Enhancing Malware Detection with Advanced Deep Learning Models

Sayam Dhingra
Department of Computer Engineering
New York University
sd5292@nyu.edu

Ankita Gupta
Department of Computer Engineering
New York University
ag9135@nyu.edu

Abstract

In response to the increasing complexity and obfuscation of modern malware, our research focuses on a cutting-edge method that employs deep learning models, specifically trained to detect and classify malware represented as RGB images. This innovative technique capitalizes on recent progress in deep learning for image classification, offering a significant enhancement in malware detection through the use of color imagery. Our methodology encompasses the adaptation of state-of-the-art Convolutional Neural Network (CNN) architectures, the incorporation of additional layers, and the application of regularization strategies. These modifications are specifically tailored to effectively identify distinctive characteristics of malware and ordinary programs, providing a more robust and accurate detection framework in the evolving landscape of cybersecurity threats. This is our code

1 Introduction

Cybersecurity is an ever-evolving battlefield where defenders and attackers continuously adapt to each other's strategies. Malware, a pervasive threat in this landscape, poses a significant challenge due to its evolving complexity and the ability to evade traditional detection systems. Recently, machine learning, particularly Convolutional Neural Networks (CNNs), has emerged as a powerful tool to counteract these threats by identifying patterns indicative of malware in data representations.

In this project, we explore the novel application of CNNs to malware detection, using RGB images as a data representation method. The RGB images were obtained from the MaleVis dataset (4) . This approach is inspired by the ability of CNNs to effectively capture and learn from complex patterns in image data. Our work is built upon the premise that malware, when visualized as RGB images, may exhibit unique textural and structural patterns that can be leveraged for more accurate classification (1). This stands in contrast to conventional methods that typically utilize binary or grayscale image representations.

Our research contributes novel applications of the latest deep learning image classification models on a less explored dataset MaleVis (4). We provide a comprehensive report on comparing different models, specifically EfficientNetB4, InceptionResNetV2, DenseNet169 and ResNetV2 trained on the MaleVis dataset with all 26 classes where 25 are malware and the last class is not malware. Through a comprehensive set of experiments, we evaluate the models' performance on several fronts, utilizing standard metrics such as accuracy, precision, recall, and AUC. Additionally, we assess the models' generalizability to ensure their robustness and reliability in practical scenarios.

NeurIPS 2023 Generative AI and Biology Workshop.

2 Literature Review and Related Work

Our project explores the efficacy of Convolutional Neural Networks (CNNs) in malware detection, specifically using RGB images. This approach, building on the work of Nataraj et al. (2011) (2) and Bensaoud et al. (2020) (1), seeks to expand beyond traditional binary and grayscale methods. Kulshrestha (2020) (3) further illustrated the power of deep learning in this domain. Our methodology differs from Jayasudha et al. (7), who excluded non-malware data, as we aim to distinguish between malware and non-malware.

While significant strides have been made in applying CNNs to malware classification, the full potential of RGB imagery remains underexplored. Our research will critically assess the impact of color image representations on malware detection model performance, focusing on model generalization and reliability. We aim to contribute to the field by providing insights into the effectiveness of CNN architectures and preprocessing techniques for identifying non-malware programs, thereby advancing the application of machine learning in cybersecurity.

3 Methodology

Our project employs several sophisticated Convolutional Neural Network (CNN) architectures to address the challenge of malware classification using RGB images. The chosen models are Inception v3, DenseNet, ResNet-50, and Inception-ResNet v2. These architectures are renowned for their high performance in image classification tasks and are selected for their unique characteristics that are hypothesized to be beneficial for malware image analysis.

We could not use other image based deep learning models like MAE (Masked Autoencoders) and DINO (Self-Supervised Learning of Pretrained Transformers) [1] because the MaleVis dataset requires classification output. Masked Autoencoders on the other hand work by masking a portion of the input data (for example, parts of an image) and then training a model to predict the masked parts. This type of learning could cause the model to predict non-malware programs as malware and introduce noise into the model. As the dataset comprises of binaries converted to images, this approach may not be effective in classifying malware from non-malware.

Similarly, DINO is a framework for training vision transformers (ViT) using self-supervision. It does not rely on labeled data. Instead, it generates its own labels in the process of learning, typically by creating different views of the same image and ensuring that the transformer's outputs for these views are similar. This would defeat the purpose of being able to classify a program between the given 25 malware classes and one non-malware class.

3.1 Dataset

The **MaleVis dataset** consists of *RGB byte images* divided into 26 classes, with 25 representing various malware types and 1 for 'clean' or legitimate software. This composition is designed to facilitate multi-class malware recognition studies.

Images in the dataset are created by converting malware and legitimate software binary files into 3-channel RGB format, using a script developed by Sultanik. These images are then standardized to resolutions of either 224x224 or 300x300 pixels. [2]

The dataset includes a total of **14,226 RGB images**, with **9,100** for training and **5,126** for validation. Each malware class in the training set contains **350 image samples**. The validation set features a larger number of 'legitimate' samples (**1,482**) compared to each malware class (**350**), reflecting the nature of malware detection and recognition. We further split the test dataset into 80-20 split for further test train data. We did not test the affect of different ratios of split. We also did not use any form of image augmentation as this would only create noise and hinder the training of the models. Since the images are derived from malware binaries we need every pixel to be unedited as it represents binary code of a malware portable executable. Augmenting the image would alter the malware binary. [3]

[1] Why not DINO and MAE

[2] How are the images related to malwares

[3] Augmentation question

A DenseNet-based convolutional neural network achieved a **state-of-the-art accuracy of 97.48%** on the MaleVis validation set in a closed-set scenario (excluding legitimate samples). Open-set scenario results (including legitimate samples) are **pending publication**.

3.2 Models

3.2.1 InceptionResNetV2

InceptionResNetV2, blending Inception's efficiency and ResNet's deep learning capabilities, excels in feature extraction for complex patterns in the MaleVis dataset's RGB images. Its resistance to overfitting, along with proficiency in handling varied image sizes like 224x224 and 300x300 pixels, makes it ideal for MaleVis. Additionally, its architecture effectively manages class imbalance, enhancing reliability in malware classification.

3.2.2 DenseNet

DenseNet, characterized by its densely connected layers, offers efficient feature reuse crucial for identifying intricate patterns in malware images. This model's computational efficiency, combined with its robustness against overfitting, suits the MaleVis dataset's requirements. Its architecture supports detailed feature mapping and transfer learning, promising high accuracy in classifying various malware types.

3.2.3 ResNet-50

ResNet-50 utilizes deep residual learning and skip connections, effectively training deep networks and extracting complex features from the MaleVis dataset. Suitable for learning from the dataset's structure, including 350 samples per malware class, ResNet-50's track record in image classification and support for transfer learning make it a strong candidate for the MaleVis dataset.

3.2.4 EfficientNet

EfficientNet's systematic scalability in network dimensions (depth, width, resolution) aligns well with the MaleVis dataset's requirements. Its ability to efficiently learn complex features relevant to malware classification, combined with high accuracy in image classification, positions EfficientNet as an effective model for training with the MaleVis dataset.

3.2.5 InceptionV3

InceptionV3, a refinement of the original Inception model, is designed to balance the network's depth and width while maintaining computational efficiency. This model is known for its innovative use of convolutional operations and network-in-network architecture, which increases the depth and width of the network without a significant increase in computational cost.

The selection of these models is grounded in the hypothesis that the feature extraction capabilities of CNNs can be effectively translated to malware classification. Given that malware binaries manifest as complex, textured patterns when visualized as images, the diverse and robust architectures of these CNNs are expected to excel in capturing the essential features for accurate classification. The inclusion of multiple architectures also allows for a comparative analysis to determine the most effective model for this specific application.

Each model's performance is rigorously evaluated using a suite of metrics, including accuracy, precision, recall, F1-score, and AUC-ROC curves. Through these metrics, we aim to validate the hypothesis that RGB image representations, when processed through these sophisticated CNNs, can lead to significant improvements in malware detection capabilities.

4 Data Prepossessing

4.1 Data Cleaning and Transformation

Our dataset preprocessing starts by converting malware binaries into RGB images, a technique initially applied in the MaleVis dataset. This method interprets binary data as RGB pixel values to reveal complex patterns. This step was already done by the creators of the MaleVis dataset (4) (8) We chose dataset-specific normalization over the standard ImageNet normalization to improve model performance. This involves using the mean and standard deviation specific to our dataset, ensuring a better fit for its unique characteristics.

Contrastingly, ImageNet normalization uses fixed values from the ImageNet dataset, suitable for similar datasets but potentially less effective for ours with distinct features.

In conclusion, dataset-specific normalization is advantageous for datasets vastly different from ImageNet, enhancing model accuracy and performance, while ImageNet normalization is better suited for similar datasets or quick prototyping.[4]

4.2 Model Architecture and Fine-Tuning

Our chosen architectures—Inception v3, DenseNet, ResNet-50, and Inception-ResNet v2—are fine-tuned for our specific classification task. Fine-tuning involves adjusting the pre-trained models, which have been initially trained on the vast ImageNet dataset, to our malware classification context. We add custom layers tailored to the nuances of our dataset, including fully connected layers and regularization mechanisms like dropout to mitigate overfitting.

4.3 Training and Validation

The training process is carefully monitored using validation sets to gauge the models' performance and to implement early stopping, ensuring that training ceases at the optimal moment to avoid overfitting while also ensuring model convergence.

Evaluation Metrics and Model Performance: We adopt a comprehensive suite of evaluation metrics to thoroughly assess our models. Accuracy, AUC, precision, recall, and F1-scores are calculated to provide a holistic view of each model's performance. These metrics are particularly important in the imbalanced domain of malware datasets, where certain classes of malware may be more prevalent than others.

4.4 Impact of Image Representation on Model Performance

The choice of image representation plays a crucial role in the performance of machine learning models, especially in the context of malware classification. In our study, we have chosen to represent malware binaries as RGB images, moving beyond the conventional grayscale or binary representations. The rationale behind this decision is rooted in the rich information that RGB images can encode. [5]

RGB images, with their three color channels, have the potential to capture intricate patterns and subtle variations in the data that grayscale images, with a single channel, might miss. The color depth in RGB images could correspond to different aspects of the binary data, such as the opcodes distribution, control flow graph, or byte-level features, which are often crucial in distinguishing between malicious and benign software.

4.5 Comparative Analysis

A critical aspect of our study involves comparing the performance of various state-of-the-art CNN architectures to ascertain their efficacy in malware classification when utilizing RGB image representations. The architectures under scrutiny include Inception v3, DenseNet, ResNet-50, and Inception-ResNet v2, each presenting unique structural benefits that could influence their ability to process and classify malware images.

[4] types of normalization

[5] Why RGB

The comparative analysis was conducted under uniform experimental conditions to ensure fairness and reliability of the results. Each model was trained on the same dataset, preprocessed using CNN model specific conversion techniques. The models were then evaluated based on a range of metrics, including accuracy, precision, recall, and F1-score, to gauge their classification performance comprehensively.

5 Experimental Results

5.1 Performance Metrics

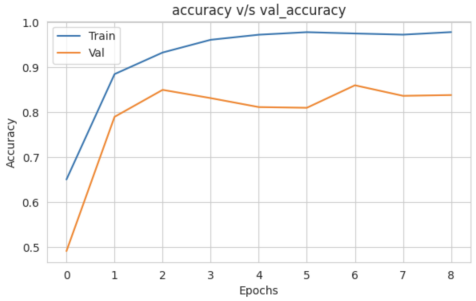
We employ a comprehensive set of performance metrics to evaluate the effectiveness of our proposed model in anomaly detection. The key metrics include classification accuracy, precision, recall, F1 score, and the area under the receiver operating characteristics (ROC) curve.

- Classification Metrics: True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN)
- True Positive Rate (Recall): Measures the ratio of correctly predicted samples to the overall number of instances of the same class. **Formula:** $TPR = \frac{TP}{TP+FN}$
- Precision: Ratio of correctly predicted samples to the total number of predicted samples for a class. **Formula:** $Precision = \frac{TP}{TP+FP}$
- F1-Score: F1-Score represents the harmonic mean of precision and recall, providing a balanced measure of a model's performance. **Formula:** $F1 = 2 \times \frac{Precision \times Recall}{Precision+Recall}$
- Accuracy (Acc): Measures the total number of data samples correctly classified across all predictions. **Formula:** $Acc = \frac{TP+TN}{TP+TN+FP+FN}$

5.2 Training history

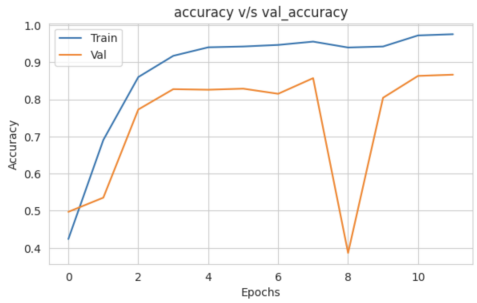
This section presents the training progress of our CNN models, showcasing their performance metrics' evolution over epochs. We visualize loss and accuracy trends to illustrate model optimization and generalization capabilities. These insights are vital for understanding Inception v3, DenseNet, ResNet-50, and Inception-ResNet v2 behaviors, and validating our malware detection approach. Note that the gap between validation and training accuracy is intentional, reflecting dataset skew for more realistic training and improved test accuracy, as discussed in the next section.

5.2.1 InceptionResNetV2



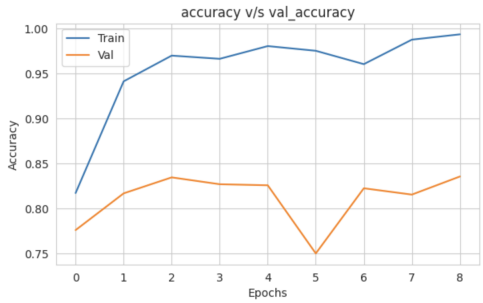
The training accuracy starts at around 60% and improves significantly over the first few epochs, reaching close to 100%. The validation accuracy starts around the same point but increases at a slower rate, peaking at around 80%.

5.2.2 DenseNet



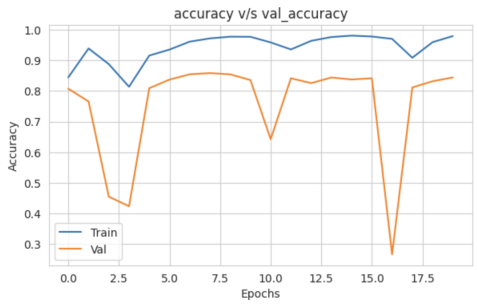
It begins at around 40% and approaches, but doesn't quite reach, 100%. This is a good sign of the model's capacity to fit the training data. We also observe a sharp dip and subsequent recovery in validation accuracy could indicate an anomaly. This could be due to a random fluctuation due to the stochastic nature of the training process, especially if the validation set is small. Despite the anomaly, the model appears to recover, which could suggest it's robust to certain types of perturbations or errors.

5.2.3 ResNet-50



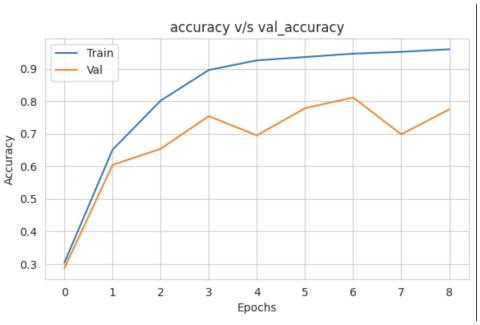
The training accuracy (blue line) quickly climbs to a high level, maintaining above 95% after the initial epochs. This shows the model has a good fit on the training dataset. The validation accuracy recovers after the dip, which could mean the model is still able to learn from the data. Despite the issues, the model does achieve relatively high validation accuracy.

5.2.4 EfficientNet



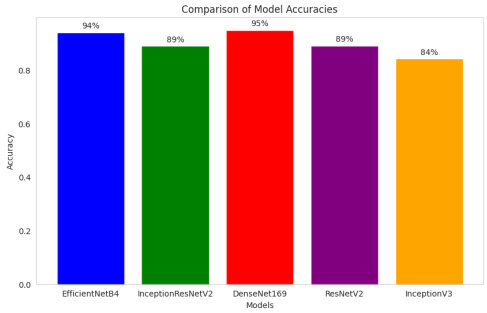
The validation accuracy (orange line) exhibits significant volatility throughout the training process. Notably, there are sharp declines at epochs 5, 10, and around 15, with recoveries following each drop. The training accuracy (blue line), in contrast, remains relatively stable throughout the epochs, maintaining a high accuracy above 90%. But ultimately the model performs well as the end validation acc is high.

5.2.5 InceptionV3

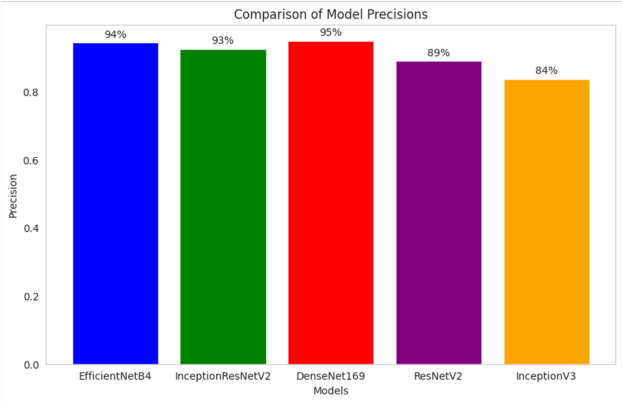


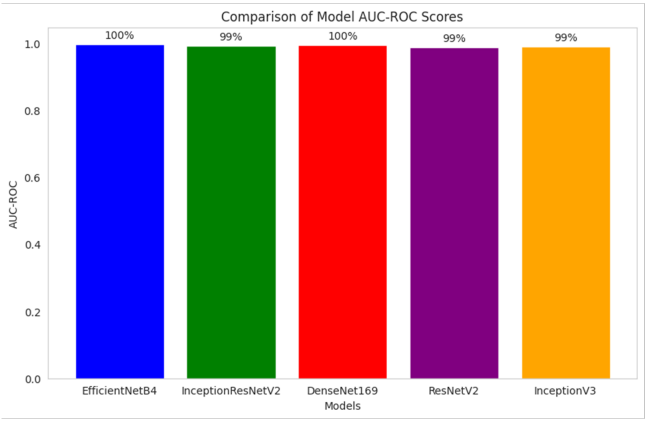
The blue curve shows that the training accuracy steadily increases with each epoch. This indicates that the model is learning and improving its performance on the training data. It appears to reach a peak and then fluctuates. This behavior suggests that the model's performance on the validation data is less stable than its performance on the training data.

5.3 Performance Results

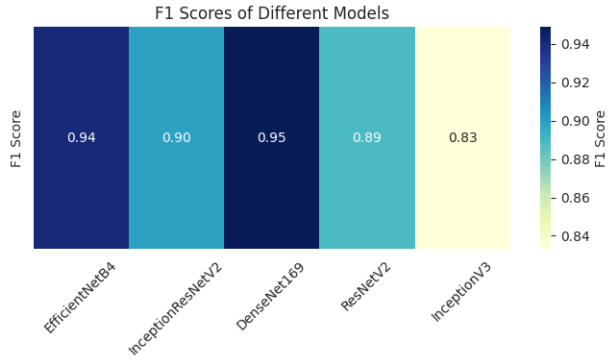


DenseNet169 leads with 95% accuracy, followed by EfficientNetB4 at 94%, with InceptionResNetV2 and ResNetV2 both at 89%, and InceptionV3 at 84%.





DenseNet169 emerges as the most precise with a 95% rating, while InceptionV3 lags behind with 84%. The second chart measures the AUC-ROC scores, where both EfficientNetB4 and DenseNet169 achieve perfect scores of 100%, indicating excellent performance in distinguishing between classes. In contrast, InceptionV3, despite its lower precision, still maintains a high AUC-ROC score of 99%, demonstrating its strong classification capabilities despite being less precise.



DenseNet169 stands out with the highest F1 score of 0.95, indicating it has the best balance of precision and recall among the models shown. EfficientNetB4 also performs strongly with an F1 score of 0.94. On the other end of the spectrum, InceptionV3 has the lowest F1 score at 0.83, which could suggest a comparative deficiency in either precision, recall, or both.

5.4 Graphs

Model	Accuracy	F1 Score	AUC	Precision
EfficientNetB4	0.9401098901098901	0.940943009215053	0.9980348508634223	0.9436976568074498
InceptionResNetV2	0.8895604395604395	0.8994352198920692	0.9937108320251178	0.9256048116130862
DenseNet169	0.9494505494505494	0.9488244290441897	0.9960731554160127	0.9496094013825357
ResNetV2	0.8906593406593407	0.8884769699697543	0.9893940345368917	0.8897436648513403
InceptionV3	0.8428571428571429	0.8330573557659939	0.9916640502354788	0.8370144282178944

The table presents performance metrics for five machine learning models, with DenseNet169 exhibiting the highest F1 score and EfficientNetB4 showing high precision. InceptionV3 has the lowest values across all metrics, while InceptionResNetV2 and ResNetV2 display competitive, middle-range scores. This concise comparison reveals each model's strengths and weaknesses across accuracy, F1 score, AUC, and precision.

6 Conclusion

This research represents a significant step forward in the application of deep learning to malware detection, particularly through the use of RGB images. Our exploration of various state-of-the-art Convolutional Neural Networks (CNNs) including DenseNet169, EfficientNetB4, InceptionResNetV2, ResNetV2, and InceptionV3 has yielded promising results. DenseNet169, in particular, demonstrated exceptional performance, leading in accuracy, precision, and F1-score. The use of RGB images in malware classification has shown to be effective, with these CNN architectures capable of capturing complex patterns unique to malware.

The experimental results highlight the strengths of each model in different aspects of malware detection. DenseNet169's leading performance across multiple metrics, including a perfect AUC-ROC score, indicates its robustness and reliability in this domain. EfficientNetB4's performance is also noteworthy, especially in terms of precision and F1-score. Future research should focus on several key areas to enhance the effectiveness of malware detection using deep learning. Investigating other neural network architectures, including newer or less conventional ones, might yield better results or more efficient training processes. Testing these models in real-world scenarios, against the latest and evolving malware threats, would provide valuable insights into their practical effectiveness and areas for improvement.

References

[1] Ahmed Bensaoud, Nawaf Abudawaood, and Jugal Kalita. Classifying Malware Images with Convolutional Neural Network Mo'dels. In *arXiv preprint arXiv:2010.16108*, 2020. URL https://arxiv.org/pdf/2010.16108.pdf.

[2] L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath. Malware Images: Visualization and Automatic Classification. University of California, Santa Barbara, 2011. URL https://vision.ece.ucsb.edu/sites/default/files/publications/nataraj_vizsec_2011_paper.pdf.

[3] Ria Kulshrestha. Malware Detection Using Deep Learning. *Towards Data Science*, 2020. URL https://towardsdatascience.com/malware-detection-using-deep-learning-6c95dd235432.

[4] A. S. Bozkir, A. O. Cankaya and M. Aydos, "Utilization and Comparision of Convolutional Neural Networks in Malware Recognition," 2019 27th Signal Processing and Communications Applications Conference (SIU), Sivas, Turkey, 2019, pp. 1-4, doi: 10.1109/SIU.2019.8806511. https://ieeexplore.ieee.org/document/8806511

[5] Rushiil Deshmukh, Angelo Vergara, Debtanu Bandyopadhyay, Kevin Huang. Malware Classification using Machine Learning and Deep Learning. URL https://medium.com/@rushiil.deshmukh/malware-classification-using-machine-learning-and-deep-learning-4de22e194dbe

[6] Copiaco, A.; El Neel, L.; Nazzal, T.; Mukhtar, H.; Obaid, W. A Neural Network Approach to a Grayscale Image-Based Multi-File Type Malware Detection System. Appl. Sci. 2023, 13, 12888. https://doi.org/10.3390/app132312888. URL https://www.mdpi.com/2076-3417/13/23/12888

[7] Jayasudha M, Ayesha Shaik, Gaurav Pendharkar, Soham Kumar, Muhesh Kumar B, Sudharshanan Balaji. Comparative Analysis of Imbalanced Malware Byteplot Image Classification using Transfer Learning In *arXiv preprint arXiv:2010.16108*, 2020. URL https://doi.org/10.48550/arXiv.2310.02742.

[8] bin2png URL https://github.com/ESultanik/bin2png