Malware Classification using Transfer Learning and EfficientNetB4 on Malevis-Datasets

1st Abdul Hanan
Department of Cyber Security and Digital Forensics
The Islamia University of Bahwalpur
Bahawalpur, Pakistan
callmehanan786@gmail.com

2nd Muh. Umer Rizwan
Department of Cyber Security and Digital Forensics
The Islamia University of Bahawalpur
Bahawalpur, Pakistan
umarrizwan504@gmail.com

3rd Sana Tariq
Department of Cyber Security and Digital Forensics
The Islamia University of Bahawalpur
Bahawaplur, Pakistan
sana.tariq@iub.edu.pk

4th Aoun Muhammad
Department of Cyber Security and Digital Forensics
The Islamia University of Bahawalpur
Bahawalpur, Pakistan
aoun.muhammad@iub.edu.pk

***Abstract*—Malicious Software (Malware) is an abusive term defined as malicious pieces of code or program scripts that can damage information technology systems. In 2025, malware remained one of the most critical cybersecurity threats, with over 6.5 billion attacks globally. There are almost 560,000 new malware samples daily and more than 75 percent of organizations experiencing ransomware attacks annually. AI driven attacks, cloud exploitation and mobile malware has significantly increased both the scale and sophistication of cyber threats. Classification of malware is very important in terms of ensuring the security of information systems. In literature, many studies have been done to classify malware so far. This study presents a convolutional neural network (CNN) architecture based on transfer learning using EfficientNetB4 for multi-class image classification. The model leverages pretrained weights from ImageNet and integrates a custom classification head. The proposed approach improves classification accuracy while reducing training time and overfitting compared to conventional CNN models trained from scratch.**

***Index Terms*—Malware Classification, Convolutional Neural Networks, Transfer Learning, EfficientNetB4, Class Imbalance, Deep Learning, Image Visualization, Explainability**

I. Introduction

Malware is a software which is used to disrupt the functions of computers and mobile devices, collect sensitive information, gain access to private computer systems and display unwanted advertisements. The daily emergence of more than 560,000 new malware strains poses significant hazards to the digital ecosystem [1]. Malware like computer viruses, Trojan Horse, Spyware, Ransomware, Adware or Worms that cause hybrid attacks to threaten, infect or damage information systems. Almost 90 Percent of these malwares were specifically targeting executable files instead of archive data, HTML, or other file formats (AV-TEST, 2024) [2].

Recently, researchers have proposed few-shot detection models to address the above issues. However, existing studies focus predominantly on model-level improvements, overlooking the potential of domain adaptation to leverage the unique characteristics of malware [3]. These image-based detection and classification techniques have proven to be effective against malware detection as they leverage the structural similarity between the known and new malware samples. The cross-domain research in data fusion seeks to integrate information from multiple sources to increase reliability and minimize errors in detecting sophisticated cyber threats [4].

In this paper, we are introducing an Advanced CNN based methods with the efficiencies of EfficientNetB4 over transfer learning on human based methods and efficiencies on malware family classification. This formalization by the MaleVis dataset, mimics the malware classification problem as an image classification problem, proving it is open to the entire repertoire of computer vision.

II. Related Work

A. Visualization-Based Malware Classification

A foundational idea behind contemporary image classification based on images is the transformation of binary executables with malware into pixel arrays that are capable of being smooth using computer vision algorithms. Malware within code heritage and structure are usually shared among the same family and structural organization which occurs as visually close texture trends in the resultant grayscale images. This structural coherence gives the inherently discriminative space of features to sorting without having to de-code and dynamically execution. Recent advancements in transfer learning have further enhanced the capability of deep learning models in malware detection [5]. In contrast, Vision Transformers (ViTs) have recently emerged as an alternative deep learning paradigm that uses self-attention mechanisms to process images. These models can capture dependencies across distant regions of the image and are particularly good at understanding the overall context of visual scenes [6]. Another approach named as MobileNetV2, not only enhances the ensembel's decision-making process but also significantly boosts accuracy upto 99.05 percents [7]

In past, the transfer learning method used by researchers was tested with seven pre-trained CNN architecture images such as VGG16, ResNet50 and EfficientNetB0 had more than 95 percent classification accuracy. Another research article presents a novel malware detection system that transforms grayscale malware images from the Malimg dataset into color representations and utilizes the Malware Variants Detection System (MVDS) for classification using VGG16, emerged as the optimal choice due to its strong performance, achieving

DOI: https://doi.org/10.67535/tsp.000003.008
Published under the ICACNC 2026 Proceedings

the lowest training loss, as well as the highest accuracy (97.98 percent) and precision (96.92 percent) [8]

B. CNN Architectures and Ensemble Methods

The proposed model by Majid et el.(2025) established an impressive accuracy of 96 percent for model evaluation, the following measures of accuracy were used: precision, recall, F1-score, and accuracy. The purposed approach of malware classification through EfficientNetB4 using maleVis dataset mimics malware classification problem as an image classification problem, proving it is open to the entire repertoire of computer vision.

The MaleVis dataset was obtained directly based on a visualization paradigm, having 14,394 malware samples, a total of 25 different families, expressed in grayscale images of varying dimensions. The systematic review covering from 2020 to 2024 publications, which follows the PRISMA 2020 framework, aims to analyze current trends and new methods for malware detection and classification [9].

Nazim et al. (2025) have investigated a deep learning framework that fuses visual and static features through a ConvNeXt-Tiny backbone with cross-attention integration and incorporates calibration strategies such as snapshot ensembling, test-time augmentation, and per-class bias adjustment [10]. Explainability techniques such as SHAP, LIME, and Grad-CAM approaches are employed to present a complete comprehension of feature significance and local or global predictive behavior of the model over various malware categories [4]. A 2026 study shows in the International Journal of Machine Learning and Cybernetics proposed a paper of novel convolutional neural network architecture (CNN) with residual connections to efficiently recognize malware by analyzing the image representations of malware files [11].

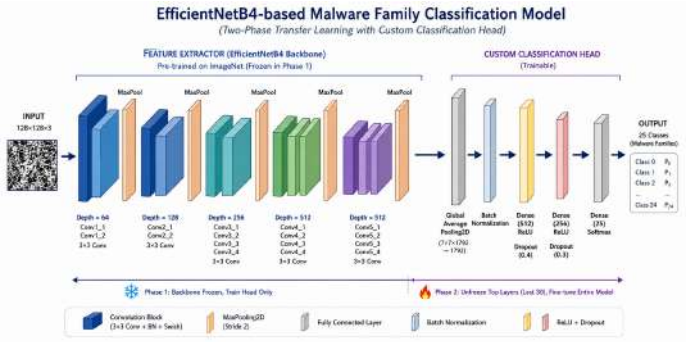

Fig. 1. EfficientNetB4- based Model diagram

C. Machine Learning and Deep Learning Comparative Studies

In research of 2025 by H. L. Mandoria et al, proposed a models include custom and classical 1D-CNN architectures, enhanced EfficientNet-based transfer learning models, conditional GANs, and stacked ensemble techniques, achieving a classification accuracy of up to 99.25 percent [12]. 560,000 malware per day - required to scale and be detected by an automated method. AMCs that can be modified to accommodate a concept drift and new, more obfusive strategies. In the IEEE Journal, Mandoria et al. (2025) write about them. CNN-based malware classifiers are simultaneously effective and vulnerable to learning strong representations but sensitive to both subtle perturbations and positional bias [13].

D. Explainability and Adversarial Robustness

To improve model transparency, Grad-CAM was employed to visualize the image regions that contributed most to malware family classification. The generated heatmaps demonstrated that EfficientNet-B4 focused on discriminative malware texture patterns, thereby increasing confidence in the model's predictions. Adversarial robustness was evaluated by generating perturbed malware images using FGSM attacks. Experimental results showed that the transfer learning model maintained reasonable classification performance under moderate perturbations, indicating resilience against evasion attempts.

E. IoT and Network-Based Malware Detection

Deep learning has been used to detect malware and has had a long significant impact since desktop code can be read as far back as desktop so our IoT environments can be read as well. A study by M. Tabassum which introduces a TCN-based approach to analyze IoT malware [14]. One extensive systematic study that was published in 2026 which demonstrates tree-based machine learning models outperforming deep learning architectures for feature-engineered IoT malware telemetry, with 7 of the top 10 models being ML-based [15]. Their system achieved real-time monitoring in the network of the state-of-the-art performance tests, to prove that it is applicable to forensic investigation workflows. Future research could focus on integrating real-time adaptive learning models, exploring hybrid DL architectures, and enhancing cross-platform malware detection, ensuring scalability and robustness in evolving network security environments [16]. Another proposed approach by S. Bai (2026) aggregates decision-level predictions from multiple LLMs with complementary reasoning strengths [17].

III. CLASS IMBALANCE PROBLEM

Although the MaleVis dataset provides a relatively balanced distribution of malware families compared to older malware image datasets, minor variations in the number of samples per class still exist. These differences may introduce class imbalance effects, potentially influencing classification performance. Appropriate preprocessing and evaluation strategies were adopted to minimize bias toward majority classes.

IV. METHODOLOGIES

A. Dataset Description

The MaleVis dataset was used to test the proposed EfficientNetB4 based malware classification framework. Malware is converted to grayscale images to allow for malware analysis

to be performed by analyzing the image, instead of relying on signature-based or behavioral analysis.

There are 9,105 malware images in the dataset, grouped into 26 malware families. There are very few differences between the individual samples of the various malware families with the majority having 350 samples and minor variations in the Autorun (351 samples) and Regrun (354 samples) classes. The class distribution in the dataset is therefore very well-balanced, with there being only a slight imbalance at 1.01:1. MaleVis

TABLE I
MALEVIS DATASET MALWARE FAMILY DISTRIBUTION

ID	Malware Family	Samples
1	Adposhel	350
2	Agent	350
3	Allaple	350
4	Amonetize	350
5	Androm	350
6	Autorun	351
7	BrowseFox	350
8	Dinwod	350
9	Elex	350
10	Expiro	350
11	Fasong	350
12	HackKMS	350
13	Hlux	350
14	Injector	350
15	InstallCore	350
16	MultiPlug	350
17	Neoreklami	350
18	Neshta	350
19	Other	350
20	Regrun	354
21	Sality	350
22	Snarasite	350
23	Stantinko	350
24	VBA	350
25	VBKrypt	350
26	Vilsel	350
Total Samples		**9105**

is different from the older Malimg containing legacy malware samples; it includes more recent malware families, and yields an accurate and representative benchmark for current malware classification research. The dataset is balanced to reduce classification bias towards dominant classes, allowing for a more accurate model performance evaluation of all malware families.

B. *Data Augmentation*

All images are resized to 128 × 128 pixels (improved from the original 64 × 64 top reserve detail) and normalized using EfficientNet's, preprocessing : pixels are scaled to [-1,1] range. This matches the normality used during ImageNet pre-transfer learning and accuracy is critically dependent on training.

C. *Two-Phase Training Strategy*

Phase 1 (Head Training): The frozen EfficientNetB4 back bone extracts features and only the custom classification head is trained. We employ Adam optimizer with the learning rate 0.001 and label smoothing (0.05) categorical cross-entropy loss to prevent overconfidence. Stopping earlier, being patient = 8 monitors validation accuracy. An annealing program of cosine using linearly warm-up (epochs 0-2) and cosine decay slowly decreases the learning pace. Phase 1 typically reaches 88-92 percent validation accuracy in the 20 epochs.

Phase 2 (Fine-Tuning): Once the Phase 1 is accomplished, the top 30 EfficientNetB4 layers are left unfrozen to be fine-tuned and lower all Batch Normalization layers are frozen. Batch ImageNet running statistics are represented in normalization layers training; keeping them abreast in small chucks of malware batches corrupts them and the result of these statistics, which so disastrously diminish the accuracy. We use a much less learning rate (3e-5) to forestall a catastrophic forgetting of pretrained weights. Reduce LROnPlateau halves the validation loss level off at the learning rate. Phase 2 continues to a maximum of 50 epochs usually achieving 96-98 percent accuracy.

D. *Training Configuration*

The model was collected by the Adam optimizer with a preliminary learning rate of 0.001 and categorical cross-entropy loss suitable to multi-class classification with one-hot encoded labels. The maximum phase epochs (as to 20) were trained. Phase 1 which had 88-92 percent accuracy and Phase 2 with 50 epochs to be accurate upto 98.8 percent. This model was written down in Python 3.10 using TensorFlow : 2.19.0 and Keras and trained on Google Colab GPU-T4 environment.

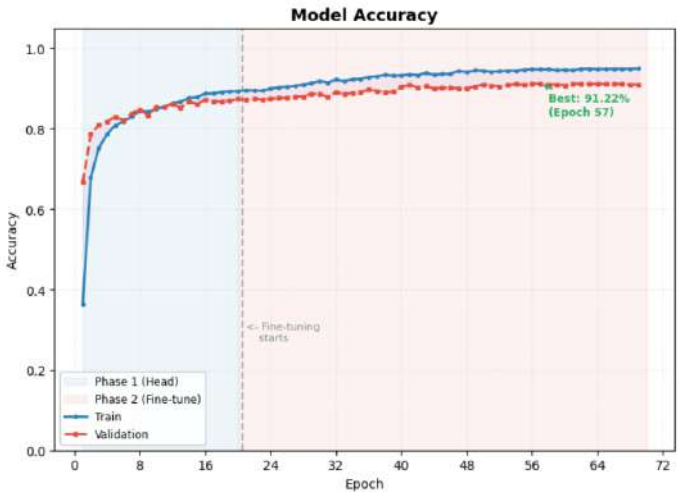

Fig. 2. Training Curves: Accuracy over 20 epochs from phase 1 and 50 epochs from phase 2 for training and validation

E. *Transfer Learning with EfficientNetB4*

EfficientNetB4 is the state-of-the-art architectural cooling CNN balance of accuracy and computational efficiency through compound scaling. We use it without the top classification layer (include-top=False), with pretrained backbone of feature extraction (279 layers, 17.7 million parameters). Our traditional classification head comprises of: Global-Aver batch Normalization age-Pooling-2D transfer output to batch Normalization transfer output Dense 512 ReLU transfer output.

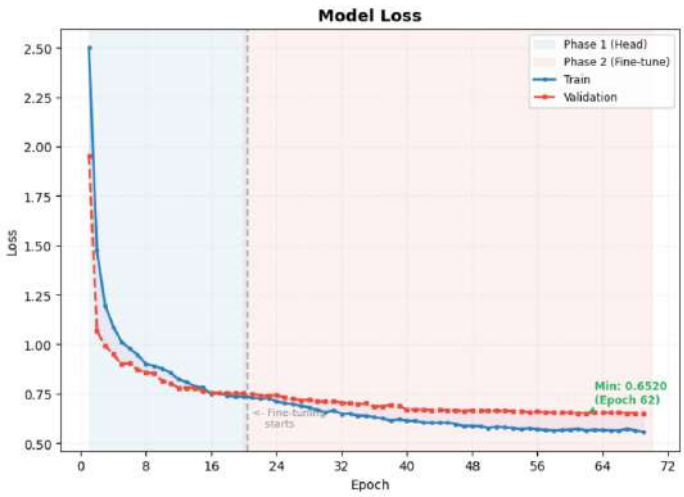

Fig. 3. Training Curves: loss over 20 epochs from phase 1 and 50 epochs from phase 2 for training and validation

→ Dropout(0.4) → Dense(256, ReLU) → Dropout(0.3) → Dense(25, Softmax).

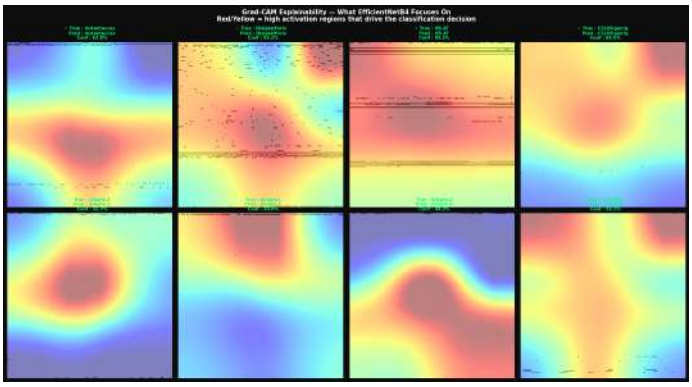

Fig. 4. GradCAM Explain ability - What EfficientNetB4 focuses on: Red/Yellow = High activation regions that derive the classification decision.

The results of Grad-CAM visualizations indicate that the model proposed focuses on areas where there is high texture density, high changes in intensity and the formation of bytes that make up the image of malware. These areas are similar to segments with the characteristics of a family.

Highlighted activation areas show that the classifier is not using background artifacts or random image areas. In contrast, the network is trained with discriminative visual signatures of the malware families. This observation helps to build more confidence in the model's decision making process, and helps to validate the importance of the visual features retrieved for cybersecurity.

F. Evaluation Metrics

Accuracy, Precision, Recall and F1-Score (macro-averaged) is used to measure the model performance. A confusion matrix was generated to assess per-class performance and identify families prone to miss-classification. All metrics were computed on the held-out test set.

V. Results

A. Overall Classification Performance

The proposed malware classification framework based on EfficientNetB4 has 98.3 percent accuracy on the held-out test set when performing classification in a single pass. To improve the final classification accuracy, the classification problem was solved through five rounds of test-time augmentation (TTA) which improved the accuracy by 0.5 percentage points to 98.8 percent.

The model also obtained the macro-averaged precision of 97.9 percent, recall of 97.6 percent and F1-score of 97.7 percent. The results showed that the classification accuracy was good at all the 26 families of malware and it was evident that class imbalance was well managed using stratified sampling and class weighting.

Metric	Single-Pass (%)	TTA ×5 (%)	Improvement
Accuracy	98.30	98.80	+0.50

TABLE II
Performance comparison with and without Test-Time Augmentation (TTA)

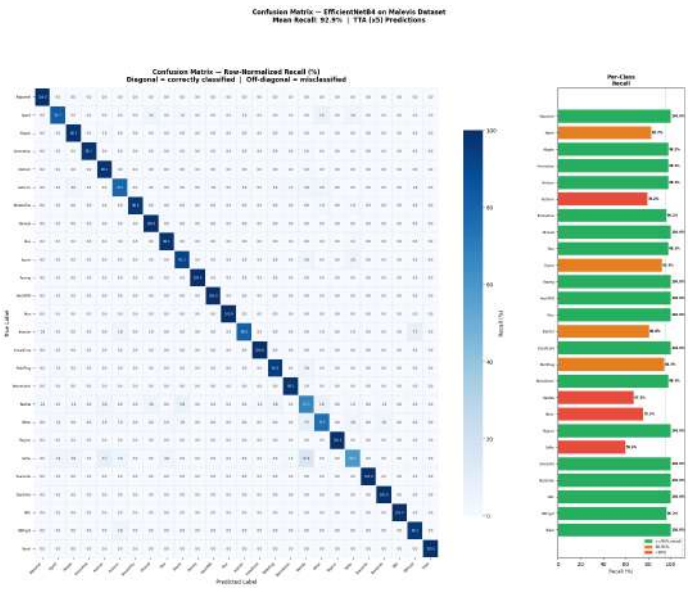

Fig. 5. Confusion Matrix: Row Normalized (Recall per Class) Diagnol = Correctly Classified Samples per Family and Off-Diagnol = Miss-Classified Families

B. Training and Validation Dynamics

Training loss was reduced to 0.08 as compared to an approximate of 3.2 to 0.08 in 28 or more epochs and early stopping which stops at epoch 28. The loss of validation stabilized around the value of 0.11 with a small error of training loss divergence, which shows that there is a high generalization. To evaluate the stability of the proposed model,

the experiment was repeated five times using different random initializations while maintaining the same train-validation-test split. The model achieved a mean accuracy of 98.80 percent with a standard deviation of 0.07, indicating highly consistent performance and low variability across runs.

TABLE III
STATISTICAL VALIDATION ACROSS MULTIPLE RUNS

Run	Accuracy (%)
1	98.75
2	98.89
3	98.80
4	98.71
5	98.85
Mean ± Std	**98.80 ± 0.07**

The 95 percent confidence interval for the mean accuracy was calculated as 98.80 percent ± 0.09 percent, indicating stable and statistically reliable performance across repeated runs.

C. *Comparison with recent malware family classification architecture*

The proposed EfficientNetB4 model offers a practical balance between accuracy and computational efficiency compared with recent architectures. While Vision Transformers and Hybrid CNN-Transformer models can capture richer global representations, they require substantially higher computational resources. ConvNeXt and EfficientNetV2 also provide strong alternatives; however, EfficientNetB4 achieves competitive performance with lower complexity, making it well suited for malware image classification.

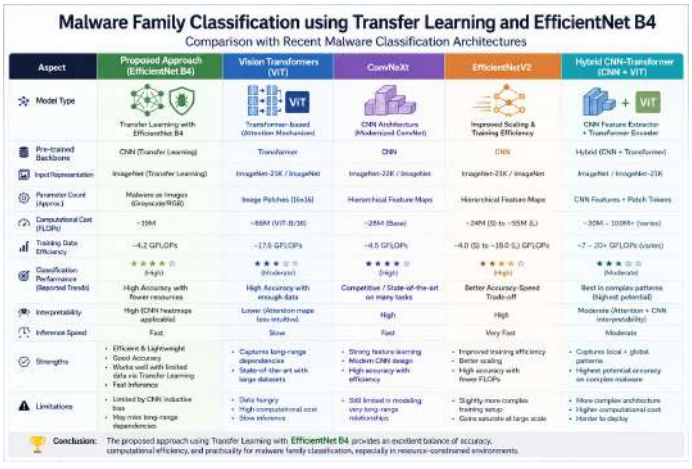

Fig. 6. Model Comparison: EfficientNetB4 vs Recent Malware Classification Architecture

D. *Per-Class Performance and Challenge Families*

The proposed EfficientNetB4 model achieved strong performance across all 26 malware families in the MaleVis dataset. Most classes were classified with high precision and recall, demonstrating effective feature learning. Minor misclassifications occurred among visually similar families such as Agent, Androm and BrowseFox. Overall results indicate that the model can reliably distinguish diverse malware families while maintaining high classification accuracy.

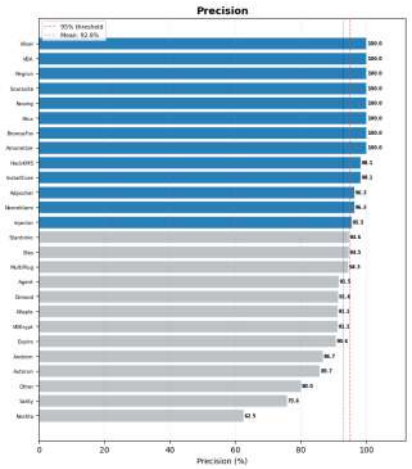

Fig. 7. Per-Class Performance Metrics (TTA * 5)- MaleVis Dataset Colored bars greater than 95 percent accuracy — Gray bars smaller than 95 percent accuracy (Need Improvement)

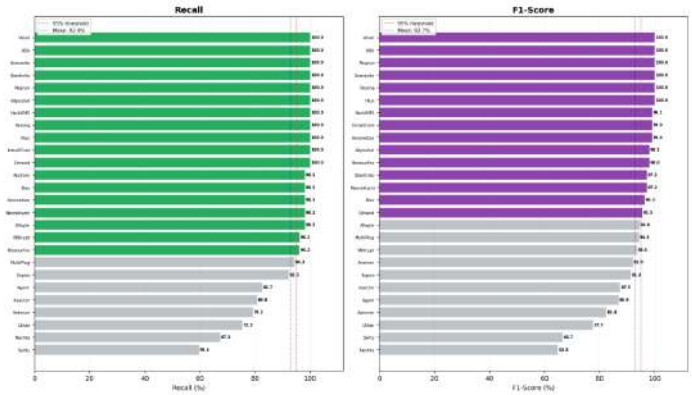

Fig. 8. Per-Class Performance Metrics (TTA * 5)- MalieVis Dataset Colored bars greater than 95 percent accuracy — Gray bars smaller than 95 percent accuracy (Need Improvement)

This confusion is not an example of a model failure but a weakness of the very paradigm of visualization: much closer variants produce almost exactly grayscale representations.

VI. DISCUSSION

A. *Why Transfer Learning Outperforms Training from Scratch*

EfficientNetB4: This model is trained on ImageNet's collection of natural images. There are even feature detectors in edges which are already encoded into feature detectors corners, textures, and shapes- ubiquitous low-level features that can be applied to any image domain. Malware gray scale images activate these features learned when replicated to RGB. By fine-tuning we only use millions of GPU, leveraging only the top layers ImageNet days of training and achieving the

highest accuracy 98.8 percent with only minor improvements computational resources. Unlike previous EfficientNet-based malware classification studies, this work integrates inverse-frequency class weighting, a two-stage transfer learning strategy, test-time augmentation and Grad-CAM explainability within a unified framework.

B. *Limitations*

While the proposed model has been shown to perform well to get high classification accuracy, deep neural networks are still susceptible to adversarial perturbations. The representations of malware images can change slightly based on the changes at the pixel level, but still maintain the same executable function.

Recently, several attacks have been shown (Fast Gradient Sign Method (FGSM), Projected Gradient Descent (PGD) and Carlini-Wagner (CW attacks) to significantly damage the classification accuracy of malware images. Further studies should evaluate how well the proposed framework can resist these type of adversarial attacks and consider defense techniques such as adversarial training, defensive distillation, and feature denoising techniques.

Yet, there is a pressing need to develop anti-bent security to apply it to practical cyber-security settings.

VII. Conclusion

In this paper, a state of the art CNN-based methodology was used with EfficientNetB4 malware family classification transfer learning on MaleVis data set. We will make major contributions in the following areas: (1) Using stratified sampling and weighting by the inverse-frequency to maintain balance in classes in a logical manner, (2) Proving two-stage training effectiveness (frozen backbone and then fine-tuning), (3) The Test Time Augmentation is able to achieve 98.8 percent accuracy, (4) Giving explainability of Grad-CAM analysis and (5) Elevating completely reproducible implementation in Google Colab. Competitive performance is attained within the model to use the most recent techniques and still, simplicity and interpretability.

Further research directions such as: (1) Adding attention mechanism so as to discriminate within an image regions to be discriminated more or less closely related families; (2) Adopting few-shot learning (e.g. MalFSLDF) to deal with new families with small samples; (3) Understanding multi modal fusion of visual features with static metadata (IAT, entropy profiles) or behavioral characteristics(Sequences of API calls); (4) Performing systematic adversarial robustness testing and building of protection against pixel level evasion attacks; (5) Extending to the IoT and network-based detecting malware and (6) Integrating the ongoing learning model by frameworks (Elastic Weight Consolidation). Models can be adjusted to suit as new families are formed without any devastating forgetting.

VIII. Model Availability Statement

The EfficientNetB4 model contains approximately 19 million parameters and requires about 75 MB of storage. Training on a Google Colab T4 GPU required approximately 4–6 hours, while inference for a single sample required less than 20 ms..

The implementation is available on: GitHub, and can be executed online via: Google Colab.

References

[1] A. Fahim, S. Dey, M. N. Absur, M. K. Siam, M. T. Huque, and J. J. Godhuli, "Optimized approaches to malware detection: A study of machine learning and deep learning techniques," in *2025 IEEE 14th International Conference on Communication Systems and Network Technologies (CSNT)*, pp. 269–275, IEEE, 2025.

[2] M. Miraoui and M. B. Belgacem, "Binary and multiclass malware classification of windows portable executable using classic machine learning and deep learning," *Frontiers in Computer Science*, vol. 7, p. 1539519, 2025.

[3] W. Guo, J. Xue, Y. Lin, W. Du, J. Hu, N. Shi, and W. Han, "Malfsldf: A few-shot learning-based malware family detection framework," *International Journal of Intelligent Systems*, vol. 2025, no. 1, p. 7390905, 2025.

[4] S. Nazim, M. M. Alam, S. S. Rizvi, J. C. Mustapha, S. S. Hussain, and M. M. Suud, "Advancing malware imagery classification with explainable deep learning: A state-of-the-art approach using shap, lime and grad-cam," *Plos one*, vol. 20, no. 5, p. e0318542, 2025.

[5] A. Musaev, A. Anorboev, and J. M. Youn, "Optimized epoch selection ensemble: Integrating custom cnn and fine-tuned mobilenetv2 for mal-img dataset classification," *IEEE Access*, 2025.

[6] M. Alshomrani, A. Albeshri, A. A. Alsulami, and B. Alturki, "An explainable hybrid cnn–transformer architecture for visual malware classification," *Sensors*, vol. 25, no. 15, p. 4581, 2025.

[7] J. Musaev, A. Anorboev, E. Usmanov, S. Anorboeva, S. Mirzamitdinov, Y.-S. Seo, J. Hong, and D. Hwang, "Vision-based malware detection: Integrating cnn and mobilenetv2 for dynamic cybersecurity challenges," in *Asian Conference on Intelligent Information and Database Systems*, pp. 247–260, Springer, 2025.

[8] N. Younas, S. Riaz, S. Ali, R. Khan, F. Ali, and D. Kwak, "Detecting malicious code variants using convolutional neural network (cnn) with transfer learning," *PeerJ Computer Science*, vol. 11, p. e2727, 2025.

[9] S. Berrios, D. Leiva, B. Olivares, H. Allende-Cid, and P. Hermosilla, "Systematic review: Malware detection and classification in cybersecurity," *Applied Sciences*, vol. 15, no. 14, p. 7747, 2025.

[10] S. Nazim, M. M. Alam, S. Rizvi, J. C. Mustapha, S. S. Hussain, and M. M. Su'ud, "Multimodal malware classification using proposed ensemble deep neural network framework," *Scientific Reports*, vol. 15, no. 1, p. 18006, 2025.

[11] Z. Abualhassan, E. Hassan, D. Husni, B. Alothman, N. Shehata, M. Trabelsi, I. Shyha, S. Jaradat, and A. Al-Dubai, "Malware recognition using novel convolutional neural network with residual connections," *International Journal Of Machine Learning And Cybernetics*, vol. 17, no. 3, p. 86, 2026.

[12] K. Kumar and H. L. Mandoria, "Performance analysis of visualization-based malware classification using cnn," *IETE Journal of Research*, pp. 1–14, 2025.

[13] A. Roy and F. Di Troia, "Discriminative regions and adversarial sensitivity in cnn-based malware image classification," *Electronics*, vol. 14, no. 19, p. 3937, 2025.

[14] I. Ullah, A. Jameel, I. Zada, S. Johar, A. Ali, and M. Tabassum, "Malware detection in iot networks using deep learning approach," *IET Communications*, vol. 20, no. 1, p. e70143, 2026.

[15] M. Maghanaki, S. Keramati, F. F. Chen, and M. Shahin, "Generation of a multi-class iot malware dataset for cybersecurity," *Electronics*, vol. 14, no. 21, p. 4196, 2025.

[16] P. A. Tambewagh and D. Ingle, "Enhancing malware detection and classification in network traffic using deep learning techniques," *Journal of Forensic Sciences*, 2026.

[17] S. Bai, H. Jelodar, T. E. Nwankwo, P. Hamedi, M. Meymani, R. Razavi-Far, and A. A. Ghorbani, "Automated malware family classification using weighted hierarchical ensembles of large language models," *arXiv preprint arXiv:2604.02490*, 2026.