly instructions and Markov transfer matrices. This method employed convolutional neural network with a simplified architecture.

In [44], presents a new visualization-based malware classification system utilizing transfer and ensemble learning (VMCTE). This system works effectively even when faced evasion techniques such as obfuscation and encryption. The proposed work extracts the features by combines three deep networks: (ResNet50, MobilenetV1, and MobilenetV2), and employs a support vector machine for classification, achieving high accuracy rates.

In [45], introduces MIGAN, that enable improved malware classification, through generate synthetic malware images and address class imbalance in datasets, by using novel GAN architecture. The framework achieves high accuracy by utilizing strategies of grayscale and color image conversion, and leverages traditional and transfer learning techniques.

In [46], introduces MalSort, a lightweight and efficient model for malware classification, that leverages a masked self-supervised learning framework using the Swin Transformer. The system enhances the efficiency of training and conserves hardware resources, though using malware color images to extracts multi-scale key features, while minimizing reliance on labeled data. The model handles the challenges of data imbalance and the scarcity of labeled samples.

Alshomrani et al. [47] introduced an explainable hybrid deep learning structure for visual malware classification, using malware binaries converted into grayscale images and fused with local feature extraction ability of the ConvNext-Tiny with global contextual modeling of the Swin Transformer. Grad-CAM was also utilized as a means of enhancing interpretability by visualizing the portions of each image that affected the classification decision. Multiple datasets were utilized to test the hybrid model with performance exceeding that of either an independent CNN or Transformer model. Hence, the study shows that the hybrid architecture of using local pattern recognition, through a CNN model, coupled with global attention, using a Transformer model can result in improved visual classification of malware.

Table 2. Deep Learning Feature Extraction Algorithms

Authors/ year	Spatial /Transform Domain	Methods	Dataset	Accuracy	advantages	limitations
Vinayakumar et al/ 2019 [27]	Spatial Domain	The proposed model utilizes a hybrid architecture combining CNN and LSTM networks	Ember, Malimg (DS1), and privately collected malware samples (DS2)	96.3% for DS1, 98.8% for DS2	1.Combines static and dynamic analysis with image processing techniques. 2.High Accuracy. 3.Real-Time Processing Capability.	1.Imbalanced Datasets. 2.Limited Dataset Variety. 4.Difficulty in tuning hyperparameters.
Azab & Khasawneh/ 2020 [28]	Transform Domain	Short-Time Fourier Transform (STFT), CNN	Special dataset (Malware + Benign)	92.8%, and 96% for binary classifying	1.Novel Framework. 2.High Accuracy. 3.Labeled Datasets: The research provides two labeled datasets.	1.The datasets used, may not be large or diverse enough. 2.The initial training of the CNN may still demand significant computational resources.
Zhao et al / 2020 [29]	Spatial Domain	Faster R-CNN,Transfer Learning, Data Augmentation, Stacked Autoencoder.	BIG 2015 1,821 malicious samples from six different families with augmentation	92.8%	1.Improved Detection Accuracy. 2.Robust Against Code Obfuscation. 3.Data Augmentation Techniques.	1. Transfer Learning Limitations. 2. Computational Complexity. 3.Limited Real-time Application. 4.Tested on small dataset.
Xiao et al/ 2021 [30]	Spatial Domain	VGG16, SVM, Colored Label boxes (CoLab)	VXV special Dataset, & BIG-2015 Dataset	96.59% and 98.94%	1.Novel Visualization Method. 2.Improved Classification Accuracy. 3.Effective Feature Extraction: reduce computational complexity while maintaining high performance.	1.May struggle with packed or obfuscated malware. 2.It is vulnerable to manipulation of partition distribution information. 3.Using limited datasets. 4.May not effectively detect zero-day malware.
Sun & Qian/ 2021 [31]	Spatial Domain	RNN, Minhash, CNN	2015 Kaggle Microsoft Malware Classification Challenge	99.55%	1.Integration of RNNs and CNNs. 2.High Classification Accuracy. 3.Ability to learn from unlabeled data.	1.Image Size Limitations. 2.Limited Dataset Diversity. 3.RNN Limitations: they can struggle with very long

					4.Innovative Visualization Technique.	sequences.
Aslan & Yilmaz / 2021 [32]	Spatial Domain	Resnet and AlexNet	Malimg, Microsoft BIG 2015, and Malevis, ImageNet	97.78% Malimg, 96.5% Malevis, 94.88% BIG 2015	1.Hybrid Model: enhancing the ability to extract distinctive features. 2.High Accuracy. 3.Feature Space Reduction.	1.Feature Similarity Leading to Misclassification. 2. Require significant computational power and resources.
Mohammed et al / 2021 [33]	Spatial domain and transform domain	CNN, ResNet, Discrete Cosine Transform (DCT)	MaleX: 1,044,394, 864,669 malware and 179,725 benign.	96%	1.Novel Visualization Method. 2.Creation of a Large Dataset: MaleX. 3.High Classification Accuracy: 96%.	1.Less scalable due to the computationally expensive feature matching. 2.High Time Complexity.
Kumar & Janet/ 2022 [34]	Spatial domain	VGG16, transfer Learning, t-Distributed Stochastic Neighbor Embedding (t-SNE), early stopping technique	MalImg and Microsoft BIG	98.92% for Malimg datasets and 93.19% for Microsoft	1.Novel Methodology. 2.High Accuracy. 3.Early Stopping Technique to prevent overfitting. 4.Resilience to Packed and Encrypted Malware.	1.Requires massive computational resources and time. 2.Might lead to premature stopping or prolonged training.
Son et al/ 2022 [35]	Spatial domain	CNN, Resampling Techniques (Bilinear)	Malimg and Malheur	97%, on Malimg, 91%, on Malheur, using CNN	1.Simplified Feature Extraction. 2.High Classification Accuracy. 3.Reduces the computational complexity and time.	1.Limited Dataset Diversity. 2.No Real-Time Testing.
Bensaoud & Kalita/ 2022 [36]	Spatial domain	CNNs, CycleGAN, Multi-Task Learning (MTL)	Six special datasets & Malimg	99.97% family classifying, average 99.91% for binary classifying	1.Novel Multi-Task Learning Framework. 2.High Accuracy. 3.Create a Comprehensive Dataset. 4.Robustness to Obfuscation.	1.Computational Complexity. 2.Does not extensively discuss other important metrics such as (precision and recall). 3.Lack of Real-World Testing.
Chaganti et al/ 2022 [37]	Spatial domain	EfficientNetB1, File Size-Based Image Width Selection, Compressed Byte Stream Length.	Microsoft Malware Classification Challenge (BIG 2015)	99%	1.High Accuracy. 2.Computational Efficiency. 3.Transfer Learning. 4.Innovative Image Representation Techniques.	1.Limited Generalization to New Malware. 2.Obfuscated Malware Challenges. 3.Hardware and Resource Constraints. 4.Dataset Limitations.
Dao et al/ 2022 [38]	Spatial domain and transform domain	CNN, VAE, Attention Mechanism, AVAE, RF	unbalanced and balanced Malimg	99.40%, 98.40%	1. Lightweight Model: making it resource-efficient. 2. High Accuracy. 3. Efficiency with Limited Data. 4. Real-time Classification.	1.It is affected by the use of evasion techniques. 2.Limited Dataset. 3.Feature selection may lead to suboptimal performance.
Falana et al/ 2022 [39]	Spatial domain	Deep Convolutional Neural Networks (DCNN), Deep Generative Adversarial Networks (DGAN)	Special dataset (Malware + Benign), Malimg, MaleVis	96.7%, 99.8%, 96.7% & (96.5% for binary classification)	1.High Accuracy. 2.Innovative Visualization Approach. 3.Robustness Against Adversarial Attacks. 4.Reduced Training Sample Requirement.	1.The search uses part of the Malimg dataset, but not all of it. 2.The implementation of deep learning models requires significant computational resources.
Zhong et al/ 2023 [40]	Spatial domain	Contrast-Limited Adaptive Histogram Equalization (CLAHE), Feature Engineering, CNN	Malimg	96%	1.Innovative Framework (VisMal). 2.Enhanced Classification Accuracy.3.Fast response.	1.Only tested on the Malimg dataset. 2.Difficulty to distinguish highly similar malware families. 3.Not evaluated in real-world scenarios.
Rustam et al/ 2023 [41]	Spatial domain	VGG-16, ResNet-50, SVC, Random Forest (RF), LR, KNN	Malimg	100%	1.Innovative Bi-model architecture. 2.High accuracy. 3.Effective use of transfer learning and hybrid features.	1.High computational cost. 2.Limited dataset diversity. 3.Increased complexity of the Bi-model architecture. 4.Potential Overfitting.
Qi et al/ 2023 [42]	Spatial domain	Image Size Adaptive Mechanism (ISA), MC-ISA, First-Order Markov State Transition Matrix,	Special dataset: 8870 malware files and 8590 benign files, totaling 17,460	99.36% Inception-ResNet-V2, for malware	1.Innovative Multi-Channel Approach. 2.Image size adaptation, color enhancement, and multi-channel	1.The data used not publicly available. 2.Complexity of Implementation compared to simpler models.

		ResNet50, VGG16	samples	detection	generation. 3.High Detection Accuracy.	3.Generalization Challenges.
Deng et al/ 2023 [43]	Spatial domain	IDA Pro, Markov Transfer Matrices, CNN	Microsoft 2015	99.44%	1.Innovative Visualization Method. 2.High Classification Accuracy. 3.Adaptability to Limited Data.	1.Limited Generalization. 3.The scalability not thoroughly addressed, which could be a concern for practical applications.
Chen & Cao/ 2023 [44]	Spatial domain	ResNet50, MobilenetV1, MobilenetV2, Transfer Learning, PCA, SVM.	Malimg	99.64%	1.High Accuracy. 2.Robustness. 3.Leverages pre-trained models. 4.Novel Approach.	1.Evaluated only on the (Malimg dataset). 2.Can be computationally expensive.
Sharma et al/ 2024 [45]	Spatial domain	Generative Adversarial Networks (GANs), CNN, Resnet50v2	Special dataset 50,000, and Malimg	99.4% & 99.5%	1.Novel GAN Variant. 2.High Accuracy and robustness to Zero-Day Malware. 3.Public Dataset Availability.	1.Requires significant resources. 2. Limited evaluation for images. 3.Model Dependency on Synthetic Data
Wang et al/ 2024 [46]	Spatial domain	Masked Self-Supervised Learning, Swin Transformer, Asymmetric Encoder and Decoder, Lightweight Prediction Head.	BIG 2015, Malimg, 1087 benign samples from the Playdronse	97.85% for Microsoft BIG 2015, and 98.28% for Malimg	1.Lightweight efficient model. 2.Eliminating the need for large-scale labeled datasets. 3.Reduces computational complexity. 4.Handling Imbalanced datasets. 5.High accuracy.	1.Self-Supervised Learning may still suffer from generalization errors. 2.The model's ability to generalize to unseen or zero-day malware is not thoroughly explored. 3.Complexity of Model Tuning.
Alshomrani et al/ 2025 [47]	Spatial domain	ConvNeXt-Tiny, Swin Transformer, Grad-CAM	Malimg, MaleVis, VirusMNIST, Maldeb, Dumpware-10	99.57% db1, 88.65% db2, 93.90% db3, 98% db4, 97% db5	1.Captures local spatial and global contextual features. 2.Uses Grad-CAM to improve model interpretability. 3.Applies optimization techniques such as adaptive fusion.	1.Require a considerable amount of computing resources. 2.Difficulty in generalizing the results. 3.Not designed for use in resource limited environments or real-time.

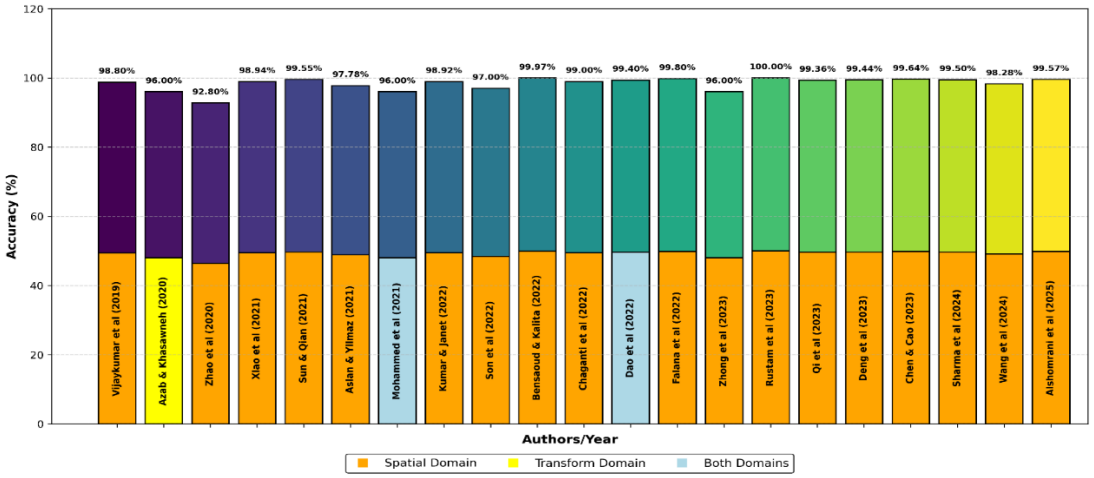


Figure 6. Comparison of accuracy between works utilizing deep learning features

3.2.1. Comparison of malware detection and classification with deep learning features

The use of binary code-to-image transforming technology to detect and classify malware, has made great strides. These methods benefit from image processing and machine learning techniques to convert the binary code of malware into grayscale or RGB images to extract important features. Features extracted using advanced deep learning architectures have proven effective, achieving accuracy rates of over 99%. If we look at the works that achieved the highest accuracy results, they are as follows:

[39], with classification rate of 99.80%, this research presented an innovative approach with minimal training sample requirements and achieved high accuracy and robustness against adversarial attacks. However, this high accuracy was achieved on the Malimg dataset, and the researcher used only a portion of this dataset, but not all

of it. The results were not as good on other datasets, so it is unfair to compare it to other work that uses the entire dataset. In addition, the model suffers from significant computational resource consumption.

[36], with classification rate of 99.97%, as noted, this work achieved high classification accuracy using an innovative framework, and the researcher also created a dataset and made it publicly available. However, the working model lacks real-world testing and suffers greatly from computational complexity. Furthermore, the system does not discuss other important metrics that must be met besides accuracy, such as precision and recall, which raises questions about the feasibility of the presented results.

[41], with classification rate of 100%, an innovative model was built and achieved high classification accuracy, but it suffers from a lack of real-world testing and high computational costs. Furthermore, since it relied on a single dataset, its results are difficult to generalize. In addition, achieving 100% accuracy raises concerns about overfitting, especially given the unbalanced dataset.

Despite the achievements of research that adopted deep learning to extract features, there are still critical challenges that have not been addressed, such as: imbalance the dataset, high expensive of the computation, dependency on specific datasets, poor ability to generalize, and the challenges of dealing with obfuscated or zero-day malware. The majority of current techniques focuses largely on the spatial domain, while efforts and work in the frequency domain are minimal, which may shed light on more effective ways to identify and classifying the malware. In addition, numerous researches place emphasis on the nonreal-time classification, which raises concerns about its feasibility in real-world cybersecurity systems.

To address the identified limitations, an approach could be proposed for future work, combining the strengths of existing techniques to overcome their weaknesses:

1.Hybrid Domain Analysis

A. Integrate spatial domain analysis (e.g. grayscale images), with the techniques of frequency domain such as discrete cosine transform (DCT) or wavelet transform.

B. This approach is more effective against obfuscation techniques because it incorporates the detection of spatial patterns and frequency based features.

2. Dynamic and Adaptive Visualization:

A. Utilize adaptive visualization techniques, like contrast enhancement or dynamic resizing to maintain critical byte-level data.

B. Investigate mapping of the color-channel, for representing multiple features (opcode, frequency, and entropy) in RGB images.

3. Advanced Feature Fusion:

A. Implement multi-channel models of deep learning to fuse features from handcrafted techniques like (SFTA) and deep learning models such as (ResNet and EfficientNet).

B. Use attention mechanisms or transformers to prioritize dynamically the most informative features

4. Improved Dataset Utilization:

A. Expand scope of the dataset through using diverse sources, and employ CycleGANs or GANs for data augmentation.

B. Validate the designed systems against unseen datasets to prove its effectiveness against zero-day malware.

5. Lightweight and Real-Time Deployment:

A. Construct lightweight models by taking advantage of efficient architectures such as MobileNet, and Swin Transformer to facilitate the deployment in the systems of real time.

B. Merge edge computing to allow classification to be performed on low-power devices.

6. Resize images by reducing the width of the image to 32, 16 or 8 pixels and slightly reducing the height. Because the texture of the malware image is similar in width and different in height. This will reduce processing time and make the model applicable in the real world.

This approach ensures robustness, scalability, and practicality, addressing gaps in recent research, while enhancing the capabilities of malware detection of cybersecurity systems in real-world.

3.3. Malware detection and classification algorithms based hand-crafted features & machine- crafted features

In the previous two sections, we observed that many researchers depend on extracting hand-crafted features using traditional machine learning techniques to identify and classify malware, while others have gone to use deep learning techniques for the same purpose. Each of the two methods has its advantages and disadvantages that may affect the recognition process. While some researchers prefer to combine the two types in order to obtain a robust

hybrid recognition system that combines the advantages of the previous two types. Through research in this area, we noticed that research papers that use hybrid systems are not widely used, as we can see in Table 3.

A novel approach was presented in [48], for classifying malware, it is addressing the challenges of class imbalance and limited training data. The system takes advantage of innovative feature extraction techniques, including 3D visualization and N-gram analysis, producing a hybrid system that combines non-parametric and deep learning classifiers to achieve high accuracy.

In [49], present the Hybrid Image Transformation (HIT) method for malware classification, where executable files converts into color images using various encoding techniques, and utilize XGBoost and CNN for feature extraction. The proposed method enhances the performance of the classifiers, and achieves a significant improvement in accuracy, but it takes more time to generate the images.

Hybrid malware classification method was proposed In [50], that combines segmentation-based fractal texture analysis (SFTA) with features extracted from pre-trained models (AlexNet and Inception-V3). At the first convert malware binaries into grayscale images, then applying data augmentation to address class imbalance, finally employing cubic SVM for the classification.

In [51], innovative two visualization methods to analyze the malware based on n-gram features of byte sequences: 1. Space Filling Curve Mapping (SFCM) method 2. Markov Dot Plot (MDP) method. VGG16 and VGG19 networks utilized for feature extraction and SVM for classification, that leading to high accuracy in both classification and detection.

In [52], a new approach was presented to detect and classify the malware using visualization techniques and deep learning. The executable files will be convert into various image formats, including grayscale and RGB, then extract features by employing Gabor filters, the study achieves significant improvements in detection accuracy.

A novel approach was presented in [53], for malware classification, where a few-shot learning framework utilized, that utilizes a new technique for visualization called the GEM image, which fuses three categories: gray-level matrix images, entropy graph, and Markov images. Classification was performed using Convolutional Siamese Neural Network (CSNN), and Shallow-CNN, and demonstrated high accuracy across multiple datasets. The study achieved considerable performance improvements compared to existing methods.

In [54], a new approach was presented for malware classification using Markov images, obtained by converting malware binaries, Which is characterized by maintaining comprehensive statistics of malware bytes. To extract texture, the study used Gabor filters, the study leverages both a custom deep CNN and a fine-tuned Xception CNN, high accuracy rates were achieved across multiple datasets.

Deep learning-based approach was proposed in [55], With the aim of detecting and classifying malware, the approach relied on visualizing malware binaries as grayscale, RGB, and Markov images. Gabor filters were employed to extract textures from these images, and models were developed using VGG-3 and DenseNet. The presented approach achieves high accuracy rates, with response times ranging from 0.004 to 0.006 milliseconds.

Malware detection new approach was introduced in [56], employs Fast and Adaptive Bidirectional Empirical Mode Decomposition (FABEMD), blended with 3D deep learning models, precisely 3D-Resnet-18 and 3D-VGG-16. The proposed methodology addressed challenges such as insufficient training data and imbalanced datasets, leading to improved malware image classification.

An image-based approach developed by Yaseen and others [57] is a form of digital forensics used to detect fileless and memory-sitting malware using memory dump files as visual images. This approach uses various techniques to analyze images based on local and global texture descriptors: HOG, SIFT, ORB, LBP, DWT, GIST, and GLCM. Multiple image sizes and different machine learning classifiers were examined for the purpose of providing an accurate output and maintaining a computationally efficient solution. Overall, the results showed that HOG and DWT performed very well, and that the LIME-guided MobileNet method identified important areas of the image and significantly decreased processing time while at the same time producing a classification accuracy of 85%.

Table 3. Hybrid Feature Extraction Algorithms

Authors/ year	Spatial/ Transform Domain	Methods	Dataset	Accuracy	advantages	limitations
Classification et al/ 2018 [48]	Spatial Domain	PCA, kNN, deep learning auto-encoders	Microsoft BIG 2015	Overall Accuracy (Hybrid System) 99.07%, accuracy for 'Simda' (kNN)	1.Novel Hybrid Approach. 2.Improved Classification of Rare Classes. 3.Maintained High Overall Accuracy.	1.Limited Generalizability. 2.Performance may vary with different dataset characteristics. 3.Complexity of Feature Extraction. 4.Not tested on classes with extremely few samples. 5.Computational

				81.0%	4.Handling Imbalance Class.	Resource Requirements.
Nguyen & Nguyen/ 2019 [49]	Spatial Domain	Hilbert curve, XGBoost, CNN, GLCM, Haralick texture features	16,000 PE samples (8,000 malware and 8,000 benign)	93.01%	1.Novel Image Transformation. 2.Improved Classification Accuracy. 3.Robustness to Obfuscation. 4.Speeding. 5.Publicly Available Dataset.	1. more computationally expensive than simpler methods. 2.Limited Evaluation on Real-World Scenarios. 3.Haralick texture features, may not capture all the relevant information in the images.
Nisa et al/ 2020 [50]	Spatial Domain	SFTA, AlexNet, Inception-v3, PCA, SVM, KNN, DT, and Linear Discriminant (LD)	Malimg, augmented to 25,000 images.	99.3%	1.Hybrid Feature Fusion. 2.High Accuracy. 3.Addressing Dataset Imbalance.	1.Computational Cost of Deep Learning Features. 2.Feature Fusion Complexity. 3.Lack of Real-World Testing. 4.Limited Exploration of Feature Selection.
Ren et al/ 2020 [51]	Spatial Domain	SFCM, MDP, VGG16, VGG19, SVM	Microsoft BIG 2015, (9,390) benign PE files, and Kaspersky Malware Samples	99.08% Classification, 99.21% Detection on BIG 2015, and 98.24%, 99.02% on Kaspersky.	1.Novel Visualization Methods. 2.Improved Classification and Detection Accuracy. 3.Resilience to Obfuscation.	1.Computational Complexity. 2.Limited Generalization: The study primarily focuses on specific malware families. 3.Dataset Limitations. 4.Evaluation metrics do not provide a complete picture of performance.
Pinhero et/ 2021 [52]	Spatial and Transform Domain	Gabor filters, VGG3, CNNs, ResNet-50	DB1:BIG 2015 +12,971 Benign samples, DB2: Malimg + 12,971 benign samples	detection: 99.20% for DB1, 99.97% for DB2, family classification: 98.25% for Big 2015, 98.46% for Malimg	1.Novel Image Generation Techniques. 2.Improving the differentiation between malware and benign samples.	1.High Computational Cost. 2.May struggle with detecting new, evolving threats. 3.The models may require frequent retraining to adapt to new malware variants. 4.Real-Time Detection Limitations.
Conti et al/ 2022 [53]	Spatial Domain	GLCM, CSNN, Shallow Few-Shot (Shallow-FS) model, Shallow-CNN	Malimg Dataset (A), Microsoft BIG 2015 Dataset (B), MalwareBazaar Dataset (C).	99.93% on the Malimg, 98.56% on the Microsoft BIG 2015, and 98.51% on the MalwareBazaar	1.A novel visualization technique. 2.High accuracy using with limited data. 3.Effective against obfuscated malware. 4.Computationally efficient.	1.Designing and fine-tuning hand-crafted features requires significant effort and expertise. 2.Few-shot learning may not scale well to large numbers of classes or highly diverse datasets. 3.Generating GEM images for large datasets can be computationally expensive.
Sharma et al/ 2023 [54]	Spatial Domain	Gabor Filter, Custom CNN and the fine-tuned Xception CNN	BIG 2015, Malimg, Custom-Built Windows Malware, Custom-Built IoT Malware Dataset	99.12% for BIG 2015, 99.22% for Malimg, 99.20% for Windows Dataset, 99.18% for IoT Dataset	1.High Classification Accuracy. 2.Suitable for resource-constrained environments. 3.Development of custom datasets.	1.The models show some vulnerability to adversarial attacks. 2.The custom datasets are relatively small; this could limit the generalizability of the results. 3.Lack of Real-World Deployment. 4.Focus on Specific Malware Types.
Anandhi et al/ 2024 [55]	Spatial and Transform Domain	Densely Connected Network (DenseNet), Gabor filter	Malimg and BIG2015	99.94% for Malimg 98.98%, for BIG2015, and 12,971 benign samples add to both datasets.	1.High Accuracy. 2.Innovative Feature Extraction. 3.Real-time Analysis Capability.	1.The system is complex and requires significant computing resources. 2.Some families, show lower classification accuracy (42.5%). 4.Aacknowledges the potential for adversarial attacks
Al-Khater & Al-Madeed/ 2024 [56]	Spatial and Transform Domain	FABEMD, VGG16, and ResNet-18	Malimg and MaleVis	99.64% on the Malimg and 99.46% on the MaleVis	1.Novel methodology. 2.Handling of imbalanced datasets. 3.High accuracy.	1.High computational complexity. 2.Need for Large Datasets. 3.concerns for real-time applications. 4.Reliance on Data Augmentation may affect classification accuracy.
Yaseen et	Spatial	HOG, LBP,	Dumpware10	85.02%	1.Solve the	1.Final Accuracy was lower than

al/ 2025 [57]	Domain	SIFT, ORB, GIST, GLCM, LIME – Local Interpretable Model-Agnostic Explanations, CNN			problem of detecting fileless and malware which reside in memory. 2.Extract features based on region of interest. 3.Much less processing time.	currently existing methods. 2.limited testing using Dumpware10 dataset. 3.Computing Time for LIME still considered an expensive problem.

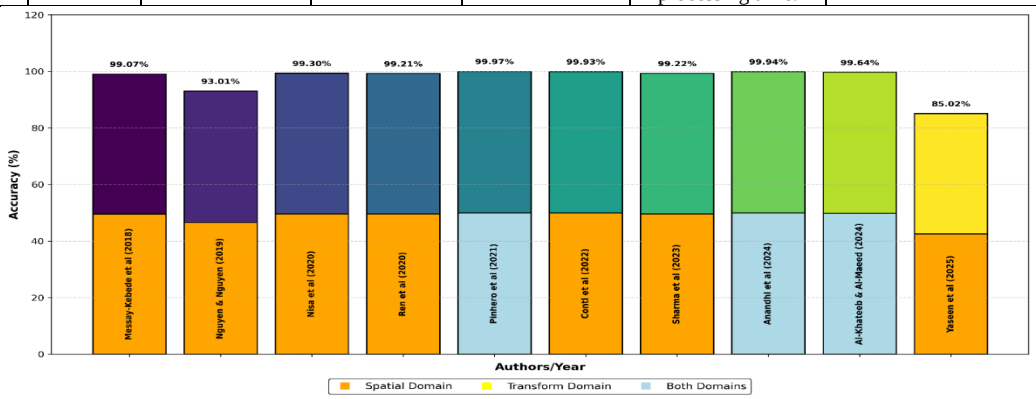


Figure 7. Comparison of accuracy between works utilizing hybrid features

3.3.1. Comparison of malware detection using hybrid features

Looking at the works that achieved the highest accuracy results, they are as follows:

[53], with classification rate of 99.93%, the work was tested on three datasets, with the best performance on the Malimg dataset. The system introduced a new technique for malware visualization, resulting in high classification accuracy, especially with limited data. The system is also computationally efficient. However, it can be computationally expensive when generating images for large datasets, raising questions about its generalizability. Furthermore, limited-band learning may not be suitable for highly diverse datasets. Manual feature extraction also requires significant expertise and effort.

[55], with classification rate of 99.94%, this research presented an innovative feature extraction model, resulting in highly accurate results and rapid execution. Despite these advantages, relying on advanced learning models such as DenseNet increases the complexity of the system. Consequently, the model requires significant computational resources. The system also failed to classify some malware families, such as Swizzor.genII. Therefore, the researcher combined it with the C2LOP.P family because it is similar to it to achieve high results. Therefore, it is unfair to compare it with work that identifies all families.

[52], with detection rate of 99.97%, the research introduced new image generation techniques and achieved high accuracy rates. However, you may notice that this high score is for detecting malware, not for classifying its families. The highest accuracy for classifying families is 98.46%, so it is unfair to compare it with higher results from research that classifies malware into families. Add to this the high computational cost. Furthermore, the model may require frequent retraining to adapt to new malware, which presents a problem that needs to be addressed.

Recent studies have significantly improved the accuracy of malware classification, focusing on advanced image processing techniques such as Gabor filtering and segmentation-based fractal texture analysis, as well as the use of deep learning models such as CNN, VGG, ResNet, and DenseNet. These measures have significantly contributed to solving problems such as obfuscation, packing, or class imbalance. Some of these methods report accuracy rates over 99%.

Despite the achievements of research that adopted hybrid features using hand crafted and deep learning to extract features, a number of challenges still remain that have not been addressed, such as: depending heavily on specific data sets by many studies, which may limit the generalizability of the results to detect other malware families or apply its systems in real-world scenarios. Another important issue is computational complexity, which remains a challenge in feature extraction and deep learning models, particularly in real-time detection systems. The third challenge relates to the issue of relying on hand-crafted features, as this approach may not capture all relevant information. The fourth challenge is the lack of real-world deployment insights, which raises questions about the practical applicability of these approaches. Furthermore, there are other long-standing challenges, such as dealing with adversarial attacks and the need for continuous retraining to adapt to evolving malware variants.

To address the above limitations, we propose the following improvements:

1. Hybrid Feature Extraction: develop a hybrid method for feature extraction that combines hand-crafted features (e.g., Gabor filters, SFTA) with features extracted automatically using deep learning. This approach will leverage the strengths of both methods, thus ensuring that local and global patterns are effectively captured.
2.Real-Time Detection Framework: design an efficient and effective framework for real-time malware detection, using precise techniques and low processing power. This could involve optimizing architectures of the deep learning models (e.g., using MobileNet or EfficientNet) and ereducing inference time by employing parallel processing techniques.
3.Adversarial Robustness: utilizing adversarial training to increase the robustness of malware detection models against adversary threats and attacks. This may involve generating adversarial examples during training and applying various adversarial defense mechanisms.
4.Generalization Cross-Dataset: analyze detection systems and enhance their effectiveness to be more generalizable, by applying them to diverse datasets as well as real-life samples.
5.Few-Shot Learning for Unknown Malware: improve the classification of unknown or rare malware families, by exploitation the few-shot learning.
6.Hybrid Image Transformation: introduce framework of hybrid transformation, which combine the advantage of both the spatial domain and transform domain techniques, in order to extract comprehensive features.
7.Incorporating interpretation techniques such as SHAP or Grad-CAM, with the aim of improving the interpretability of malware detection models, allows the end user of the system to gain a deeper understanding of the reasons for classification.
8.Continuous Learning and Adaptation: implementing continuous learning mechanisms, enables systems to quickly adapt to any new threats, that emerge over time without the need to completely retrain the model. These mechanisms may include online learning algorithms or incremental learning techniques, which update the model as new data becomes available.

By addressing these domains, future research can build on the strengths of existing systems while mitigating their limitations, leading to more robust, efficient, and real-world malware detection and classification systems.

Tables 1, 2, and 3 provide a comparison between the methods used to extract features manually, using deep learning, or using a hybrid between the two methods. The comparison was made based on several factors, including determining the domain used, whether it is the spatial domain or the frequency domain. As well as determining the methods and tools used to extract features, process images, or classify them, to determine the strength of the work in terms of accuracy, system response speed, and scalability. In addition, the focus was on the datasets used in these studies, to determine whether there is bias in the results and to know the balanced and unbalanced datasets. All this information determines whether the system is applicable in reality and determines its scalability, and evaluates whether the speed at which it operates is compatible with cybersecurity, which requires accuracy and speed in response.

4. CONCLUSION

Software Binaries analysis has long been used and continues to be used to detect and classify malware. Although these systems are powerful, it requires significant expertise and time to address new generations and mutations of malware. To address many of the above problems, software analysis based on analyzing patterns generated from malware images emerged. Many researchers have presented their work in this field, and detection is performed either through hand crafted feature extraction, deep learning, or a hybrid approach. Each of these three approaches has its pros and cons. Perhaps the most significant cons that should be highlighted is the need for an expert when hand crafted features, while the problem with deep learning is its high resource consumption. In addition, several other issues have been discussed in detail, and some basic solutions have been proposed that can be relied upon to build an efficient detection system. In the future, we will implement the suggestions presented in each section to develop a malware detection framework and contribute to a safer cyberspace.

REFERENCES

[1] D. Gibert, C. Mateu, J. Planes, and R. Vicens, “Using convolutional neural networks for classification of malware represented as images,” *J. Comput. Virol. Hacking Tech.*, vol. 15, no. 1, pp. 15–28, 2019, doi: 10.1007/s11416-018-0323-0.

[2] S. Ni, Q. Qian, and R. Zhang, “Malware identification using visualization images and deep learning,” *Comput. Secur.*, vol. 77, pp. 871–885, 2018, doi: 10.1016/j.cose.2018.04.005.

[3] J. Jeon, J. H. Park, and Y. S. Jeong, "Dynamic Analysis for IoT Malware Detection with Convolution Neural Network Model," *IEEE Access*, vol. 8, pp. 96899–96911, 2020, doi: 10.1109/ACCESS.2020.2995887.

[4] H. Pancharіya and S. Bharkad, "Comparative Analysis of Feature Extraction Methods for Optic Disc Detection," *IOSR J. Comput. Eng.*, vol. 16, no. 3, pp. 49–54, 2014, doi: 10.9790/0661-16334954.

[5] J. Lee and J. Lee, "A Classification System for Visualized Malware Based on Multiple Autoencoder Models," *IEEE Access*, vol. 9, pp. 144786–144795, 2021, doi: 10.1109/ACCESS.2021.3122083.

[6] L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, "Malware images: Visualization and automatic classification," in *ACM International Conference Proceeding Series*, 2011, pp. 1–7. doi: https://doi.org/10.1145/2016904.20169.

[7] A. S. Bozkir, A. O. Cankaya, and M. Aydos, "Utilization and comparision of convolutional neural networks in malware recognition [Kötücül Yazilimlarin Taninmasinda Evrişimsel Sinir Ağlarinin Kullanimi ve Karşilaştirilmasi]," in *27th Signal Processing and Communications Applications Conference, SIU 2019*, IEEE, 2019, pp. 1–4. doi: 10.1109/SIU.2019.8806511.

[8] Microsoft, "Microsoft Malware Classification Challenge (BIG 2015)," Kaggle. [Online]. Available: https://www.kaggle.com/c/malware-classification

[9] M. Ahmadi, D. Ulyanov, S. Semenov, M. Trofimov, and G. Giacinto, "Microsoft malware classification challenge," *arXiv Prepr. arXiv1802.10135*, pp. 183–194, 2018, doi: https://doi.org/10.48550/arXiv.1802.10135.

[10] A. Makandar and A. Patrot, "Malware class recognition using image processing techniques," in *2017 International Conference on Data Management, Analytics and Innovation, ICDMAI 2017*, IEEE, 2017, pp. 76–80. doi: 10.1109/ICDMAI.2017.8073489.

[11] A. Makandar and A. Patrot, "Wavelet Statistical Feature based Malware Class Recognition and Classification using Supervised Learning Classifier," *Orient. J. Comput. Sci. Technol.*, vol. 10, no. 2, pp. 400–406, 2017, doi: 10.13005/ojcst/10.02.20.

[12] A. Makandar and A. Patrot, "Trojan malware image pattern classification," in *In Proceedings of International Conference on Cognition and Recognition: ICCR*, 2018, pp. 253–262. doi: 10.1007/978-981-10-5146-3_24.

[13] J. Fu, J. Xue, Y. Wang, Z. Liu, and C. Shan, "Malware Visualization for Fine-Grained Classification," *IEEE Access*, vol. 6, pp. 14510–14523, 2018, doi: 10.1109/ACCESS.2018.2805301.

[14] H. Naeem, B. Guo, and M. R. Naeem, "A light-weight malware static visual analysis for IoT infrastructure," in *2018 International Conference on Artificial Intelligence and Big Data, ICAIBD 2018*, IEEE, 2018, pp. 240–244. doi: 10.1109/ICAIBD.2018.8396202.

[15] Y. S. Liu, Y. K. Lai, Z. H. Wang, and H. B. Yan, "A New Learning Approach to Malware Classification Using Discriminative Feature Extraction," *IEEE Access*, vol. 7, pp. 13015–13023, 2019, doi: 10.1109/ACCESS.2019.2892500.

[16] H. Naeem, B. Guo, M. R. Naeem, F. Ullah, H. Aldabbas, and M. S. Javed, "Identification of malicious code variants based on image visualization," *Comput. Electr. Eng.*, vol. 76, pp. 225–237, 2019, doi: 10.1016/j.compeleceng.2019.03.015.

[17] I. Baptista, S. Shiaeles, and N. Kolokotronis, "A novel malware detection system based on machine learning and binary visualization," in *2019 IEEE International Conference on Communications Workshops, ICC Workshops 2019 - Proceedings*, IEEE, 2019, pp. 1–6. doi: 10.1109/ICCW.2019.8757060.

[18] V. Verma, S. K. Muttoo, and V. B. Singh, "Multiclass malware classification via first- and second-order texture statistics," *Comput. Secur.*, vol. 97, p. 101895, 2020, doi: 10.1016/j.cose.2020.101895.

[19] V. Moussas and A. Andreatos, "Malware detection based on code visualization and two-level classification," *Inf.*, vol. 12, no. 3, pp. 1–14, 2021, doi: 10.3390/info12030118.

[20] T. Gao, L. Zhao, X. Li, and W. Chen, "Malware detection based on semi-supervised learning with malware visualization," *Math. Biosci. Eng.*, vol. 18, no. 5, pp. 5955–6011, 2021, doi: 10.3934/MBE.2021300.

[21] I. T. Ahmed, N. Jamil, M. M. Din, and B. T. Hammad, "Binary and Multi-Class Malware Threads Classification," *Appl. Sci.*, vol. 12, no. 24, 2022, doi: 10.3390/app122412528.

[22] S. O'Shaughnessy and S. Sheridan, "Image-based malware classification hybrid framework based on space-filling curves," *Comput. Secur.*, vol. 116, p. 102660, 2022, doi: 10.1016/j.cose.2022.102660.

[23] M. Benchadi Djafer Yahia, B. Batalo, and K. Fukui, "Efficient Malware Analysis Using Subspace-Based Methods on Representative Image Patterns," *IEEE Access*, vol. 11, pp. 102492–102507, 2023, doi: 10.1109/ACCESS.2023.3313409.

[24] I. T. Ahmed, B. T. Hammad, and N. Jamil, "A Comparative Performance Analysis of Malware Detection Algorithms Based on Various Texture Features and Classifiers," *IEEE Access*, vol. 12, no. January, pp. 11500–11519, 2024, doi: 10.1109/ACCESS.2024.3354959.

[25] R. Chauhan, K. K. Ghanshala, and R. C. Joshi, "Convolutional Neural Network (CNN) for Image Detection and Recognition," in *ICSCCC 2018 - 1st International Conference on Secure Cyber Computing and Communications*, IEEE, 2018, pp. 278–282. doi: 10.1109/ICSCCC.2018.8703316.

[26] A. Darem, J. Abawajy, A. Makkar, A. Alhashmi, and S. Alanazi, "Visualization and deep-learning-based malware variant detection

using OpCode-level features," *Futur. Gener. Comput. Syst.*, vol. 125, pp. 314–323, 2021, doi: 10.1016/j.future.2021.06.032.

[27] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, and S. Venkatraman, "Robust Intelligent Malware Detection Using Deep Learning," *IEEE Access*, vol. 7, pp. 46717–46738, 2019, doi: 10.1109/ACCESS.2019.2906934.

[28] A. Azab and M. Khasawneh, "MSIC: Malware Spectrogram Image Classification," *IEEE Access*, vol. 8, pp. 102007–102021, 2020, doi: 10.1109/ACCESS.2020.2999320.

[29] Y. Zhao, W. Cui, S. Geng, B. Bo, Y. Feng, and W. Zhang, "A malware detection method of code texture visualization based on an improved faster RCNN combining transfer learning," *IEEE Access*, vol. 8, pp. 166630–166641, 2020, doi: 10.1109/ACCESS.2020.3022722.

[30] M. Xiao, C. Guo, G. Shen, Y. Cui, and C. Jiang, "Image-based malware classification using section distribution information," *Comput. Secur.*, vol. 110, p. 102420, 2021, doi: 10.1016/j.cose.2021.102420.

[31] G. Sun and Q. Qian, "Deep Learning and Visualization for Identifying Malware Families," *IEEE Trans. Dependable Secur. Comput.*, vol. 18, no. 1, pp. 283–295, 2021, doi: 10.1109/TDSC.2018.2884928.

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

[33] T. M. Mohammed, L. Nataraj, S. Chikkagoudar, S. Chandrasekaran, and B. S. Manjunath, "Malware detection using frequency domain-based image visualization and deep learning," *arXiv Prepr. arXiv2101.10578*, 2021, doi: 10.24251/hicss.2021.858.

[34] S. Kumar and B. Janet, "DTMIC: Deep transfer learning for malware image classification," *J. Inf. Secur. Appl.*, vol. 64, no. December 2021, p. 103063, 2022, doi: 10.1016/j.jisa.2021.103063.

[35] T. T. Son, C. Lee, H. Le-Minh, N. Aslam, and V. C. Dat, "An enhancement for image-based malware classification using machine learning with low dimension normalized input images," *J. Inf. Secur. Appl.*, vol. 69, no. August, p. 103308, 2022, doi: 10.1016/j.jisa.2022.103308.

[36] A. Bensaoud and J. Kalita, "Deep multi-task learning for malware image classification," *J. Inf. Secur. Appl.*, vol. 64, 2022, doi: 10.1016/j.jisa.2021.103057.

[37] R. Chaganti, V. Ravi, and T. D. Pham, "Image-based malware representation approach with EfficientNet convolutional neural networks for effective malware classification," *J. Inf. Secur. Appl.*, vol. 69, no. August, p. 103306, 2022, doi: 10.1016/j.jisa.2022.103306.

[38] T. Van Dao, H. Sato, and M. Kubo, "An Attention Mechanism for Combination of CNN and VAE for Image-Based Malware Classification," *IEEE Access*, vol. 10, no. June, pp. 85127–85136, 2022, doi: 10.1109/ACCESS.2022.3198072.

[39] O. J. Falana, A. S. Sodiya, S. A. Onashoga, and B. S. Badmus, "Mal-Detect: An intelligent visualization approach for malware detection," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 5, pp. 1968–1983, 2022, doi: 10.1016/j.jksuci.2022.02.026.

[40] F. Zhong, Z. Chen, M. Xu, G. Zhang, D. Yu, and X. Cheng, "Malware-on-the-Brain: Illuminating Malware Byte Codes With Images for Malware Classification," *IEEE Trans. Comput.*, vol. 72, no. 2, pp. 438–451, 2023, doi: 10.1109/TC.2022.3160357.

[41] F. Rustam, I. Ashraf, A. D. Jurcut, A. K. Bashir, and Y. Bin Zikria, "Malware detection using image representation of malware data and transfer learning," *J. Parallel Distrib. Comput.*, vol. 172, no. 2023, pp. 32–50, 2023, doi: 10.1016/j.jpdc.2022.10.001.

[42] X. Qi, W. Liu, R. Lou, Q. Li, L. Jiang, and Y. Tang, "MC-ISA: A Multi-Channel Code Visualization Method for Malware Detection," *Electron.*, vol. 12, no. 10, 2023, doi: 10.3390/electronics12102272.

[43] H. Deng, C. Guo, G. Shen, Y. Cui, and Y. Ping, "MCTVD: A malware classification method based on three-channel visualization and deep learning," *Comput. Secur.*, vol. 126, p. 103084, 2023, doi: 10.1016/j.cose.2022.103084.

[44] Z. Chen and J. Cao, "VMCTE: Visualization-Based Malware Classification Using Transfer and Ensemble Learning," *Comput. Mater. Contin.*, vol. 75, no. 2, pp. 4445–4465, 2023, doi: 10.32604/cmc.2023.038639.

[45] O. Sharma, A. Sharma, and A. Kalia, "MIGAN: GAN for facilitating malware image synthesis with improved malware classification on novel dataset," *Expert Syst. Appl.*, vol. 241, no. November 2022, p. 122678, 2024, doi: 10.1016/j.eswa.2023.122678.

[46] F. Wang *et al.*, "MalSort: Lightweight and efficient image-based malware classification using masked self-supervised framework with Swin Transformer," *J. Inf. Secur. Appl.*, vol. 83, no. May, p. 103784, 2024, doi: 10.1016/j.jisa.2024.103784.

[47] M. Alshomrani, A. Albeshri, and A. A. Alsulami, "An Explainable Hybrid CNN – Transformer Architecture for Visual Malware Classification," *Sensors*, vol. 25, no. 15, p. 4581, 2025, [Online]. Available: https://doi.org/10.3390/s25154581

[48] T. Messay-kebede, B. N. Narayanan, and O. D.-B. Djaneye-boundjou, "Combination of Traditional and Deep Learning based Architectures to Overcome Class Imbalance and its Application to," in *NAECON 2018 - IEEE National Aerospace and Electronics Conference*, IEEE, 2018, pp. 73–77. doi: 10.1109/NAECON.2018.8556722.

[49] D. V. T. Nguyen and T. N. Nguyen, "HIT4Mal : Hybrid image transformation for malware classification," no. October, pp. 1–15, 2019, doi: 10.1002/ett.3789.

[50] M. Nisa, J. H. Shah, S. Kanwal, M. Raza, and T. Blažauskas, "applied sciences Hybrid Malware Classification Method Using Segmentation-Based Fractal Texture Analysis and Deep Convolution Neural Network Features," *Appl. Sci.*, vol. 10, no. 14, p. 4966, 2020, [Online]. Available: https://www.mdpi.com/2076-3417/10/14/4966

[51] Z. Ren, G. Chen, and W. Lu, "Malware visualization methods based on deep convolution neural networks," *Multimed. Tools Appl.*, vol. 79, no. 15–16, pp. 10975–10993, 2020, doi: 10.1007/s11042-019-08310-9.

[52] A. Pinhero *et al.*, "Malware detection employed by visualization and deep neural network," *Comput. Secur.*, vol. 105, p. 102247, 2021, doi: 10.1016/j.cose.2021.102247.

[53] M. Conti, S. Khandhar, and P. Vinod, "Computers & Security A few-shot malware classification approach for unknown family recognition using malware feature visualization," vol. 122, 2022, doi: 10.1016/j.cose.2022.102887.

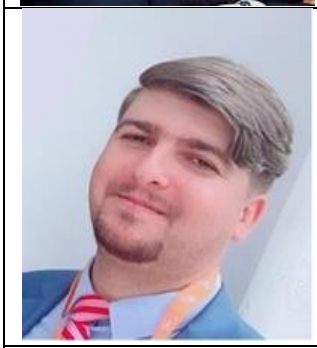
[54] O. Sharma, A. Sharma, and A. Kalia, "Windows and IoT malware visualization and classification with deep CNN and Xception CNN using Markov images," *J. Intell. Inf. Syst.*, vol. 60, no. 2, pp. 349–375, 2023, doi: 10.1007/s10844-022-00734-4.

[55] V. Anandhi, P. Vinod, and V. G. Menon, "Malware visualization and detection using DenseNets," *Pers. Ubiquitous Comput.*, vol. 28, no. 1, pp. 153–169, 2024, doi: 10.1007/s00779-021-01581-w.

[56] W. Al-Khater and S. Al-Madeed, "Using 3D-VGG-16 and 3D-Resnet-18 deep learning models and FABEMD techniques in the detection of malware," *Alexandria Eng. J.*, vol. 89, no. December 2023, pp. 39–52, 2024, doi: 10.1016/j.aej.2023.12.061.

[57] Q. M. Yaseen, E. Oudat, and M. Aldwairi, "Efficient Image-Based Memory Forensics for Fileless Malware Detection Using Texture Descriptors and LIME-Guided Deep Learning," *Computers*, vol. 14, no. 11, p. 467, 2025, [Online]. Available: https://doi.org/10.3390/computers14110467

Prof. Dr. Khattab M Ali Alheeti, I am a computer scientist working on security for autonomous systems, self-driving vehicles, and drones. I obtained my M.Sc. in information technology from the University of Al Al-Bayt University, Mafreq, Jordan 2008, and my PhD in computer science from the University of Essex, Colchester, United Kingdom, in 2017. I carried out a doctoral degree in computer science at the University of Essex. From December 2017, I was a lecturer in the Department of Computer Science, College of Computer, University of Anbar, where I taught in the B.Sc. (computer science) course.
My personal research interests include security, image processing, bioinformatics and computational technology, Petri net modeling and simulating, data mining, and fuzzy logic applications to predictive modeling. Official Website: https://dr-khattabheeti.weebly.com/



Asst. Prof Dr. SMAIL TAHA AHMED received his B.E. and M.Sc. degrees in Computer Science from College of Computer Science and Information Technology, University of Anbar, Anbar, in 2005 and 2009, respectively. He received his Ph.D. degrees in Computer Science from College of Computer Science and Information Technology, Universiti Tenaga Nasional, Putrajaya, Malaysia, in 2018. Currently he is a Assistant Professor in the College of Computer Science and Information Technology, University of Anbar. His research interests Include Image Processing, Computer Vision, Visual Cryptography, Image Quality Assessment, Steganalysis, Image Forgery, Watermarking, Image Enhancement, Texture Recognition, Machine Learning, and Deep Learning.
SCOPUS Account: https://www.scopus.com/authid/detail.uri?authorId=57193324906

Assistant Lecturer Wisam K. Jummar was born in Anbar, Iraq in 1986. He received the B.Sc. in (2009) and the master degree in 2019 from the faculty of the Computer Sciences and Information Technology, University of Anbar, Iraq. He is one of the faculty members within the presidency of Anbar University. His research interest includes image processing, data security, and data mining.
email: wisamkhalid6@uoanbar.edu.iq.