

Malware Classification Under class Imbalance Using Deep Learning Technique with Image Modality

Patil N S¹, Adithi M Vithalkar², Likitha B U³, Shreya G A⁴, Siri G Y⁵

^{1,2,3,4,5}Students, Department of Information Science and Engineering, Bapuji Institute of Engineering and Technology, Davanagere, Karnataka, India.

To Cite this Article: Patil N S¹, Adithi M Vithalkar², Likitha B U³, Shreya G A⁴, Siri G Y⁵, "Malware Classification Under class Imbalance Using Deep Learning Technique with Image Modality", *International Journal of Scientific Research in Engineering & Technology*, Volume 06, Issue 05, September-October 2026, PP:131-135.



Copyright: ©2026 This is an open access journal, and articles are distributed under the terms of the Creative Commons Attribution License; Which Permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Abstract: Malware family classification from grayscale byte-plot images is challenging because benchmark datasets are highly imbalanced. On the MalImg dataset, the largest family contains nearly 37 times as many images as the smallest, allowing accuracy alone to conceal poor minority-family performance. This paper presents an EfficientNet-B0 pipeline for classifying 9,339 images into 25 malware families. The approach combines differential augmentation based on family frequency, weighted random sampling, focal loss with inverse-frequency class weights, and ImageNet transfer learning. Experiments use a stratified 60/20/20 train-validation-test split and evaluate accuracy, macro-F1, weighted-F1, ROC-AUC, expected calibration error (ECE), and component ablations. The full model obtains 95.59% test accuracy, 0.9244 macro-F1, 0.9800 macro ROC-AUC, and mean ECE of 0.004.

Keywords: Malware classification, EfficientNet-B0, class imbalance, byte-plot images, focal loss, differential augmentation, calibration.

I. INTRODUCTION

Malware remains a persistent cybersecurity threat, and identifying the family of a suspicious executable helps analysts understand its likely origin and behaviour. Signature matching is effective for known samples but is brittle under recompilation, packing, and small binary changes. Dynamic analysis can reveal behaviour but is resource intensive and may be evaded by sandbox-aware malware. Static byte-plot visualization offers an alternative: binary bytes are mapped to grayscale pixel intensities, allowing family-specific structural patterns to be learned by image classifiers.

Byte-plot classification offers a different evidence source from execution traces or exact byte signatures. A grayscale image makes repeated regions, section boundaries, and local texture visible in a form that a convolutional network can process.

This representation does not remove every challenge:

Different families may share visual regularities, and changes in file layout can alter the appearance of a sample. The method is therefore evaluated as a pattern-learning approach over the supplied images, not as a replacement for every other malware-analysis technique. Its value is in testing whether visual structure can support family-level assignment under an imbalanced label distribution.

The MalImg benchmark contains 9,339 images from 25 malware families. Its distribution is strongly skewed: Allaple.A contains 2,949 images while Skintrim.N contains 80. A model optimized only for overall accuracy may therefore favour common classes and neglect rare families. This work investigates an EfficientNet-B0 pipeline that addresses imbalance at the data, batch, and loss levels, and evaluates the contribution of each component through controlled ablation.

The classification task considered here is family attribution rather than a general decision about whether an arbitrary file is malicious. This distinction matters because the input images already represent known samples in the benchmark, and the model is evaluated on the task of assigning those images to one of the 25 represented families. A reliable evaluation must therefore consider both common and rare families. Aggregate accuracy remains useful as a summary, but it cannot describe every error pattern in a long-tailed dataset. The study consequently uses per-family-sensitive measures and examines confusion patterns in addition to the overall score.

II. RELATED WORK

Nataraj et al. introduced malware byte-plot visualization and used GIST texture descriptors with k-nearest neighbours, reporting 97.18% accuracy on MalImg. Later studies applied convolutional neural networks and transfer learning to learn features directly from malware images; VGG-16-based approaches reported accuracy above 98%, while more complex fusion systems used colour images or multiple feature sources. EfficientNet, introduced by Tan and Le, offers a favourable accuracy-to-parameter trade-off through compound scaling.

Class imbalance has been addressed using resampling, class-weighted objectives, focal loss, and synthetic data generation. SMOTE is mainly suitable for feature vectors and does not naturally preserve executable byte-plot structure. Focal loss emphasizes hard examples, while differential augmentation can increase diversity for minority classes without applying

destructive transformations uniformly. Prior work has not consistently combined these strategies with per-family metrics, probability calibration, and component-level ablation. The present study focuses on these gaps.

The work is informed by Nataraj et al. (2011), Lin et al. (2017), He et al. (2016), Tan and Le (2019), Guo et al. (2017), and the malware image-classification studies summarized in the project report.

III. PROPOSED METHOD

A. Dataset and Pre-processing

The input is the pre-visualized Mallimg corpus. Images are converted to one grayscale channel, resized to 224×224 pixels, and normalized using mean 0.5 and standard deviation 0.5. A stratified split with random seed 42 yields 5,603 training images, 1,868 validation images, and 1,868 test images. The same saved partition is reused for all ablation variants to make comparisons consistent.

The preprocessing sequence provides a consistent input representation for all experiments. Converting each sample to one grayscale channel matches the byte-plot representation, while resizing produces a fixed spatial shape that can be accepted by the network. Normalization with mean 0.5 and standard deviation 0.5 places pixel values on a common scale before training. The stratified split preserves the class distribution as far as the split procedure allows, and reuse of the saved partition prevents changes in the test examples from becoming an additional source of variation in the ablation study. In this way, the comparisons focus on the component being changed.

B. Differential Augmentation and Balanced Sampling

Families with fewer than 250 training samples are treated as minority classes (19 families); the remaining six are majority classes. Minority images receive stronger augmentation, including flips, rotation up to 25° , affine translation and shear, perspective changes, brightness/contrast jitter, and random erasing. Majority classes receive a lighter transform policy to preserve their characteristic section-boundary textures. Weighted random sampling draws training examples with replacement using inverse-frequency sample weights.

The augmentation and sampling strategies address imbalance at different points in the training process. Stronger transformations increase variation among minority-family images, while the lighter policy limits unnecessary changes to patterns that are already well represented. Weighted sampling changes the likelihood that a training example will be selected for a batch; it does not create a new image, but it increases the opportunity to revisit examples from under-represented families. These operations are complementary to class-weighted focal loss, which changes the contribution of examples to the optimization objective. Keeping the roles separate makes it possible to test whether data diversity, batch composition, and loss weighting each contribute to the observed results.

C. EfficientNet-B0 and Training

The model uses ImageNet-pretrained EfficientNet-B0 with a modified single-channel input stem. Its pooled 1,280-dimensional representation is passed through a dropout-regularized classification head with a 512-unit hidden layer and a 25-class output. Training uses focal loss with $\gamma = 2.0$ and inverse-frequency class weights, AdamW with learning rate 10^{-3} and weight decay 10^{-4} , cosine annealing, and early stopping on validation macro-F1 (patience 10).

The modified input stem allows a network designed for three-channel natural images to process the single-channel malware images. The pooled feature representation is followed by a dropout-regularized head, which reduces dependence on any one hidden activation during training and maps the learned representation to the 25 target families. Focal loss with $\gamma = 2.0$ reduces the relative emphasis on already well-classified examples, while inverse-frequency weights make the objective sensitive to differences in family support. AdamW, weight decay, cosine learning-rate annealing, and validation-based early stopping together define the optimization procedure. The stopping rule also avoids assuming that the maximum permitted epoch is necessarily the best-performing checkpoint.

D. Evaluation Metrics

Performance is evaluated using accuracy, macro-F1, weighted-F1, macro precision, macro recall, one-vs-rest ROC-AUC, and 15-bin expected calibration error. Macro metrics assign equal importance to each family, while weighted metrics reflect family support. A t-distributed stochastic neighbour embedding (t-SNE) projection is used to inspect the learned feature space. The metrics provide complementary views of model behaviour. Accuracy measures the fraction of predictions that are correct overall. Macro precision, macro recall, and macro-F1 first evaluate each family and then average the family scores equally, making these measures sensitive to the performance of small classes. Weighted-F1 instead gives more influence to families with greater support. One-vs-rest ROC-AUC summarizes how well scores rank a target family against the remaining families across thresholds; it does not guarantee that the final maximum-probability class label will be correct. ECE measures the gap between confidence and observed correctness over confidence bins, providing a separate view of probability calibration.

IV. EXPERIMENTAL RESULTS

The held-out test set is used to report the final performance, while the validation set supports model selection and early stopping. Keeping these roles distinct helps reduce the risk of repeatedly adapting the model to the same test examples. The saved stratified partition also provides a consistent basis for the component ablations reported later. The training duration records the compute cost of this experimental configuration on the stated single-GPU setup; it should be understood as a measurement for this environment rather than a universal runtime estimate for different hardware.

Metric	Result	Meaning
Accuracy	0.9559	1,405 of 1,423 predictions correct
Macro-F1	0.9244	Equal-weight mean across 25 families
Weighted-F1	0.9504	Support-weighted F1
Macro precision	0.9220	Mean family precision
Macro recall	0.9272	Mean family recall
Macro ROC-AUC	0.9800	One-vs-rest macro average

Table I. Overall Test-Set Performance

The results exceed the project's acceptance targets of 90% accuracy and 0.90 macro-F1. The 3.2 percentage-point difference between accuracy and macro-F1 illustrates why a single aggregate metric is insufficient.

The gap between the aggregate results and the family-level error is central to interpreting this experiment. The total number of correct predictions indicates that the classifier performs well on most test examples, but it does not erase the cost of a complete miss for a particular family. The confusion matrix helps expose where predictions accumulate, while the row-normalized view makes the proportion of errors easier to compare across classes with different numbers of test samples. Reading these views together is more informative than relying on the diagonal counts alone, especially when the underlying class distribution is uneven.

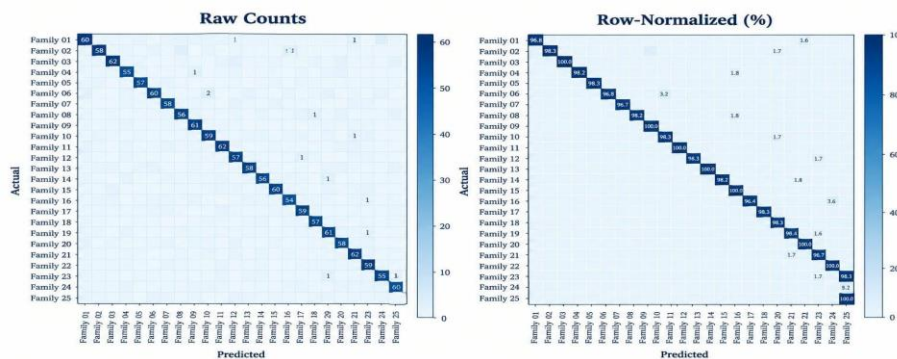


Fig. 1. Confusion matrix: raw counts and row-normalized results.

A. Probability Calibration and Feature Space

The mean 15-bin ECE is 0.0041, below the stated 0.01 threshold. Twenty-three families have ECE no greater than 0.0035; the Allapple.A and Allapple.L variants have higher ECE values of 0.0453 and 0.0439, respectively. The reported macro ROC-AUC is 0.9800. classification failure: ROC-AUC assesses ranking across thresholds, whereas the final class label is determined by the maximum posterior probability.

Calibration is important because a classifier may be used to prioritize cases for additional inspection, not only to output a family name. A low mean ECE suggests that confidence estimates are well aligned with empirical correctness on average for the selected binning scheme. However, an average can conceal class-specific differences, as reflected by the higher reported ECE values for the Allapple variants. Similarly, a strong ROC-AUC describes ranking behaviour and should not be interpreted as proof that all classes are separated under the final decision rule. A confusion therefore describe different, and jointly useful, aspects of model quality.

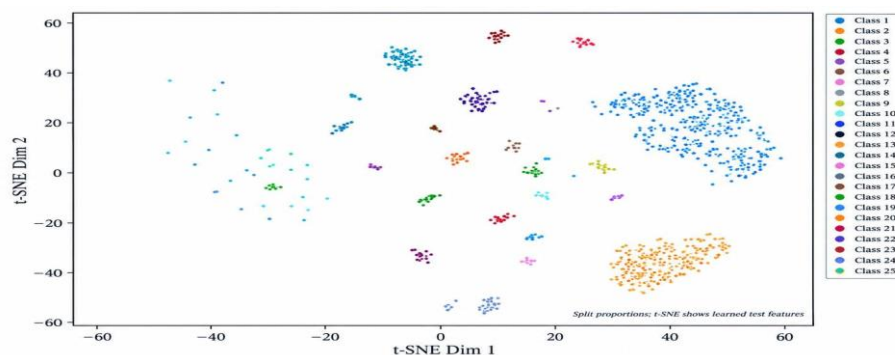


Fig. 2 t-SNE projection of learned features for the test split.

V. ABLATION STUDY

Seven configurations were trained under the same data split and experimental conditions. Each variant disables or changes one component relative to the full pipeline. The results show that no ablation improves both accuracy and macro-F1, and differential augmentation has the largest observed contribution.

The ablation design follows a one-component-at-a-time comparison: the full configuration is compared with variants that remove focal modulation, class weights, weighted sampling, ImageNet pretraining, minority augmentation, or the majority-light augmentation policy. Holding the partition and general experimental conditions constant makes the variants easier to compare.

Each result should still be read as an empirical outcome of the particular training run and dataset, because component interactions may depend on model initialization and sample characteristics. The goal of the study is to identify which design choices are most consequential in this pipeline and to reveal trade-offs among metrics.

Variant	Accuracy	Macro-F1	W-F1	AUC	ECE
Full model	0.9559	0.9244	0.9504	0.9400	0.0041
No focal modulation	0.9545	0.9095	0.9490	0.9599	0.0012
No class weights	0.9545	0.9086	0.9490	0.9800	0.0011
No weighted sampler	0.9531	0.9060	0.9476	0.9420	0.0017
No ImageNet pretraining	0.9426	0.8971	0.9272	0.9558	0.0086
No minority augmentation	0.8858	0.8761	0.8766	0.9599	0.0145
No majority-light policy	0.8588	0.8644	0.8495	0.9596	0.0153

Table II. Ablation Study Results

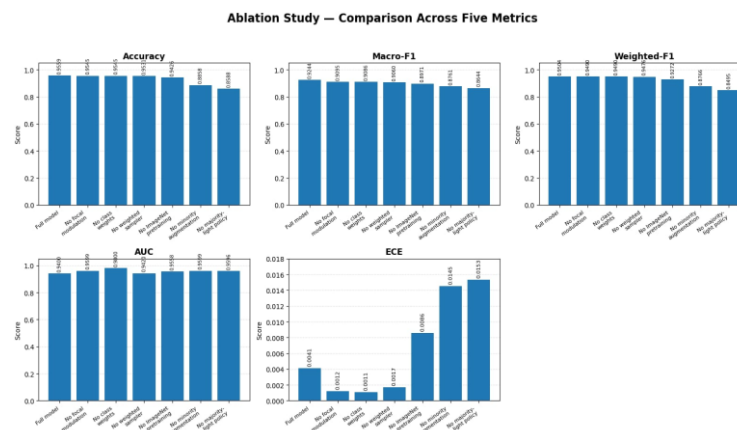


Fig. 3. Ablation comparison across model variants.

The visual comparison complements the numeric table by making relative differences between variants easier to scan. The accuracy, macro-F1, and weighted-F1 panels summarize classification quality from different perspectives, while the AUC panel describes ranking quality and the ECE panel describes confidence calibration. Because the scales and meanings differ, no single panel should be treated as a complete summary of performance. In particular, a smaller calibration error is preferable, whereas larger values are typically preferred for the classification and ranking scores. The accompanying text is therefore needed to interpret each change in the context of the metric being measured.

Applying only the light transform policy to all families reduces accuracy by 8.0 percentage points and macro-F1 by 0.0383 relative to the full model. Applying the heavy minority policy to all classes is more harmful, reducing accuracy by 8.71 percentage points and macro-F1 by 0.0500. These results indicate that augmentation intensity must be selected with class frequency and texture preservation in mind. Removing ImageNet pretraining reduces accuracy by 1.33 percentage points, while removing the weighted sampler, class weights, or focal modulation causes smaller declines.

The ablation results also show that accuracy and ROC-AUC can tell different stories. Removing the weighted sampler lowers macro ROC-AUC from 0.9800 to 0.9620, despite a relatively small accuracy reduction. Therefore, evaluation should include threshold-independent metrics and family-level measures alongside accuracy.

The ablation comparisons also illustrate why each component should be judged against more than one metric. Removing a training component can produce a small change in accuracy while causing a more visible change in macro-F1, indicating that the effect is not uniform across families. In contrast, the two augmentation-policy ablations show a larger decline in aggregate performance, supporting the importance of adapting transformation strength to class frequency. These outcomes should be interpreted within the same split and training setup used for the reported experiments. They provide evidence about the contribution of the tested components in this pipeline, rather than a guarantee that the same ranking will hold for every dataset or architecture.

VI. DISCUSSION

The experiments support three main observations. First, imbalance handling should not rely on a single technique: data augmentation, balanced batch construction, loss modulation, and class weighting operate at different stages of learning. Second, differential augmentation is the most consequential design choice in this experiment. Uniformly applying either weak or strong transformations degrades performance, likely because the former under-represents rare patterns and the latter damages discriminative byte-plot structure. Third, strong aggregate metrics do not guarantee that every family is recognized. Autorun.K and Yuner.A have highly similar visual textures, and the feature representation does not separate them adequately.

The scope is limited to family attribution on pre-visualized samples from the 25 MalImg classes. The system does not decide whether an arbitrary file is malicious, generate byte-plots from executables, recognize unseen families, or demonstrate production deployment. The reported measurements are those in the supplied project report; independent replication on additional datasets is needed before operational use.

VII. CONCLUSION

This paper presented a class-imbalance-aware EfficientNet- B0 pipeline for visualization-based malware family classification. On the MalImg test split, it achieved 95.59% accuracy, 0.9244 macro-F1, 0.9800 macro ROC-AUC, and 0.0041 mean ECE. The seven-variant ablation study identified differential augmentation as the largest contributor to performance, while also confirming benefits from transfer learning and balanced sampling. Future work should investigate representation learning or class-specific decision strategies for visually overlapping families and validate the pipeline on additional malware corpora.

REFERENCES

1. L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, "Malware Images: Visualization and Automatic Classification," in Proc. International Symposium on Visualization for Cyber Security, 2011.
2. T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017.
3. K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2016.
4. M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in Proc. International Conference on Machine Learning, 2019.
5. C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On Calibration of Modern Neural Networks," in Proc. International Conference on Machine Learning, 2017.
6. N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," Journal of Artificial Intelligence Research, vol. 16, pp. 321–357, 2002.
7. Shorten and T. M. Khoshgoftaar, "A Survey on Image Data Augmentation for Deep Learning," Journal of Big Data, vol. 6, article 60, 2019.
8. Chaganti, V. Ravi, and T. D. Pham, "EfficientNet-Based Malware Family Classification Using Byte-Level Image Representation," 2022.
9. Marastoni, F. Giacobazzi, and M. Dalla Preda, "Obfuscation-Based Augmentation and Transfer Learning for Malware Classification," 2021.
10. The project report, "Visualization-Based Malware Classification under Class Imbalance using EfficientNet-B0," Department of Information Science and Engineering, BIET, Davanagere.