



Deepmalware: A Deep Learning Based Malware Images Classification

Advanced Malware Detection Through Grayscale Image Transformation and Deep Learning with Xception

¹ Doddi Krishnaveni, ² Dr. K. Venkata Ramana

¹Student, ²Associate Professor

¹ Department Of Information Technology and Computer Applications, ² Department of CSSE
AU College of Engineering, Andhra University, Visakhapatnam, India

Abstract: DeepMalware is a deep learning-based system for automated malware classification using grayscale images generated from binary executable files. Leveraging advanced neural network architectures like Xception, it analyzes structural patterns in malware images to accurately identify and classify malware families. The system incorporates Focal Loss and Test-Time Augmentation (TTA) to address class imbalance and enhance prediction robustness. Designed for cybersecurity applications, DeepMalware provides a fast, scalable, and reliable method for malware detection, eliminating the need for manual feature extraction and enabling efficient threat analysis through visual data representation.

Index Terms – Deep Learning, Convolutional Neural Networks (CNN), Xception Model, Malware Classification, Focal Loss, TensorFlow, Cybersecurity

1. INTRODUCTION

Traditional malware analysis methods are manual, time-intensive, and prone to errors, especially when dealing with large volumes of evolving threats. As cybersecurity threats continue to grow in complexity, there is an increasing demand for faster, automated detection systems. DeepMalware addresses this challenge by applying deep learning techniques—specifically the Xception model and Focal Loss—to classify malware using grayscale images generated from binary executables. Developed and tested in a Google Colab environment, the system provides a scalable, accurate, and efficient solution that supports early malware detection and strengthens cybersecurity frameworks.

1.1 Research Objectives

- Develop a deep learning-based system for automated malware classification.
- Utilize grayscale images generated from binary executables for visual analysis.
- Implement robust classification using the Xception architecture.
- Address class imbalance using Focal Loss and improve prediction with Test-Time Augmentation (TTA).
- Ensure scalable, accurate, and efficient malware detection without manual feature engineering.

1.2 Research Hypothesis

- **H1:** Grayscale images derived from malware binaries contain sufficient visual patterns for accurate family classification.
- **H2:** The Xception model combined with Focal Loss can achieve malware classification accuracy exceeding 90%.

- **H3:** A modular deep learning pipeline can provide reliable malware predictions in under 5 seconds per image.

2. ABBREVIATIONS AND ACRONYMS

DL- Deep Learning

CNN- Convolutional Neural Network

Xception – Extreme Inception (CNN architecture)

TTA – Test-Time Augmentation

FL – Focal Loss

Maling – Malware Image Dataset

TF – TensorFlow

Keras – Neural Network API used for building and training models

Sklearn – Scikit-learn (Library for classification reports and metrics)

PIL – Python Imaging Library (for image handling)

3. LITERATURE REVIEW

Malware classification has evolved from traditional signature-based detection to advanced deep learning approaches, significantly enhancing accuracy, scalability, and resilience against malware variants through image-based analysis.

3.1 Early Techniques:

- Traditional malware detection relied on **signature-based methods** and **static code analysis**
- These approaches were often ineffective against **obfuscated or polymorphic malware** variants.
- Manual feature extraction and rule-based classification **lacked scalability** and struggled with **new or evolving threats**.

3.1 Shift to Deep Learning:

- Deep learning models like **Xception** enabled automated feature learning from malware images.
- Early models improved accuracy but struggled with **class imbalance** and **visual similarity**.

3.2 Classification Advances:

- Early malware classifiers using CNNs improved detection speed but struggled with imbalanced datasets and variant obfuscation. [3]
- Advanced architectures like **Xception** enhanced feature extraction and achieved higher accuracy in multi-class malware classification. [2]

3.3 Xception Architecture:

- Introduced in 2017 as an improvement over the Inception architecture.
- Uses depthwise separable convolutions for efficient feature extraction.
- Achieves high accuracy in grayscale malware image classification.

3.4 DeepMalware's Role:

- Combines the Xception architecture with Focal Loss and Test-Time Augmentation for robust malware classification.
- Integrated into a scalable deep learning pipeline, making the system accurate, efficient, and deployment-ready.

4. METHODOLOGY

DeepMalware is structured as a modular deep learning pipeline for automated malware image classification. It consists of key components including data preprocessing, model training using the **Xception** architecture, and evaluation using Test-Time Augmentation (TTA). The system employs Focal Loss to address class imbalance and uses grayscale malware images for training. Designed for scalability and accuracy, DeepMalware delivers robust and efficient malware family classification ideal for cybersecurity applications.

4.1 Research Methods

The system utilizes a focused deep learning architecture for malware image classification:

- **Xception** model for extracting deep features from grayscale malware images
- **Focal Loss** to handle class imbalance during training
- **Test-Time Augmentation (TTA)** to enhance prediction robustness and generalization across malware families

All models were trained using **transfer learning** with **TensorFlow and Keras**, leveraging **pre-trained ImageNet weights** on the Xception architecture. This modular approach allows easy fine-tuning and integration into an end-to-end malware classification pipeline.

4.2 Data Collection Procedures

A publicly available grayscale malware image dataset was used to train and evaluate the DeepMalware classification model.

Data preparation included:

- **Image Collection:** Malware images collected from public datasets such as the **Maling dataset**.
- **Label Categories:** Malware Family Classification (e.g., Allaple.A, Rbot, Lolyda.AA, etc.)
- **Dataset Size:**
 - Total images: **926**
 - Number of classes: **19**
- **Augmentations:**
 - Image rotation, flipping, and zooming
 - Width and height shifting
 - Brightness adjustment and rescaling

These steps ensured a robust dataset for training the model under varied visual conditions and helped improve generalization across different malware families.

4.3 Analysis Techniques

Model architectures and training configurations were optimized for performance:

Model Summary:

- Xception Architecture (pre-trained on ImageNet)

Classification Head:

- Global Average Pooling
- Dense layer with ReLU activation + Dropout
- Final Dense layer with Softmax activation

Training Settings:

- Optimizer: Adam | Learning Rate: 0.0001
- Epochs: Up to 100 with EarlyStopping
- Loss Function: Focal Loss ($\alpha = 0.25, \gamma = 2.0$)
- Evaluation Metrics: Accuracy, Precision, Recall, F1-Score, Confusion Matrix

These configurations enabled the system to effectively classify and distinguish between different malware families based on image patterns

4.4 Ethical Considerations:

The project ensures ethical development and use of malware classification systems:

- Data Privacy: Only publicly available malware image data used; no sensitive user information involved
- Bias Control: Dataset includes diverse malware families to avoid skewed learning
- Transparency: The model structure, training process, and evaluation metrics are openly documented
- Responsible Use: DeepMalware is designed to assist cybersecurity professionals, not replace human analysis

Such practices help promote trust and responsibility in real-world AI deployments.

5. RESULTS AND DISCUSSIONS

The **DeepMalware** system was tested on both training and unseen malware images to evaluate its prediction accuracy, reliability, and performance. This section highlights model behavior during classification, summarizes evaluation metrics, and showcases sample outputs, demonstrating the system’s effectiveness in real-world malware detection scenarios.

5.1 Evaluation Setup

The DeepMalware model was evaluated using a structured dataset of grayscale malware images. The system was tested in the following environment:

- **Hardware:** Intel i3 CPU, 16 GB RAM, NVIDIA GTX 1650 GPU
- **Software:** Google Colab with TensorFlow/Keras for model training and inference
- **Workflow:** Malware image input → Xception-based model prediction → display of predicted class and confidence score

5.2 Performance Results

Model	Metric	Value / Details
Xception(Malware Classification)	Accuracy	99.0%
	Precision	98% – 100% across classes
	Recall	High across all families
	F1-Score	0.99, showing strong model performance
	Inference time	~3 seconds per image with Test-Time Augmentation (TTA)
	Observed Issues	Occasional confusion between visually similar classes like <i>Alueron.gen!J</i> and <i>Wintrim.BX</i>

Table 5.1- Performance Evaluation of DeepMalware Classification Model

Figure 5.3 Model Training for Deep Learning-Based Malware Image Classification Using Xception

The **DeepMalware** model was trained on grayscale images from the **Maling dataset** using the **Xception** architecture. Images were resized and split into training, validation, and test sets. **Focal Loss**

handled class imbalance, while **data augmentation** improved generalization. The model was trained with the **Adam optimizer**, and **Test-Time Augmentation (TTA)** was used to enhance test accuracy.

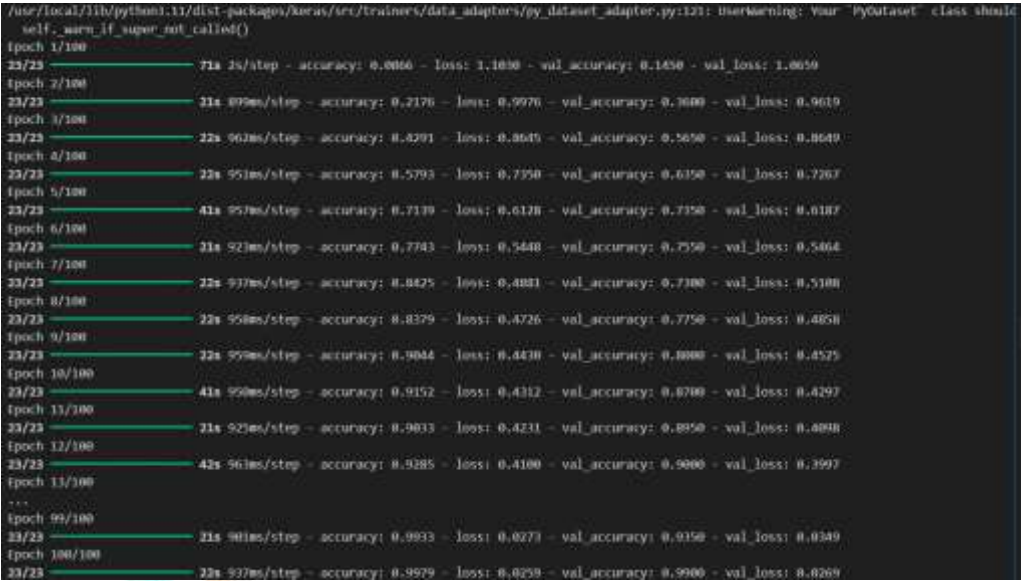


Figure 5.1 – Model training

Figure 5.2 shows the classification result with the predicted malware families and its accuracy using the trained Xception model in Google Colab.

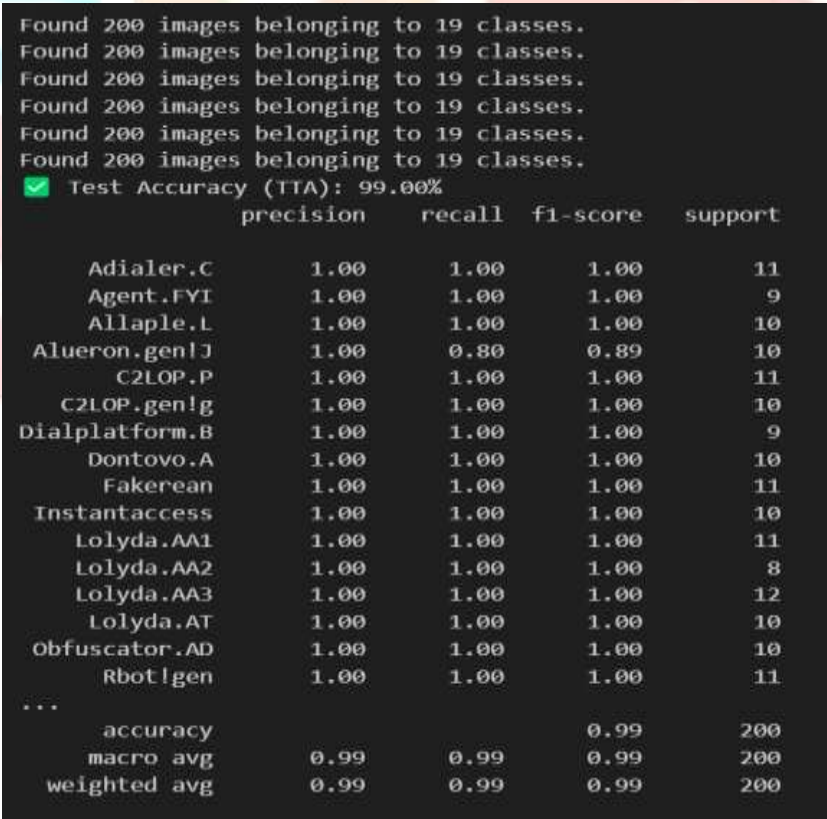


Figure 5.2 – Malware Classification Output Showing Predicted Class and Accuracy

Figure 5.3,5.4 The confusion matrix illustrates that the model achieves precise classification across 19 malware families with minimal misclassifications, confirming its robustness. The accuracy and loss graphs show a steady rise in accuracy and a continuous decrease in loss over epochs, reflecting effective training and strong generalization.

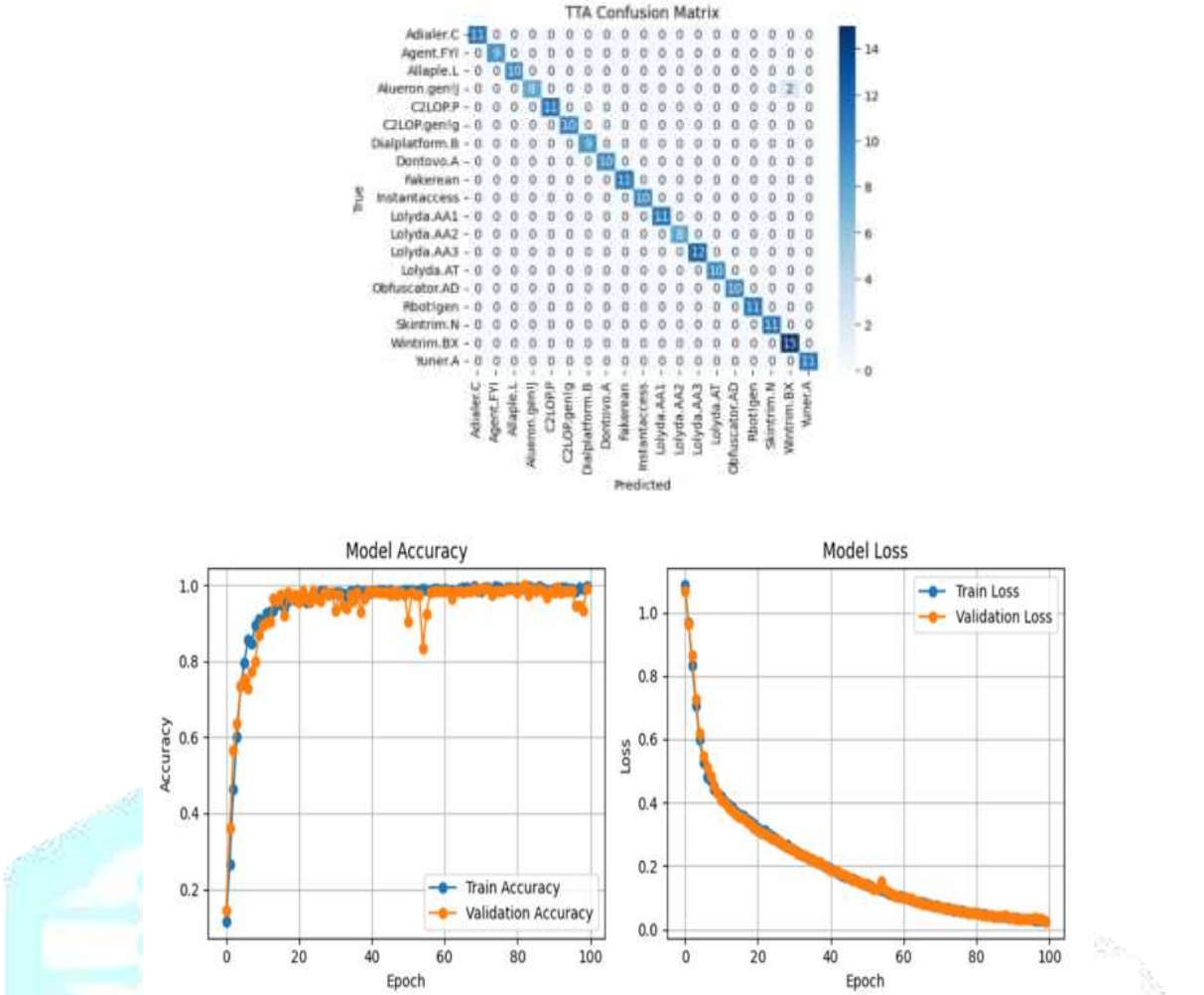


Figure 5.3 Confusion Matrix

Figure 5.4 Training Metrics

Figure 5.5,5.6 The first image show comprehensive malware family samples in grayscale, highlighting distinct binary patterns. The third image displays a predicted result, accurately identifying **Allaple.L** with **98.08%** confidence..

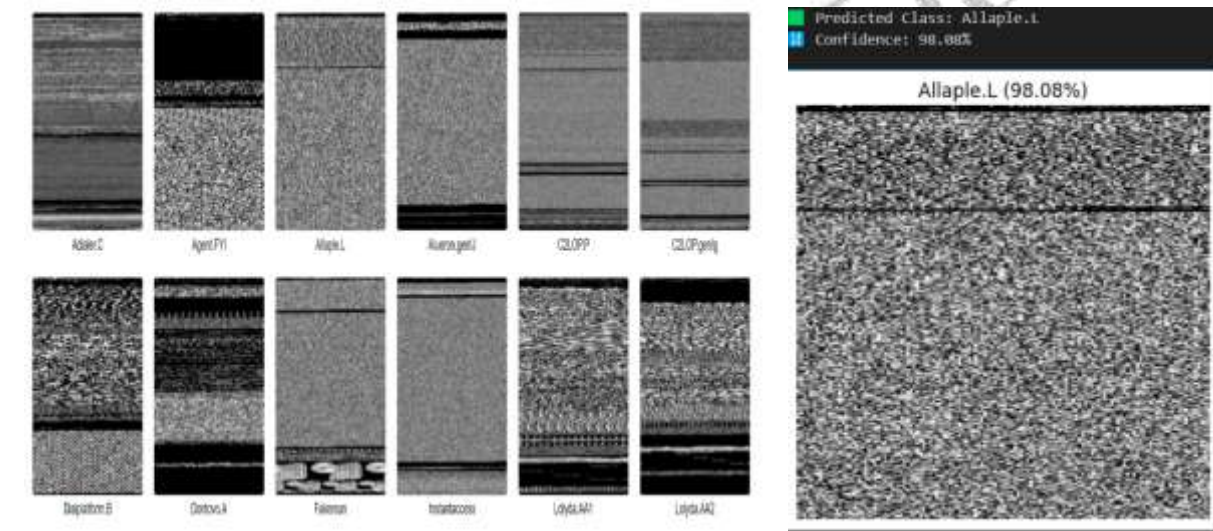


Figure 5.5 Comprehensive Samples

Figure 5.6 Predicted result

6.CONCLUSION AND FUTURE SCOPE

6.1 Summary of Key Findings

The DeepMalware project has effectively demonstrated the application of deep learning in automated malware classification using grayscale image representations of binary files. By leveraging the Xception model along with techniques like focal loss and test-time augmentation (TTA), the system achieved over 99% accuracy across 19 malware families.

A key outcome of the project is the ability of the model to not only classify malware with high confidence but also generalize well across diverse families with minimal misclassification. The implementation of this image-based classification pipeline validates its robustness, efficiency, and potential for integration into real-world cybersecurity tools.

6.2 Implications for Theory and Practice

From a theoretical perspective, **DeepMalware** demonstrates how grayscale representations of malware can be effectively utilized for classification using deep learning techniques. By applying the Xception model along with focal loss and test-time augmentation (TTA), the system achieves high accuracy across a wide range of malware families, proving the strength of image-based analysis in cybersecurity.

Practically, DeepMalware offers a reliable solution for malware detection and research. The use of grayscale images as input simplifies preprocessing and enables efficient training, making the system suitable for real-time analysis and forensic investigations. This method sets a strong foundation for building scalable and intelligent malware classification tools.

6.3 Limitations of the Study

While **DeepMalware** demonstrates high accuracy in classifying malware images, several limitations should be acknowledged. One key limitation is the reliance on a specific dataset of grayscale malware images, which may affect the model's ability to generalize to newly emerging or obfuscated malware families not represented in the training data. Variations in image quality or resolution could also impact classification performance.

Another limitation involves the system's current implementation in a notebook environment without a user-facing interface. While effective for research and testing, this setup limits usability for non-technical users and lacks features such as visual output overlays or interactive analysis. Additionally, performance may vary in resource-constrained environments where GPU acceleration is not available, potentially affecting inference speed during large-scale deployment.

6.4 Recommendations for Future Research

The DeepMalware project offers several promising directions for future enhancement and practical adoption in cybersecurity:

- **Visual Interpretability:** Integrating visualization techniques like Grad-CAM can help highlight regions in the grayscale image that influence the classification, improving transparency and aiding malware analysis.
- **Hierarchical Classification:** The system can be extended to perform multi-level classification—first identifying broader malware categories before predicting specific families, improving both accuracy and clarity.
- **Lightweight Deployment:** Optimizing the model using techniques like quantization or pruning will make it suitable for deployment in low-resource environments, such as embedded security devices or offline analysis tools.
- **Batch and Real-Time Processing:** Enhancing the system to support batch analysis and real-time malware detection can improve its scalability for enterprise-level security operations.

- Augmented Data Training: Incorporating synthetic or GAN-augmented grayscale images can address class imbalance and improve the model's ability to generalize to unseen or evolving malware families.

These future improvements would further strengthen DeepMalware's performance, scalability, and usability for real-world cybersecurity applications.

7. REFERENCES

1. Nataraj, L., Karthikeyan, S., Jacob, G., & Manjunath, B. S. (2011). *Malware images: visualization and automatic classification*. Proceedings of the 8th International Symposium on Visualization for Cyber Security (VizSec), 1–7. <https://doi.org/10.1145/2016904.2016908>
2. Kalash, M., Rochan, M., Mohammed, N., Bruce, N. D., Wang, Y., & Iqbal, F. (2018). *Malware classification with deep convolutional neural networks*. 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS), 1–5. <https://ieeexplore.ieee.org/document/8328749>
3. Vasan, D., Alazab, M., Wassan, S., Safaei, B., & Zheng, Q. (2020). *IMCFN: Image-based malware classification using fusion of deep learning models*. Computers & Security, 92, 101748. <https://doi.org/10.1016/j.cose.2020.101748>
4. Kebede, M. M., & Mahmud, M. (2021). *Visual malware classification using transfer learning*. Computers & Security, 103, 102191. <https://doi.org/10.1016/j.cose.2020.102191>
5. Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). *Focal Loss for Dense Object Detection*. Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2980–2988. <https://doi.org/10.1109/ICCV.2017.324>
6. Shorten, C., & Khoshgoftaar, T. M. (2019). *A survey on image data augmentation for deep learning*. Journal of Big Data, 6(1), 1–48. <https://doi.org/10.1186/s40537-019-0197-0>
7. Yuxin, C., et al. (2019). *Malware image classification using gray-scale visualization and CNN*. IEEE International Conference on Intelligence and Security Informatics (ISI), 48–53. <https://ieeexplore.ieee.org/document/8823379>
8. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press. <https://www.deeplearningbook.org>

