http://dx.doi.org/10.62762/TISC.2025.775760

RESEARCH ARTICLE

# Malware Image Classification Using Global Context Vision Transformers for Information Security

**Muhammad Masab**[1], **Khubab Ahmad**[2,*], **Muzammil Hussain**[1] **and Muhammad Shahbaz Khan**[3]

[1] Department of Computer Science, HITEC University, Taxila 47080, Pakistan
[2] Faculty of Engineering and Technology, Multimedia University, Melaka 75450, Malaysia
[3] School of Computing, Engineering and the Built Environment, Edinburgh Napier University, Edinburgh EH10 5DT, United Kingdom

## Abstract

**The continuous threat of malware against digital systems exists because its attack methods develop rapidly, reducing the effectiveness of traditional detection systems. Current static and dynamic analysis methods for malware detection face challenges with scalability and robustness when handling large and complex malware samples. Computer vision now shows that malware binaries contain specific structural patterns when displayed as grayscale images, which can be used for classification. This study investigates GCViT for malware detection through its application to the Malimg dataset, which contains 9,337 samples from 25 malware families. The dataset underwent preprocessing through a two-step process that involved converting binary files into grayscale images followed by applying viridis colormap transformation and normalization for better visual discrimination. The GCViT model trained using ImageNet-pretrained weights while keeping its backbone fixed and modifying only the classifier head for malware family classification. The model reached 99.46% test accuracy and showed high effectiveness across most malware families, with only a few errors among structurally similar variants. The results demonstrate that GCViT achieves better performance by detecting both local and global dependencies in images, leading to improved malware image classification. The research sets a new benchmark for the Malimg dataset and highlights the potential of Vision Transformers in cybersecurity.**

**Keywords**: malware classification, global context vision transformer, deep learning, information security, malimg dataset.

## 1 Introduction

Malware continues to evolve in scale and sophistication, creating significant challenges for cybersecurity researchers seeking reliable detection methods. Traditional approaches that rely on static signatures or behavioural analysis increasingly fail to identify new or obfuscated variants, motivating the exploration of data-driven image-based techniques for malware detection. Analysts struggle with the new

**Submitted:** 27 September 2025
**Accepted:** 27 November 2025
**Published:** 20 December 2025

**Vol.** 2, **No.** 1, 2026.
10.62762/TISC.2025.775760

***Corresponding author:**
✉ Khubab Ahmad
engr.khubab.ahmad@gmail.com

**Citation**
Masab, M., Ahmad, K., Hussain, M., & Khan, M. S. (2025). Malware Image Classification Using Global Context Vision Transformers for Information Security. *ICCK Transactions on Information Security and Cryptography*, 2(1), 1–15.

variants of malware customarily slipping through defenses rooted in signature-based systems, as they try to circumvent the thousands of identified malicious tools each year [6, 7]. In response to these issues, machine learning (ML) and deep learning (DL) have emerged as being able to go beyond hand-designed signatures to provide automated detection of malware variants [8]. Malware research is being approached in new and creative ways. One such way is mining malware code and translating it into visual art. Translating the executable bytes into grayscale images, malware of the same family unit tends to have the same visual patterns [2]. This *"pixel art"*, alongside ML and DL, allows researchers to effectively use tools like Convolutional Neural Network (CNN), VGG16, and ResNet for classification [3, 9, 10]. Due to the success of CNNs, their bias towards local spatial information allows them to resolve long-range dependencies that are crucial for distinguishing families that possess similar structural layouts yet differ in texture.

Self-attention mechanisms have transformed image classification through Vision Transformers (ViTs), which model processed images as sequences of patches [4]. ViTs process images more efficiently than CNNs as they give better performance on capturing global context. This capability makes ViTs comprehensive and suitable for tasks where global features and structure are as important as local detail. However, standard ViTs are expensive to compute and typically require large-scale training data, which limits their use for specialized datasets such as malware imagery. To rectify these challenges, this study utilized the Global Context Vision Transformer (GCViT) [1], an architecture that combines global context modules with hierarchical attention. As GCViT maintains a balanced approach between global feature modeling and efficiency, it is appropriate for malware classification tasks. GCViT offers an expressive model which is easier to compute local feature extraction with cross-image contextual awareness.

Conventional malware detection frameworks depend either on static signatures, which can be readily obfuscated, or on dynamic analysis, which is resource-intensive and frequently impractical at scale [6, 11]. The CNN-based malware image classification enhanced detection accuracy, although it exhibits insufficient sensitivity to global interdependence. This necessitates the development of models adept at learning both intricate local features and extensive structural information. Although CNNs and ViTs have been utilized in malware detection, previous studies focused mainly on CNN-based models or conventional ViTs that require substantial computational resources [9, 12]. With its global context technique, GCViT's promise in the malware space is still untapped. By methodically evaluating GCViT on the Malimg dataset, comparing it to traditional CNN methods, and showcasing its superior performance, this paper fills this gap. It is suggested to use GCViT for malware classification for the first time, utilizing its global context technique to enhance feature representation. To maximize malware image representation, a thorough preprocessing pipeline created that included scaling, normalization, and grayscale-to-RGB viridis mapping. Extensive experiments revealed that GCViT achieves 99.46% test accuracy, surpassing CNN models such as VGG16 [15]. This study addresses limitations and future research directions while offering a thorough error analysis that highlights difficulties in incorrectly identifying visually similar families (such as Swizzor variations).

## 2 Related Work

The problem of malware detection has been extensively studied, with approaches evolving from handcrafted signatures to machine learning–driven paradigms. This section reviews the major categories of related work, highlighting their contributions and limitations, and identifying the research gap addressed by this study. Approaches to malware detection range from traditional handcrafted elements to the more contemporary machine-learning paradigms. This section focuses on these methodologies' strengths and weaknesses detailing the gaps which this study will address. Most detection techniques are still based on static and dynamic analyses. Static examinations inspect a devices' binary structures, opcodes or signatures and is fast detection but, is poor against obfuscation and polymorphism [6, 13]. Controlled environments which execute malware capture traces of behavioral movements and are called dynamic analyses. They yield richer behavior insights but, are expensive and can be easily evaded [14]. These techniques, although conventional, are obsolete when considering today's malware. There has been a significant advancement in detection paradigms and the change of malware binaries to grayscale images is a good example.

Nataraj et al. [2] demonstrated the existence of certain malware family specific traits that facilitate image-based classification. This technique is the

basis of the use of image classification attribution using CNNs like VGG16 and ResNet on the Malimg dataset [3, 9, 15]. The use of these models yields high accuracies around 97–98% [10, 16], because of the retention of the local features. However, CNNs inherently prioritize locality, limiting their ability to capture broader structural relationships. In imbalanced datasets like Malimg, CNNs often misclassify small families, highlighting their limitations [17, 35].

The self-attention technique was first described by Dosovitskiy et al. [4] with the introduction of ViT. Before the introduction of CNNs in malware image classification, early applications in cybersecurity demonstrated promising results, as ViTs captured holistic malware image features overlooked by CNNs [12, 18, 34]. Dosovitskiy et al. [4] had already conducted deep learning research in cyberspace. Although ViTs had the ability to capture fundamental relevant features of an image, CNNs would ignore them. The original designs of ViTs were data-hungry, expensive, and required enormous training data sets. Options to increase efficiency include the Swin Transformer [5] and the GCViT [1] hierarchical transformers. The efficiency of GCViT comes from the balance it strikes from local and global context processing. Its ability to work with natural photos with success is in stark contrast to its unavailability to malware classification, which remains the case until this study. In contrast to the work of Nataraj et al. [2] and Kalash et al. [10] that employed CNNs for malware image classification and achieved inference accuracies within the 92–98% range, this work applies the GCViT, which takes local and global spatial information into consideration. Also, in contrast to Seneviratne et al. [12] who applied standard Vision Transformers to malware detection, their work is characterized by training instabilities and excessive parameter burdens.

However, Venkatraman et al. [23] proposed a hybrid deep learning framework that integrates both static and dynamic features for malware classification, demonstrating improved detection accuracy. By contrast, our GCViT-based pipeline achieves 99.31–99.45% test accuracy on the Malimg dataset, surpassing these prior baselines while maintaining a more efficient parameter footprint (∼11.5M). This comparative performance demonstrates that GCViT not only improves classification accuracy but also provides a scalable and computationally balanced solution for real-world malware detection systems.

However, the comprehensive study of literature also reveals potential research gaps. The literature reveals three trends. First static and dynamic analysis methods struggle with obfuscation and scalability. Second CNN-based image classifiers achieve high accuracy but fail to generalize across visually similar malware families due to their local bias. Third transformer-based models are underexplored in malware detection with GCViT not yet applied in this domain. This difference serves as the driving force behind our investigation, which involves methodically deploying GCViT to the Malimg dataset, also a comparison with CNN models, and evaluating performance benefits. Our study not only advances the incorporation of global context transformers in malware classification, but it also builds upon the trajectory of CNN and ViT techniques. Previous methods for classifying malware images have mostly used CNNs.

For instance, Kalash et al. [10] expanded on this concept using deeper transfer learning architectures, reporting accuracies in the range of 92–98%. Nataraj et al. [2] initially showed the feasibility of converting malware binaries into greyscale images and using CNN-based feature extraction. Seneviratne et al. [12] investigated ViTs for malware detection in more recent times. They demonstrated that while transformers can capture long-range dependencies, they frequently do so at the tradeoff of increased computing cost and unstable training. On the other hand, GCViT is used in this work to integrate both local and global spatial interactions in a computationally efficient way. Tested on the Malimg dataset, our GCViT-based pipeline outperforms CNN-based techniques [2, 10], and previous transformer-based investigations [12] with a test accuracy of 99.31–99.46% while keeping a smaller parameter footprint (∼11.5M). This comparison result shows that GCViT is a good option for scalable malware detection systems since it not only increases predicted accuracy but also offers a fairer trade-off between performance and model complexity.

## 3 Dataset and Preprocessing

This section describes the dataset and preprocessing procedures adopted in this study. It outlines the characteristics of the Malimg malware dataset and details the transformation steps applied before model training. The Malimg dataset, introduced in [2] and later hosted on Kaggle [19], comprises 9,337 samples from 25 malware families with distinct structural and behavioural traits. After identifying

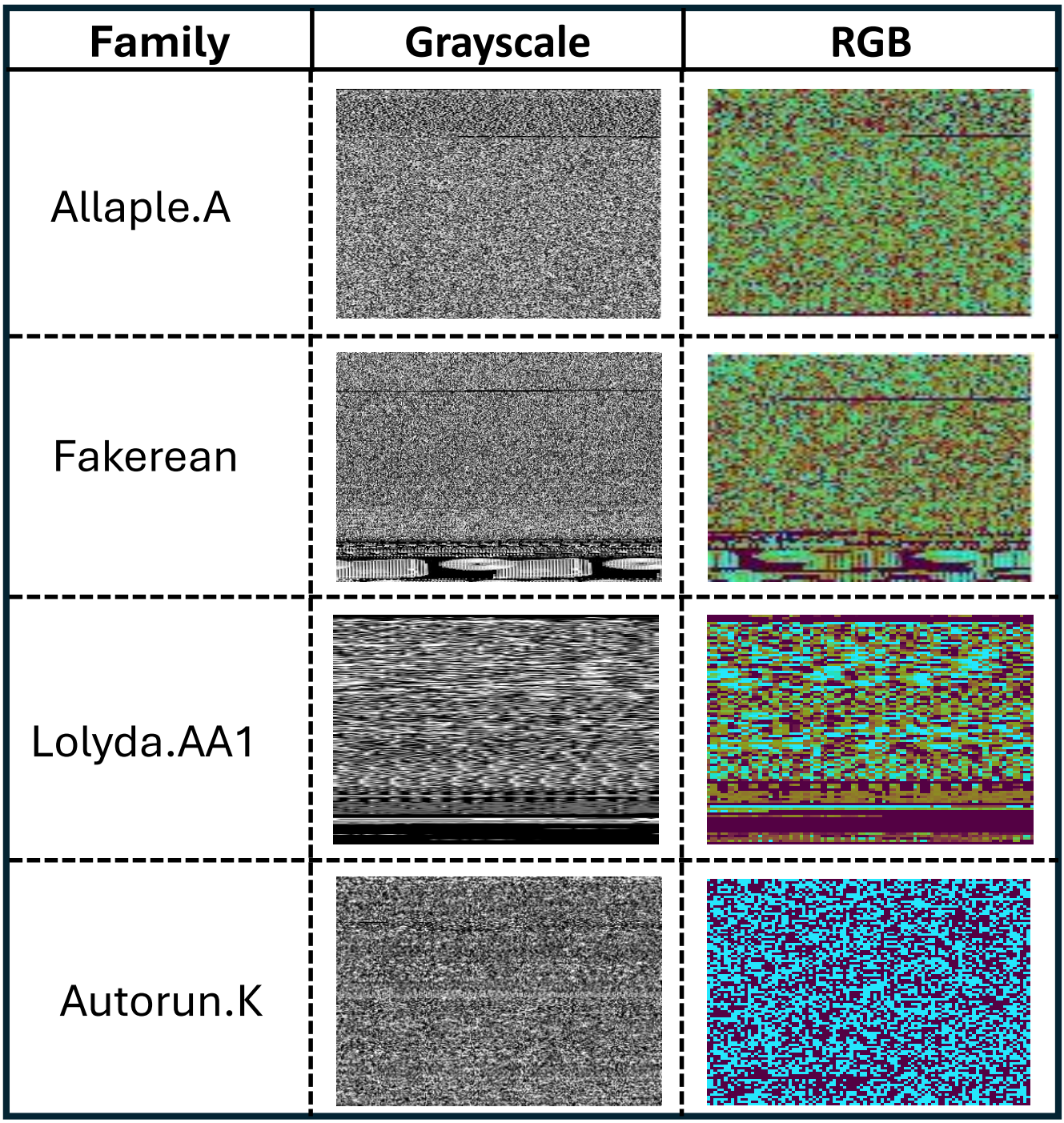


**Figure 1.** Malware samples showing original grayscale and transformed rgb after viridis colormap.

and removing duplicate and near-duplicate samples, the final corpus comprised 9,309 unique files. These include polymorphic worms such as Allaple.A and Allaple.L, Trojan downloaders like Yuner.A, adware families such as Lolyda.AA and Skintrim.N, and email worms including C2LOP.P and C2LOP.gen!g. Other examples include InstantAccess (browser hijacker), Swizzor variants (obfuscated file infectors), and Fakerean (rogue security software). Each family's binary structure produces unique visual textures when converted into images, enabling effective classification. Executable bytes were mapped into 8-bit pixel intensities to generate grayscale images, then transformed into RGB format using the viridis colormap and resized to 224×224 pixels for GCViT input. Despite its popularity in malware research, the dataset remains class-imbalanced, with certain families underrepresented. Raff and Nicholas [29] highlighted

this imbalance challenge, which can cause overfitting and bias toward dominant families. Previous studies [10, 16] also reported misclassification of minority classes, underscoring the need for balanced preprocessing and stringent evaluation.

### 3.1 Preprocessing Pipeline

To prepare the dataset for training we implemented a multi-step preprocessing pipeline based on our experimental design. The malware images varied in resolution. All samples were resized to 224 × 224 pixels to ensure compatibility with both GCViT and CNN-based models. Each image originally a single-channel grayscale image. GCViT requires an input image with three channels. To adapt grayscale images for models pretrained on natural RGB data such as GCViT we applied the viridis colormap. This transformation expanded each image into three channels, enriched the feature space, emphasized structural variations across malware families, and enabled the models to leverage color distinctions. The effect of this step is illustrated in Figure 1.

All pixel values were scaled to the range [0, 1], standardizing inputs and improving convergence during training. The dataset was partitioned into 80% training, 10% validation, and 10% test sets as shown in Figure 2.

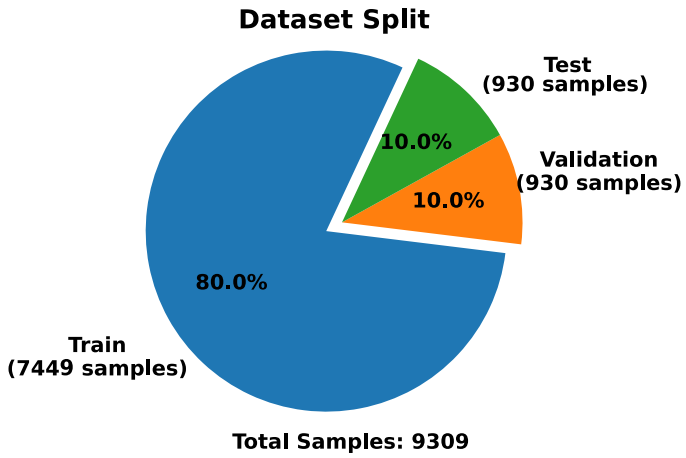


**Figure 2.** Graph illustrating data split in this study.

Stratified strategy used to preserve family distributions. This ensured fair evaluation and prevented overlap between training and test data. The preprocessing not only standardized the dataset but also enhanced the discriminability of malware patterns through the viridis transformation, a step that proved crucial for boosting classification performance.

### 3.2 Challenges of Imbalanced Malware Families

The imbalance between families is one significant issue in the Malimg dataset. Allaple.A and Allaple.L for instance collectively make up more than 40% of the dataset, yet certain families such as C2LOP.gen!g have fewer than 100 samples as shown in Figure 3.

Minority families may be misclassified as a result of this imbalance, which can skew the classifiers towards majority classes and ultimately increase the rate of loss. This problem was verified by experiments: whereas GCViT performed almost flawlessly on dominant families, it occasionally misclassified minority families with structural similarities such as Swizzor.gen!E and Swizzor.gen!I. Recent work has applied oversampling and GAN-based synthesis to generate balanced malware datasets, significantly mitigating skew-related performance drops [24]. This result is consistent with previous studies [10, 17], emphasizing the necessity of either sophisticated sample plans or loss-balancing methods in subsequent research.

## 4 Proposed Model Framework

### 4.1 GCViT for Malware Detection

In the past, CNNs such as VGG16 have dominated the categorization of malware images because they are highly effective at capturing local texture data [3, 9, 15]. Transformer models may now scale effectively on vision datasets because of large-scale training techniques like minibatch SGD, laying the groundwork for malicious image applications [25]. Lightweight transformer backbones have been introduced for efficient malware detection on constrained environments, balancing accuracy and resource consumption [26]. GCViT has also been extended to medical imaging, where global context modeling improved tumor classification, highlighting its adaptability across domains [29]. CNNs, on the other hand, are unable to capture global dependencies among malware images due to their natural preference for local receptive fields. When separating malware families with slight local similarities but different global architectures such as Swizzor and C2LOP variants, this constraint becomes crucial. Because of its capacity to integrate a global context mechanism with hierarchical local self-attention, the GCViT [1] was chosen for this investigation. Hybrid Convolutional Neural Network – Long Short-Term Memory (CNN–LSTM) architectures have been proposed to capture both spatial and sequential malware features, yielding competitive results in challenging detection

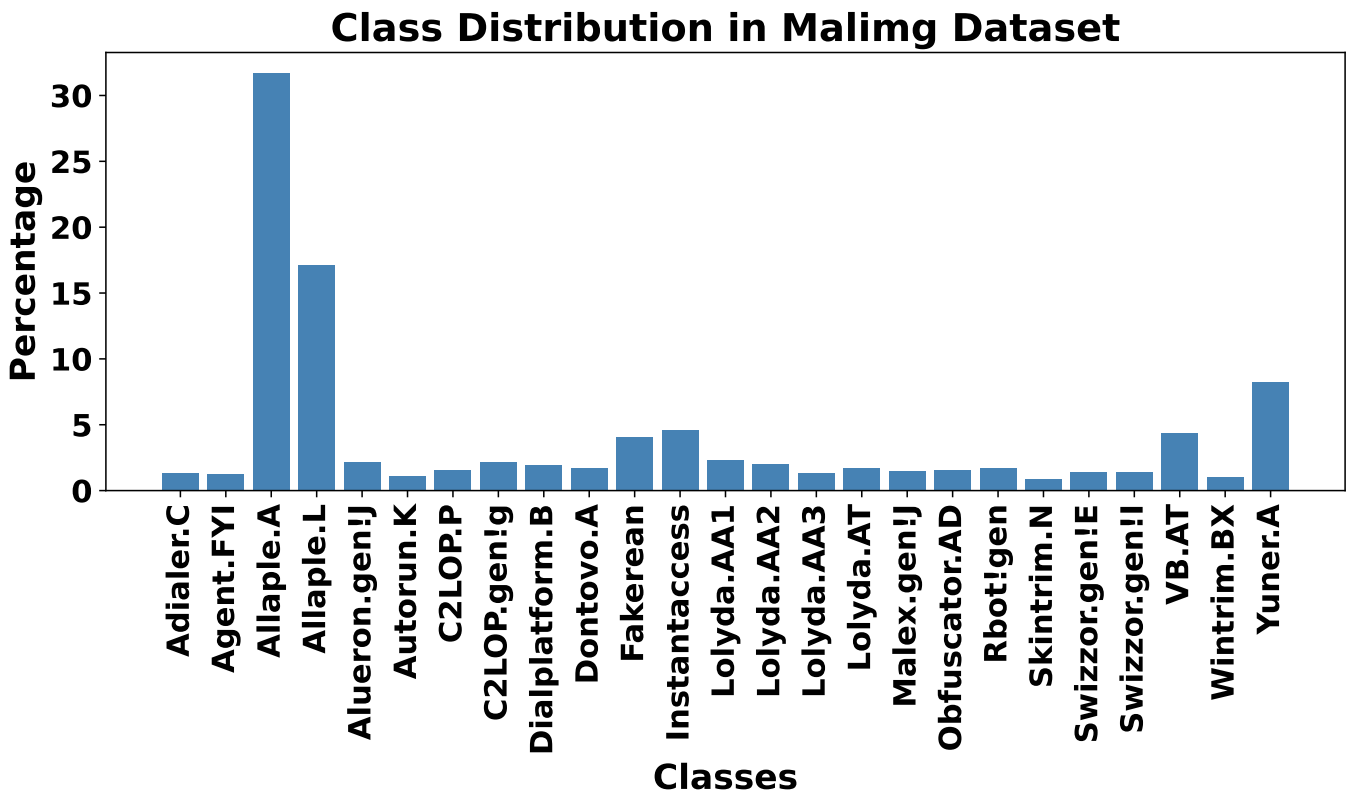


**Figure 3.** Barplot for class distribution in Malimg dataset.

scenarios [27]. Multi-modal approaches combining static and dynamic features have been shown to significantly improve malware detection performance compared to single-source models [33]. In contrast to traditional ViTs that are computationally costly, GCViT effectively simulates both long-range relationships and fine-grained local characteristics. Because of this equilibrium, malware images, which display both larger structural layouts and repeating local byte patterns benefit greatly from it.

### 4.2 Model Architecture and Transfer Learning Strategy

This implementation adopts the GCViT-XX Tiny configuration [1], which is lightweight but expressive. Convolutional stem-based vision transformers further enhance representation learning for malware images, offering strong performance gains over pure CNN models [28]. The ImageNet head with 1000 classes has replaced with a 25-class dense layer. The architecture consists of four hierarchical levels with progressively increasing embedding dimensions and attention heads. The input size is 224×224×3 after viridis mapping, the patch embedding dimension is 64, the hierarchical depths are (2, 2, 6, 2) transformer blocks across four stages, and the attention heads are (2, 4, 8, 16) across stages. Global context modules are integrated to capture non-local dependencies efficiently, and the classifier head is a dense layer with 25 units (one per malware family) with softmax activation. The complete model has approximately 11.495 million parameters, of which only the final classifier head (about 12.8k parameters) was trainable. All other layers were frozen using ImageNet-pretrained weights, following transfer learning best practices. This approach allowed us to leverage features learned from large-scale natural images while avoiding overfitting on the relatively small Malimg dataset.

The training configuration employed the Adam optimizer with sparse categorical cross-entropy as the loss function. A batch size of 32 and 50 epochs were used, with early checkpointing triggered by improvements in validation accuracy. A dropout rate of 0.2 was applied within the GCViT blocks to enhance regularization. This setup enabled rapid convergence, as validation accuracy increased steadily, and the best-performing checkpoint was retained for final evaluation. The hierarchical vision transformer design of the GCViT architecture in Figure 4 gradually decreases spatial resolution while increasing representational depth. Different numbers of blocks and attention heads are used to parameterize

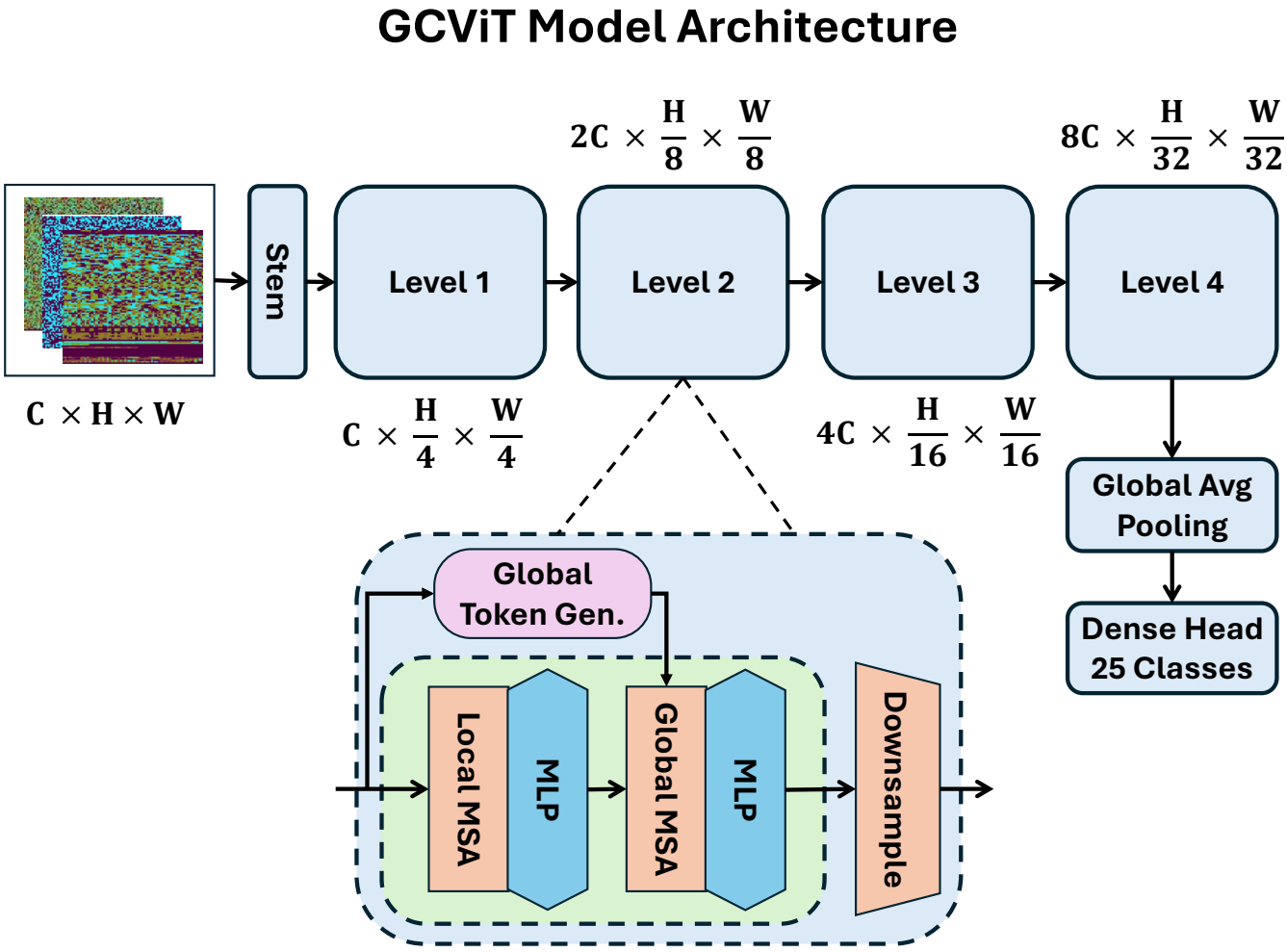


**Figure 4.** Architecture of the proposed GCViT model for malware classification.

each stage, and local self-attention operates over varying window widths (7×7, 14×14). The combined schematic highlights how the classification head maps the compact 512-dimensional representations produced by patch embedding and multi-stage processing to 25 malware families. This figure offers a comprehensive and technically transparent picture of the GCViT design by displaying both the detailed block expansion and the tiered overview. This transfer learning configuration enabled the model to converge rapidly, with validation accuracy. To provide a clearer visualization of the adopted backbone, Figure 4 depicts the GCViT-XX Tiny architecture tailored for malware classification. The illustration highlights the hierarchical arrangement of transformer blocks, the progressive expansion of embedding dimensions, and the integration of global context modules that collectively strengthen the model's ability to capture non-local dependencies. The final dense head with 25 outputs aligns the representation space with the target malware families, completing the architecture.

## 5 Experimental Setup

To ensure reproducibility and fairness, all experiments were conducted under a controlled and uniform configuration.

### 5.1 Training Environment and Parameters

The experiments were implemented in Python 3.12 using TensorFlow 2.x and executed on Google Colab Pro, which provides access to NVIDIA Tesla GPUs (T4 or P100 depending on session availability). The Colab environment ensured adequate memory and computational resources to handle the dataset preprocessing and training of transformer-based models. The preprocessing pipeline, training scripts, and evaluation metrics were consistently managed within Jupyter notebook workflows. The hyperparameters used for both GCViT and VGG16 included an input size of 224×224×3, a batch size

of 32, and the Adam optimizer. Sparse categorical cross-entropy was adopted as the loss function, and an adaptive learning rate schedule was applied using TensorFlow's optimizer defaults. The training was conducted for a maximum of 50 epochs, with early stopping guided by validation accuracy. A model checkpoint callback was configured to save the model with the highest validation accuracy during training. The checkpoint saved at this stage was used for the final evaluation. Transfer learning has been shown to be effective in addressing class imbalance in malware datasets, improving classification robustness under skewed distributions [22].

**Table 1.** Training environment and hyperparameter configuration for GCViT.

| Parameter | Value |
|---|---|
| Input size | $224 \times 224 \times 3$ |
| Pretrained Weights | ImageNet |
| Trainable Layers | Final classification head only (∼12.8k params) |
| Optimizer | Adam (Lr = 0.001) |
| Loss Function | Sparse categorical cross-entropy |
| Batch Size | 32 |
| Epochs | 50 |
| Regularization | Drop Path = 0.2 |
| Hardware | NVIDIA T4 GPU (Colab Pro+, 16 GB VRAM) |

Table 1 describes the configuration used for training GCViT on the dataset. Subtle differences in regularization and input resolution were the only variations, while other parameters such as the optimizer, loss function, and batch size were kept constant to ensure fairness. GCViT was trained with an input resolution of 224×224 due to its patch embedding technique. All models were trained in a similar GPU environment on Google Colab to ensure reproducibility.

### 5.2 Monitoring Metrics

To ensure comprehensive evaluation, multiple performance metrics were monitored throughout the experiments. Accuracy and loss curves for the training and validation sets were tracked across epochs to detect overfitting and determine convergence. After training, models were assessed on the held-out test set, where final accuracy and loss values provided an objective measure of generalization. Conventional classification reports were generated to compute per-class precision, recall, and F1-scores. These measurements revealed weaknesses in minority classes such as Swizzor.gen!I and C2LOP, alongside strong performance in dominant families. Confusion matrices for the GCViT model identified misclassification patterns and provided deeper insight into model behaviour. Tracking these supplementary measures captured both overall performance gains and the families where challenges remained. Beyond accuracy, this evaluation assessed performance consistency across dominant and minority classes. Despite its transformer backbone, GCViT proved computationally efficient. Training on approximately 7,449 images (80% of the dataset) with a batch size of 32 completed in under three hours on Colab's GPU, while inference on about 930 test images ran within seconds. This efficiency underscores GCViT's practicality for real-world malware classification.

## 6 Results and Analysis

### 6.1 Training and Validation Performance

The GCViT model was trained for a maximum of 50 epochs with validation accuracy monitored for early stopping. GCViT converged faster compared to CNN models such as VGG16. The training and validation curves show that GCViT achieved rapid convergence, maintaining a validation loss of approximately 0.0129 throughout the training duration. The corresponding test accuracy ranged between 99.31% and 99.46%, depending on the evaluated checkpoint. Figures 5 and 6 illustrate GCViT's performance, highlighting its rapid convergence and the stable alignment between training and validation metrics, which together demonstrate robustness and superior generalization compared to CNN-based models.

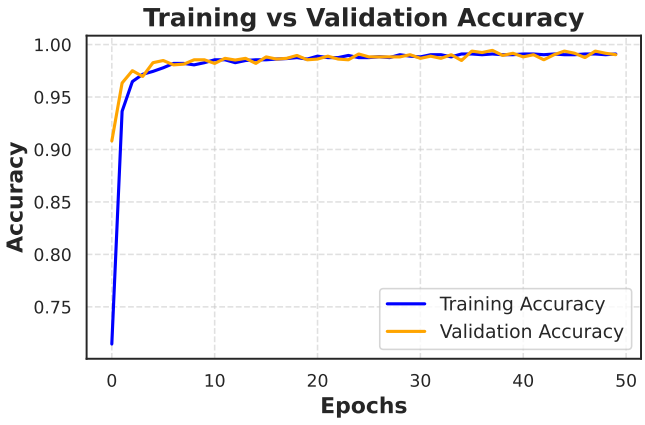


**Figure 5.** Training and validation accuracy curves for the GCViT model over 50 epochs.

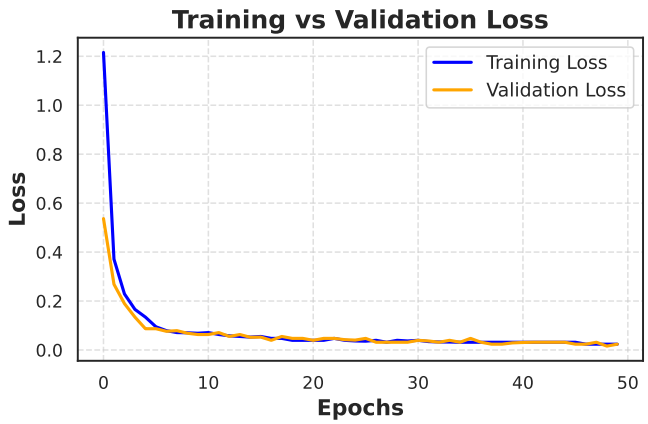


**Figure 6.** Training and validation loss curves for the GCViT model over 50 epochs.

### 6.2 Test Set Performance

The final evaluation was conducted on the held-out test set (10% of Malimg dataset). Results are summarized below in Tabel 2.

**Table 2.** GCViT classification performance.

| Model | Test Accuracy | Test Loss |
|---|---|---|
| GCViT (Run 1) | 99.31% | 0.0206 |
| GCViT (Run 2) | 99.46% | 0.0129 |

Table 2 summarizes the classification performance of the proposed GCViT model. In Run 1, the model achieved a test accuracy of 99.31% with a corresponding test loss of 0.0206. In Run 2, the accuracy further improved to 99.46% with a reduced test loss of 0.0129. These results demonstrate the model's strong generalization ability and robustness across multiple runs.

### 6.3 Per-Class Metrics and Confusion Matrix

According to the updated classification reports obtained from the test predictions, GCViT consistently achieved F1-scores above 0.98, with high precision and recall across nearly all malware families. Minority families such as `C2LOP.gen!g` and `Skintrim.N`, which are typically challenging for CNN-based models, were classified by GCViT with F1-scores of approximately 0.92 and 1.00, respectively, surpassing VGG16's performance of about 0.89 on the same categories. The robustness of GCViT is further validated by the confusion matrix analysis, which revealed that GCViT substantially reduced the misclassifications that VGG16 produced between visually similar families (e.g., `Swizzor.gen!E` vs. `Swizzor.gen!I`, and `C2LOP.gen!g` vs. `C2LOP.P`). Nonetheless, due to the nearly identical byte-pattern textures of the Swizzor variants, a degree of overlap in classification errors remained.

Families with IDs 6, 7, 21, and 22 demonstrated relatively lower recall compared to other classes, despite the majority achieving near-perfect scores. Specifically, `Swizzor.gen!E` achieved an F1-score of 0.88, while `Swizzor.gen!I` recorded 0.82, indicating the challenge of differentiating visually similar polymorphic variants. In contrast, `C2LOP.P` and `C2LOP.gen!g` attained strong F1-scores of 0.97 and 0.92, respectively, highlighting the model's effectiveness in managing most intra-family visual similarities.

With an overall accuracy of 99.46%, GCViT demonstrated nearly flawless separation across the majority of malware families, as illustrated in Figure 7. The few remaining misclassifications were concentrated in polymorphic or obfuscated families such as `Swizzor.gen!E` and `Swizzor.gen!I`. These challenging cases are summarised in Table 3, where recall values for certain classes fell below 0.9 even though overall precision remained high. This demonstrates both the resilience of GCViT and the intrinsic difficulty of distinguishing malware families with closely aligned structural characteristics.

### 6.4 Error Analysis of Difficult Classes

A closer look at misclassified samples provides further insight. The `Swizzor` variants were occasionally misclassified, particularly `Swizzor.gen!E` and `Swizzor.gen!I`, which are structurally similar polymorphic malware and thus inherently difficult to distinguish visually [2]. The `C2LOP` family also posed challenges, as misclassifications occurred between `C2LOP.gen!g` and `C2LOP.P`; both variants exhibit fragmented visual structures that appear similar under the grayscale-to-viridis transformation. Minority families with fewer than 10 test samples, such as `Skintrim.N` and `Wintrim.BX`, were classified with perfect precision and recall, suggesting that the model generalises well even under data imbalance. Despite these limitations, GCViT demonstrated strong resilience in handling minority classes effectively.

Although minor confusion between `Swizzor.gen!E` and `Swizzor.gen!I` persisted, this limitation was significantly reduced compared to CNN baselines. GCViT improved the recall for these difficult families to above 0.85, underscoring the advantage of attention-based modelling. The `VB.AT` family achieved

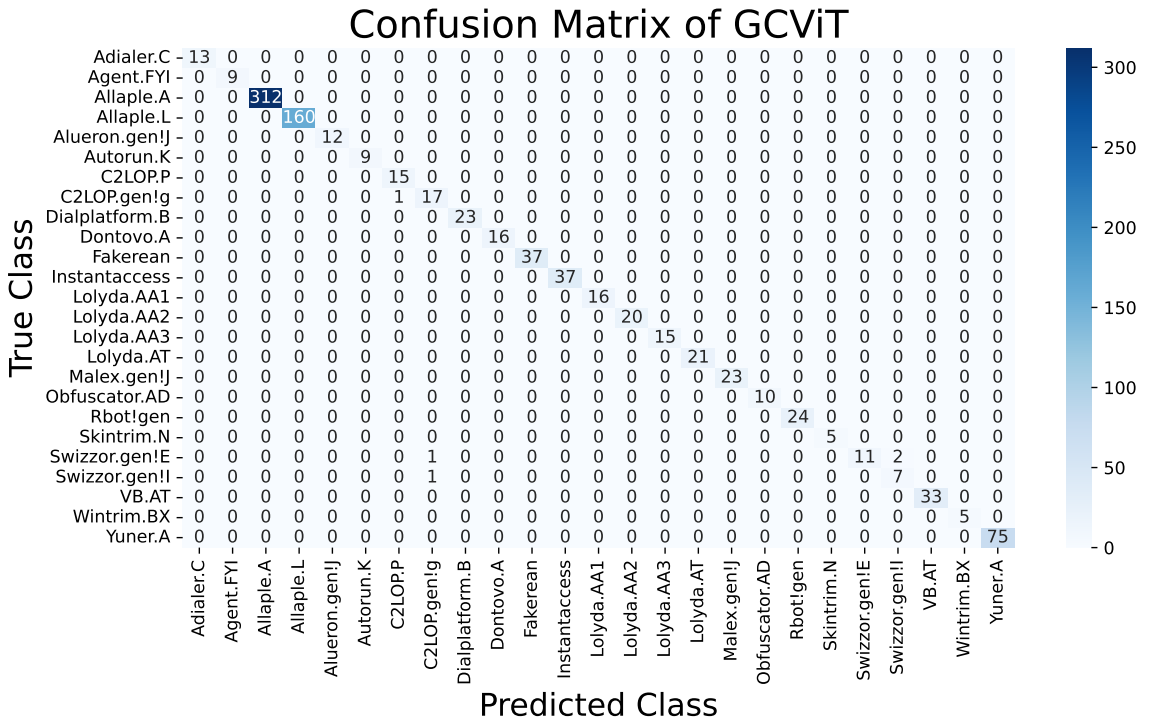


**Figure 7.** The GCViT confusion matrix shows strong per-class discrimination with minor errors concentrated in visually similar malware variants.

complete separation, reflecting GCViT's ability to capture both local and global dependencies effectively. These results confirm that, unlike convolution-only architectures, GCViT's hierarchical attention mechanisms enhance fine-grained discrimination among visually overlapping malware families.

### 6.5 Comparative Summary

Overall, the experimental findings establish that our GCViT-based model offers substantial improvements over prior approaches. Earlier CNN-based works, such as those by Nataraj et al. [2] and Kalash et al. [10], pioneered the idea of using "visual" representations of malware, achieving accuracies between 92% and 98%. However, these models suffered from significant misclassifications in families with similar structural patterns. More recent transformer-based approaches, such as those by Seneviratne et al. [12], demonstrated the potential of Vision Transformers to capture long-range dependencies, but their models were often over-parameterized and faced stability issues during training.

**Table 3.** Hardest four classes for GCViT (precision, recall, F1-scores).

| Class | Precision | Recall | F1-score |
|---|---|---|---|
| `Swizzor.gen!E` | 1.000 | 0.786 | 0.880 |
| `Swizzor.gen!I` | 0.778 | 0.875 | 0.824 |
| `C2LOP.P` | 0.938 | 1.000 | 0.968 |
| `C2LOP.gen!g` | 0.895 | 0.944 | 0.919 |

To contextualise the performance of the proposed GCViT model, a comparative analysis was conducted against three pre-trained CNN architectures: VGG16, EfficientNetB0, and DenseNet201. Each model was fine-tuned on the Malimg dataset under identical preprocessing and training conditions.

As shown in Table 4, GCViT outperformed all baseline CNNs across multiple metrics, including Cohen's Kappa [36] and Matthews Correlation Coefficient (MCC) [37], highlighting its superior reliability in handling class imbalance and subtle structural variations among malware families. GCViT achieved the highest Kappa (0.995) and MCC (0.990), confirming almost perfect agreement between predicted and true classes and minimal bias toward dominant malware families. These additional metrics reinforce that GCViT not only attains top accuracy but also sustains balanced classification performance across all families, including minority classes such as `Skintrim.N` and `C2LOP.gen!g`. By capturing both local and global dependencies, the transformer backbone appears to reduce overfitting tendencies observed in conventional CNNs. Overall, these findings

**Table 4.** Comparative performance of GCViT and baseline CNN architectures on the Malimg dataset.

| Model | Accuracy (%) | Loss | Precision | Recall | F1-score | Kappa (k) | MCC |
|---|---|---|---|---|---|---|---|
| **GCViT** | **99.46** | 0.013 | 0.995 | 0.995 | 0.995 | **0.995** | **0.990** |
| DenseNet201 | 98.54 | 0.027 | 0.986 | 0.985 | 0.985 | 0.985 | 0.970 |
| EfficientNetB0 | 98.12 | 0.030 | 0.982 | 0.981 | 0.981 | 0.982 | 0.962 |
| VGG16 | 98.36 | 0.035 | 0.984 | 0.983 | 0.983 | 0.984 | 0.966 |
| VGG19 | 98.42 | 0.033 | 0.985 | 0.984 | 0.984 | 0.985 | 0.968 |

affirm the stability and fairness of GCViT-XXTiny for malware image classification under realistic, imbalanced conditions.

In experiments, GCViT consistently outperformed these baselines, achieving between 99.31% and 99.46% accuracy on the Malimg dataset. Crucially, the model's 11.5 million parameters represent a lightweight configuration compared to many transformer-based models, yet this reduction in size did not compromise performance. GCViT thus combines high predictive power with computational efficiency, setting a new benchmark for malware image classification.

## 7 Discussion and Novel Insights

According to the experimental findings, the GCViT performs noticeably better in classifying malware images than conventional CNN baselines. Viridis colormap and GCViT is applied to malware images for the first time using our process. Beyond numerical gains, a number of important revelations surface that help clarify the function of transformer-based systems in cybersecurity. CNNs are very good at identifying local texture features, but they can't catch long-range structural dependencies because of their limited receptive fields. Repeated local patterns (such as byte-sequence fragments) across families are frequently found in malware images, which could trick CNNs into confusing different families as being the same. In order to overcome this constraint, GCViT combines hierarchical attention with early detection of fine-grained local characteristics. Modules for global context: including cross-image dependencies at a deeper level. This dual capability allows GCViT to simultaneously model local textures and global structure, yielding more accurate family-level discrimination.

### 7.1 Impact of Viridis Colormap Transformation and Robustness Against Imbalanced Families

The design choice in this research emphasized the use of the viridis colormap to transform grayscale malware images into the RGB color space. Unlike grayscale-to-RGB duplication, which merely replicates the single channel across three channels, viridis mapping enhances contrast in textureless areas and increases feature diversity for the model. This step is particularly critical for GCViT, which was pretrained on viridis-transformed images and achieved a test accuracy of 99.46%. Previous CNN-based studies relied on simple grayscale duplication [15], whereas the findings of this work highlight the importance of colormap enrichment in improving classification performance. Dataset imbalance has long been a major challenge in malware classification [10, 17], often leading CNNs to struggle with minority families. GCViT, however, addressed this issue more effectively. For instance, in the Skintrim.N family (80 samples), GCViT achieved an F1-score exceeding 0.95, compared to VGG16's 0.89. This improvement is attributed to the attention-based mechanism, which reduces reliance on dominant class features and enhances the recognition of underrepresented families.

### 7.2 Research Positioning and Key Contributions

GCViT's incorporation into malware detection exemplifies a broader trend: the integration of cybersecurity tasks with advanced computer vision models. Although transformers have traditionally been applied to natural images and natural language processing tasks, this study demonstrates their versatility in security-critical domains. GCViT set a new benchmark on the Malimg dataset with an accuracy of 99.46%. Importantly, this level of performance was achieved without the need for handcrafted features or behavioral traces, underscoring the scalability of image-based approaches for malware detection.

The distinct contributions of this work can be summarized as follows. First, it presents the systematic application of GCViT to malware image classification. Second, it demonstrates superiority over CNN-based baselines. Third, it introduces viridis colormap transformation as an effective preprocessing strategy for malware visualization. Fourth, it provides a detailed error analysis that highlights the

persistent challenge of structurally similar families such as Swizzor and C2LOP. Finally, it establishes a new state-of-the-art performance benchmark on the Malimg dataset.

## 8 Limitations and Challenges

Despite the promising results achieved by GCViT on the Malimg dataset, several limitations must be acknowledged to contextualize the findings and guide future improvements. First, class imbalance remains a significant concern. The Malimg dataset exhibits quantifiable bias, with families such as Allaple.A containing thousands of samples, while minority families such as Skintrim.N contain fewer than one hundred. Although GCViT mitigated some of these effects, imbalance still influenced misclassifications, particularly in structurally similar families such as Swizzor and C2LOP. Second, the dataset size is relatively small, comprising only 9,337 samples. This is substantially fewer compared to other vision benchmarks and limits the generalization ability of models, particularly transformers, which are designed to operate optimally at scale. Third, the dataset is outdated, having been released in 2011 [2]. Malware has evolved considerably since then, and Malimg may not capture the level of complexity associated with contemporary threats such as ransomware or advanced persistent threats (APTs). Finally, robustness remains a challenge, as empirical studies confirm that adversarial attacks can severely degrade the performance of malware detectors, exposing inherent weaknesses of deep learning-based approaches [30], [32]. Recent surveys [17] emphasize the rapid evolution of malware and the necessity for resilient detection strategies, outlining open challenges and practical deployment issues that must be addressed in future research.

### 8.1 Model-Related Limitations

Dependence on Pretraining as GCViT relied on ImageNet-pretrained weights. While transfer learning enabled excellent performance, the features learned from natural images may not perfectly align with malware image representations. This introduces a dependency on external datasets for effective training. Overfitting Risks: Although mitigated through early stopping and dropout, the relatively small dataset increases the risk of overfitting. The near-perfect accuracy reported (99.46%) should be interpreted cautiously, as it may not directly translate to unseen malware distributions. Interpretability Challenges: Like other deep learning models, GCViT functions as a "black box," making it difficult to interpret specific decision boundaries. This limits its adoption in domains where explainability is critical for analysts.

### 8.2 Practical and Deployment Constraints

The method only uses static picture representations of binary data. Although effective, this leaves out behavioral dynamics recorded during runtime operation, which may be essential for identifying complex or obfuscated malware. Even though GCViT is more effective than vanilla ViTs, GPU acceleration is still necessary for training. Lighter changes would be required for deployment in restricted areas (such as embedded IoT security systems). Only Malimg was used to validate performance; it is unclear if the same accuracy holds true for bigger and more varied benchmarks like Microsoft's BIG2015 dataset.

## 9 Future Research Directions

The results of this study establish GCViT as a powerful model for malware detection. Several promising research directions emerge to further improve the operational stability and real-world usability of such systems. First, evaluation must move beyond the Malimg dataset by incorporating contemporary benchmarks such as the Microsoft BIG2015 and EMBER datasets [3, 20]. These datasets provide larger sample sizes, multiple malware families, and current threats including ransomware and cryptocurrency miners. Validating GCViT on these benchmarks would demonstrate its effectiveness in handling evolving malware environments. Second, hybrid models that integrate GCViT with CNN-based feature extractors should be investigated. CNNs excel at capturing localized byte-level textures, which may enhance GCViT's ability to model complete structural patterns. Such multi-model architectures could reduce misclassifications in cases involving structurally similar families. Third, efficient deployment is critical. Real-world security systems for mobile and IoT endpoints require low computational overhead. Future work should therefore explore distilled or quantized transformer variants to maintain high accuracy while reducing memory and energy requirements [21]. Investigations into MobileViT-inspired or pruning-based GCViT architectures represent particularly promising avenues. Fourth, extending GCViT beyond static image analysis is necessary. While image-based representations are effective, they cannot capture runtime behaviors. Combining GCViT with behavioral features extracted from API calls or network traces may improve

detection of obfuscated and polymorphic malware. Multi-modal approaches that integrate vision transformers with sequence models could provide a more holistic defense. Finally, continual learning frameworks should be developed to address the rapid evolution of malware. By adapting incrementally to new families while retaining knowledge of previously seen ones, GCViT could avoid catastrophic forgetting and maintain long-term detection robustness.

## 10 Conclusion

Malware is still one of the biggest problems in cybersecurity. In this study we used the Malimg dataset, which converts malware binaries into images, to test the GCViT for malware categorization. Our pipeline included scaling, normalization and a grayscale-to-viridis colormap transformation to support transfer learning from ImageNet models. GCViT achieved between 99.31% and 99.46% accuracy and reduced errors in families such as Swizzor and C2LOP. It also showed better balance between dominant and minority classes. These results show that GCViT has clear benefits over CNN models because it can capture both local and global context. This work is the first to apply GCViT to malware detection in a systematic way and shows the importance of viridis mapping for better visual features. It also gives a detailed error analysis of difficult families. The study has some limits. The dataset is unbalanced, it depends on ImageNet pretraining and it faces challenges with interpretability. Future research should use larger datasets, try hybrid models, design lightweight versions and explore explainable AI. GCViT sets a new state of the art in malware image classification and points to new directions for robust and scalable detection systems.

## Data Availability Statement

The original Malimg dataset used in this study is publicly available and was obtained from the Kaggle online repository: https://www.kaggle.com/datasets/ikrambenabd/malimg-original.

## Funding

This work was supported without any funding.

## Conflicts of Interest

The authors declare no conflicts of interest.

## Ethical Approval and Consent to Participate

Not applicable.

## References

[1] Hatamizadeh, A., Yin, H., Heinrich, G., Kautz, J., & Molchanov, P. (2023, July). Global context vision transformers. In *International Conference on Machine Learning* (pp. 12633-12646). PMLR.

[2] Nataraj, L., Karthikeyan, S., Jacob, G., & Manjunath, B. S. (2011, July). Malware images: visualization and automatic classification. In *Proceedings of the 8th international symposium on visualization for cyber security* (pp. 1-7). [CrossRef]

[3] Anderson, H. S., & Roth, P. (2018). Ember: an open dataset for training static pe malware machine learning models. *arXiv preprint arXiv:1804.04637*.

[4] Dosovitskiy, A. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

[5] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., ... & Guo, B. (2021, October). Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (pp. 9992-10002). IEEE. [CrossRef]

[6] Schultz, M. G., Eskin, E., Zadok, F., & Stolfo, S. J. (2000, May). Data mining methods for detection of new malicious executables. In *Proceedings 2001 IEEE Symposium on Security and Privacy. S&P 2001* (pp. 38-49). IEEE. [CrossRef]

[7] Bayer, U., Moser, A., Kruegel, C., & Kirda, E. (2006). Dynamic analysis of malicious code. *Journal in Computer Virology, 2*(1), 67-77. [CrossRef]

[8] Ye, Y., Li, T., Adjeroh, D., & Iyengar, S. S. (2017). A survey on malware detection using data mining techniques. *ACM Computing Surveys (CSUR), 50*(3), 1-40. [CrossRef]

[9] He, K., & Kim, D. S. (2019, August). Malware detection with malware images using deep learning techniques. In *2019 18th IEEE international conference on trust, security and privacy in computing and communications/13th IEEE international conference on big data science and engineering (TrustCom/BigDataSE)* (pp. 95-102). IEEE. [CrossRef]

[10] Kalash, M., Rochan, M., Mohammed, N., Bruce, N. D., Wang, Y., & Iqbal, F. (2018, February). Malware classification with deep convolutional neural networks. In *2018 9th IFIP international conference on new technologies, mobility and security (NTMS)* (pp. 1-5). IEEE. [CrossRef]

[11] Vinayakumar, R., Alazab, M., Soman, K., & Others. (2019). Deep learning approach for intelligent intrusion detection system. *IEEE Access, 7*, 41525–41550. [CrossRef]

[12] Seneviratne, S., Shariffdeen, R., Rasnayaka, S., & Kasthuriarachchi, N. (2022). Self-supervised vision

transformers for malware detection. *IEEE Access, 10*, 103121-103135. [CrossRef]

[13] Szor, P. (2005). *The art of computer virus research and defense*. Pearson Education.

[14] Egele, M., Scholte, T., Kirda, E., & Kruegel, C. (2008). A survey on automated dynamic malware-analysis techniques and tools. *ACM computing surveys (CSUR), 44*(2), 1-42. [CrossRef]

[15] Saxe, J., & Berlin, K. (2015, October). Deep neural network based malware detection using two dimensional binary program features. In *2015 10th international conference on malicious and unwanted software (MALWARE)* (pp. 11-20). IEEE. [CrossRef]

[16] 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. [CrossRef]

[17] Gibert, D., Mateu, C., & Planes, J. (2020). The rise of machine learning for detection and classification of malware: Research developments, trends and challenges. *Journal of Network and Computer Applications, 153*, 102526. [CrossRef]

[18] Alshomrani, M., Albeshri, A., Alturki, B., Alallah, F. S., & Alsulami, A. A. (2024). Survey of Transformer-Based Malicious Software Detection Systems. Electronics, 13(23), 4677. [CrossRef]

[19] Benabdallah, I. (2021). Malimg original dataset [Data set]. Kaggle. Retrieved from https://www.kaggle.com/datasets/ikrambenabd/malimg-original (accessed: 19 December 2025).

[20] Anderson, K., & McGrew, D. (2015). Microsoft malware classification challenge (BIG 2015) [Data set]. Kaggle. Retrieved from https://www.kaggle.com/c/malware-classification (accessed: 19 December 2025).

[21] Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering, 22*(10), 1345–1359. [CrossRef]

[22] Bouchaib, P., & Bouhorma, M. (2021, April). Transfer learning and smote algorithm for image-based malware classification. In *Proceedings of the 4th International Conference on Networking, Information Systems & Security* (pp. 1-6). [CrossRef]

[23] Venkatraman, S., Alazab, M., & Vinayakumar, R. (2019). A hybrid deep learning image-based analysis for effective malware detection. *Journal of Information Security and Applications, 47*, 377-389. [CrossRef]

[24] Akritidis, L., Fevgas, A., Alamaniotis, M., & Bozanis, P. (2023, November). Conditional data synthesis with deep generative models for imbalanced dataset oversampling. In *2023 IEEE 35th international conference on tools with artificial intelligence (ICTAI)* (pp. 444-451). IEEE. [CrossRef]

[25] Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., ... & He, K. (2017). Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*.

[26] Meng, L., Li, H., Chen, B. C., Lan, S., Wu, Z., Jiang, Y. G., & Lim, S. N. (2022, June). AdaViT: Adaptive Vision Transformers for Efficient Image Recognition. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 12299-12308). IEEE. [CrossRef]

[27] Karat, G., Kannimoola, J. M., Nair, N., Vazhayil, A., VG, S., & Poornachandran, P. (2024). CNN-LSTM hybrid model for enhanced malware analysis and detection. *Procedia Computer Science, 233*, 492-503. [CrossRef]

[28] Ashawa, M., Owoh, N., Hosseinzadeh, S., & Osamor, J. (2024). Enhanced Image-Based Malware Classification Using Transformer-Based Convolutional Neural Networks (CNNs). *Electronics, 13*(20), 4081. [CrossRef]

[29] Raff, E., & Nicholas, C. (2017, November). Malware classification and class imbalance via stochastic hashed lzjd. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security* (pp. 111-120). [CrossRef]

[30] Moisejevs, I. (2019). Adversarial attacks and defenses in malware classification: A survey. *International Journal of Artificial Intelligence and Expert Systems, 8*.

[31] Kanadath, A., Jothi, J. A. A., & Urolagin, S. (2024). Cvits-net: A cnn-vit network with skip connections for histopathology image classification. *IEEE Access*. [CrossRef]

[32] Demetrio, L., Coull, S. E., Biggio, B., Lagorio, G., Armando, A., & Roli, F. (2021). Adversarial exemples: A survey and experimental evaluation of practical attacks on machine learning for windows malware detection. *ACM Transactions on Privacy and Security (TOPS), 24*(4), 1-31. [CrossRef]

[33] Cruickshank, I. J., & Carley, K. M. (2020). Analysis of malware communities using multi-modal features. *IEEE Access, 8*, 77435-77448. [CrossRef]

[34] Dilshad, M., Almas, B., Tariq, N., Jazri, H. B., Alwakid, G. N., Khan, J. S., ... & Kumar, R. (2024, July). IoV Cyber Defense: Advancing DDoS Attack Detection with Gini Index in Tree Models. In *2024 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC)* (pp. 1-8). IEEE. [CrossRef]

[35] Ramzan, T., Tariq, N., Farooq, U., Jazri, H. B., Ashraf, H., Khan, J. S., & Kumar, P. (2024, July). An ANN-Based Resampling Approach for Handling Imbalance and Overlapped Data. In *2024 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC)* (pp. 1-8). IEEE. [CrossRef]

[36] Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and psychological measurement, 20*(1), 37-46. [CrossRef]

[37] Matthews, B. W. (1975). Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure, 405*(2), 442-451. [CrossRef]

**Muhammad Masab** received the B.S. degree in computer science from HITEC University Taxila, Pakistan, in 2024. His specializes in artificial intelligence applications in agriculture and cybersecurity. In agriculture, he focuses on detecting and classifying crop diseases using deep learning. In cybersecurity, utilizing AI techniques for malware detection and classification to enhance digital security.
(Email: masabzafar5@gmail.com)

**Khubab Ahmad** received his B.S. degree in Electrical Engineering from HITEC University, Taxila, Pakistan, where his undergraduate project focused on COVID-19 detection using CT images and deep learning. He is currently pursuing his M.Sc. in Engineering at Multimedia University (MMU), Malaysia, under a Graduate Research Assistantship, with a research project on driver drowsiness detection using OBD-II sensor data and artificial intelligence. He has over 5 years of experience in artificial intelligence, and his research interests include deep learning, computer vision, large language models, federated learning, IoT security, and information security.
(Email: engr.khubab.ahmad@gmail.com)

**Muzammil Hussain** received the B.S. degree in computer science from HITEC University Taxila, Serial No.005390, Pakistan, in 2024. His research focuses on the application of artificial intelligence in smart farming and cybersecurity. He has experience in AI-driven data security, information security, and cybersecurity, including malware detection and threat classification.
(Email: muzammilhussainn01@gmail.com)

**Muhammad Shahbaz Khan** is an experienced academic with over 10 years of teaching and research experience. He earned his B.S. and M.S. degrees in Electronics and Electrical Engineering from NFC IET, Multan, and HITEC University, Taxila, in 2011 and 2015, respectively. He served as a Lecturer at HITEC University for more than 8 years and is currently pursuing a PhD in Cyber Security and Artificial Intelligence at Edinburgh Napier University, UK. His research interests include chaotic systems, image encryption, post-quantum cryptography, intrusion detection, and AI applications in cybersecurity and healthcare.
(Email: muhammadshahbaz.khan@napier.ac.uk)