Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 4841741, 8 pages
https://doi.org/10.1155/2022/4841741

Research Article

An Efficient Convolutional Neural Network with Transfer Learning for Malware Classification

Musaad Darwish AlGarni,[1] Roobaea AlRoobaea,[1] Jasem Almotiri,[1] Syed Sajid Ullah,[2] Saddam Hussain,[3] and Fazlullah Umar[4]

[1]*College of Computers and Information Technology, Taif University, Taif 21974, Saudi Arabia*
[2]*Department of Information and Communication Technology, University of Agder (UiA), N-4898 Grimstad, Norway*
[3]*Department of Information Technology, Hazara University, Mansehra, 21120 KPK, Pakistan*
[4]*Department of Information Technology, Khana-e-Noor University, Pol-e-Mahmood Khan, Shashdarak, 1001 Kabul, Afghanistan*

Correspondence should be addressed to Roobaea AlRoobaea; mr.robai@gmail.com, Syed Sajid Ullah; sajidullah718@yahoo.com, and Fazlullah Umar; fazlullahumer@gmail.com

Received 17 June 2022; Revised 20 August 2022; Accepted 3 September 2022; Published 6 October 2022

Academic Editor: Kuruva Lakshmanna

Rising prevalence of malicious software (malware) attacks represent a serious threat to online safety in the modern era. Malware is a threat to anyone who uses the Internet since it steals data and causes damage to computer systems. In addition, the exponential growth of malware hazards that affect many computer users, corporations, and governments has made malware detection, a popular issue in academic study. Current malware detection methods are slow and ineffectual because they rely on static and dynamic analysis of malware signatures and behavior patterns to detect unknown malware in real-time. Thus, this paper discusses the role of deep convolution neural networks in malware classification and solutions for utilizing machine learning to detect and classify malware families through transfer learning. We proposed a CNN pretrained model learning to classify malware families. The experiment was conducted using two classification datasets, including Malimg and ImageNet. We classified the Malimg dataset, which has turned malware binaries into malware images by using Portable Executable. The result shows that the EfficientNet3 model achieved a high accuracy of 99.93%.

1. Introduction

Malware is a severe danger to computer security, and the ability to identify and categorize malware is vital for maintaining a computer's security level [1]. Much research has been done to identify malware families using malware visualization, which converts malware's binary structure into grayscale visuals. Numerous publications have used CNN to classify malware visualization images. Still, there is no method for selecting a model that matches a specific malware dataset and yields higher classification accuracy has been reported. Machine learning is a rapidly expanding area that has benefited immensely from technological breakthroughs. Many technologies are available in this domain that can conduct various functions on massive datasets.

In a world that is becoming more dependent on computers, malicious software, such as worms and viruses, has been a concern for a long time and continues to worsen. Worms and viruses are two examples of this type of software. One-time occurrences often bring about financial losses for companies amounting to tens of millions of dollars [2, 3]. For instance, in 2014, it was projected that the economic expenses associated with a subset of malware that was disseminated through pirated software amounted to over $500 billion [4]. Malware has seen an enormous increase in the number of updated versions over the past many years, demonstrating that malware has proven to be successful for the developers [5]. The proliferation of malicious software serves only to emphasize the urgent requirement for improved tools that can eliminate the threat

posed by malware and support the work of security researchers and professionals.

Malware classification is one area that could use better. Malware classification has been conducting intensive research for a significant amount of time on the detection of malware attacks (i.e., distinguishing between harmless and harmful programmes) and the differentiation between two or more existent virus classes or groups [6]. The identification of malware is absolutely necessary in order to stop the spread of malware. On the other hand, the vast majority of antivirus (AV) software accomplishes this primarily through the use of signatures, which requires the analyst to invest time and effort. Because of this, it is less probable that signatures will scale. In addition, providers of antivirus software are well aware of this issue [7].

Malware has seen a dramatic growth in its production as of late, posing a grave threat to the information security of businesses, organizations, and individual users [8–12]. In order to put a halt to the propagation of malware, new methods of immediately identifying and classifying malware samples in order to investigate their behavior are required. Even while machine learning techniques for the classification of malware are becoming more popular, the majority of malware classifications are extremely simplistic. In addition, traditional machine learning necessitates the use of a substantial amount of resources and the engineering of features. Therefore, convolutional neural networks (CNN), excellent tools for picture categorization, have showed improved accuracy when compared to traditional techniques of learning [13].

1.1. Motivation and Contributions. Convolutional neural networks have gained a lot of momentum in the field of deep learning due to their usefulness in a variety of computer vision applications, including image processing, computer vision tasks such as localization and segmentation, video analysis, recognizing obstacles in autonomous vehicles, and natural language processing [14]. On the other hand, the process of extracting features was done manually. In this research, we proposed using transfer learning to categorize different families of malware using a CNN model that has been pretrained. The experiment was carried out with the help of two different classification datasets, namely, the Malimg and the ImageNet sets [15, 16]. The following are some of the major contributions of our research.

(i) The paper is aimed at describing transfer learning (TL) by applying the deep convolution network models for family malware classification

(ii) In addition, we used CNN pretrained models for this work to transfer learning to the malware classification task that had already been learned

(iii) We classified 9,342 malware sample by using EfficientNet deep learning model

1.2. Paper Organization. The paper is organized as follows. Section 2 gives a detailed review of related literature on malware classification methods. Section 3 describes the methodologies used in the malware classification experiment. Section 4 explains the experiment and result. Finally, Section 5 summarizes and concludes the paper with future work.

2. Literature Review

When it comes to computer security, one of the most challenging tasks may be figuring out what kind of malicious software a system is designed to protect against. This is because different malware families employ a variety of tactics. Image-based, dynamic, and static approaches are the three that are most commonly used when discussing the topic of malware type classification [17]. A technique known as static analysis is one that does not involve actively running the binary programme in order to extract information from it. Dynamic analysis refers to the process of studying malicious software by observing its behavior in real time within a controlled environment. The study and application of malware classification through the use of images is a substantial and expanding field. Deep learning, on the other hand, is one of the primary motivating factors behind research into computer vision and image processing. In the field of image processing, numerous deep convolutional neural networks (DCNs) have demonstrated promising results [18]. Moreover, Table 1 summarizes the literature review.

One of the earliest approaches that was utilized for the classification of malware was a method known as grayscale image classification. Grayscale representations of malicious characteristics can be extracted from the raw malware executable files [15, 19]. This is possible. Analysis of malware can be carried out by isolating visual components from images of this kind. Nataraj et al. [20] use a dataset of malware pictures that contained 9,342 different samples of malware representing 25 separate types. They were the first individuals to look into the possibility of using byte graphs as grayscale pictures for the purpose of automatically classifying malware. Torralba et al. [21] utilize this approach to extract GIST characteristics from grayscale images and categorize them utilizing the Euclidean distance as a metric. This was done in order to determine the relationship between the two. However, their approach comes at a significant cost in terms of computation. Using the wavelet transform, Makandar and Patrot [22] construct an efficient texture-based feature vector from the malware images. They then classified the malware using a multiclass support vector machine with the images of the malware serving as input. As a direct consequence of this, the dimensions of the feature vector as well as its temporal complexity were both reduced.

Deep learning is a common approach for analyzing enormous amounts of data. Deep learning makes use of intricate algorithms and artificial neural networks (ANNs) to train machines and computers to continue growing and developing, categorizing, and identifying data and pictures in the same manner that a human brain does [23]. Many researchers suggested using CNN as a classification system for malware. Cui et al. [24] perform a straightforward CNN analysis and identified the variant of codes by converting them into grayscale images. Malware was categorized by

Table 1: Summary of the literature.

S. no.	Scheme	Summary
1	[17]	(i) Classify types of malwares in three ways (1) Image-based (2) Dynamic (3) Static
2	[18]	(i) Shown many deep convolutional neural networks (DCNs) potential in image processing
3	[15, 19]	(i) Used the raw malware executable files to extract grayscale representations of malicious characteristics (ii) By extracting visual elements from such photos, malware can be analyzed.
4	[20]	(i) Worked with a malware picture dataset that included 9,342 malware samples from 25 distinct types. (ii) They were the first to examine the use of byte graphs as grayscale pictures for automated malware categorization.
5	[21]	(i) Utilized the approach of [20] to extract GIST characteristics from grayscale pictures and categorize them using the Euclidean distance as a metric. (ii) However, their method has a considerable computational cost.
6	[22]	(i) Built an effective texture-based feature vector from the malware images using the wavelet transform. (ii) Conducted malware classification using a multiclass support vector machine with malware input as images.
7	[24]	(i) Performed a simple CNN detected the variant of codes by turning them into grayscale images. Kalash et al. [25] classified malware by using two datasets, Malimg [15] and Microsoft [26]. (ii) Converted the malware binaries into malware images, their approach achieved high accuracy of 98.52% and 99.97%.
8	[27]	(i) Used two datasets, Malimg dataset and BigData gathering to build CNN model with four layers.
9	[28]	(i) Built a model by using deep transfer learning to classify two datasets, ImageNet [16] and Malimg [15]. (ii) Demonstrated high accuracy of 99.18%.
10	[29]	(i) Used two different approaches of feature extractions to classify Windows API Calls database. (ii) Shows that they depend on inverse document frequency vector and categorical vector. (iii) Proposed method score high accuracy above 90.0%.
11	[30]	(i) Presented one-dimensional CNNs to detect and classify malware families by using two datasets, Malimg [15] and Microsoft [26].

Kalash et al. [25] using two different datasets, namely, Malimg [15] and Microsoft [26]. They took the malware binaries and converted them into malware images, and as a result, their method achieved a high accuracy of between 98.52% and 99.97%. Gibert et al. [27] construct a CNN model with four layers by utilizing two datasets, namely, the Malimg dataset and the BigData gathering dataset.

Deep transfer learning was used by Kumar [28] to create a model that could classify two different datasets, namely, ImageNet [16] and Malimg [15]. They have a high level of accuracy, which is 99.18%. When classifying the Windows API Calls database, Schofield et al. [29] use two distinct methods of feature extractions to approach the problem. They are dependent on the inverse of the document frequency vector as well as the categorical vector. Accuracy levels above 90% were achieved using their suggested approach. One-dimensional convolutional neural networks (CNNs) are presented by Lin and Yeh [30] in order to detect and classify malware families. They do this by utilizing two different datasets, namely, Malimg [15] and Microsoft [26].

3. Methodology

In this paper, we used transfer learning to classify malware on the Malimg dataset. The models we used are EfficientNets B0-B7. Before that, we take consider turning malware from binaries into images.

3.1. Malimg Dataset. The main idea of the Malimg dataset is how malware binaries are transformed into grayscale images for inclusion in the Malimg collection [15]. The malware binary is broken down into 8-bit vectors, and each vector is treated as a pixel value in a grayscale image. There are 9,339 malware images in the dataset, representing twenty-five different malware families. Table 2 depicts the number of malware strains that are known to exist. It is worth noting that the Allaple.A malware family has the most samples (2,949) while the Skintrim.N malware family has the fewest (80 samples). As a result, the dataset in Table 1 shows that it is unequal.

3.1.1. Malware as Binaries. Portable Executable (PE) files are programs with file extensions like .bin, .dll, and .exe as shown in Figure 1 [31]. PE files are typically identified by their component names, which are .data, .tex, .rsrc, and .rdat. The code part (.txt) is the initial component, and it contains the program's instructions. The. rdata is the portion that contains read-only data, .data is the part that contains data that can be edited, and .data is the part that contains data that can be amended, and .data is the part that contains data that can be modified, and .data is the part that contains data that can be modified. The fourth component is .rsrc, which stands for the malware's resource. Malicious data binaries can be transformed into grayscale images made up of textural patterns 8 bits at a time.

Table 2: Sample count in each malware.

Malware family	Numbers	Malware type
Adialer.C	123	Dialer
Agent.FYI	117	Backdoor
Allaple.A	2950	Worm
Allaple.L	1592	Worm
Alueron.gen!J	199	Trojan
Autorun.K	107	Worm AutoIT
C2LOP.gen!g	201	Trojan
C2LOP.p	147	Trojan
Dialplatform.B	178	Dialer
Donoto.A	163	Trojan downloader
Fakerean	382	Rouge
Instaccess	432	Dialer
Lolyada.AA1	214	PWS
Lolyada.AA2	185	PWS
Lolyada.AA3	124	PWS
Lolyada.AT	160	PWS
Malex.gen!J	137	Trojan
Obfuscator.AD	143	Trojan downloader
RBot!gen	159	Backdoor
Skintrim.N	81	Trojan
Swizzor.gen!E	129	Trojan downloader
Swizzor.gen!I	133	Trojan downloader
VB.AT	409	Worm
Wintrim.BX	98	Trojan download
Yuren.A	801	Worm

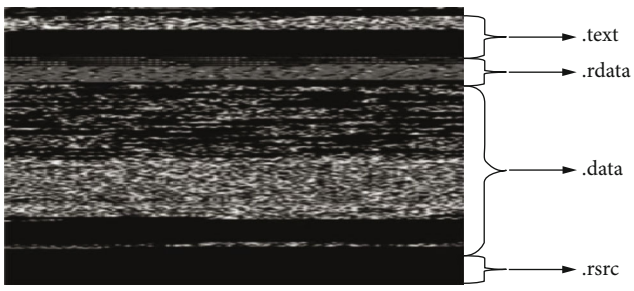

Figure 1: Portable Executable file represented as an image [25].

3.1.2. Malware as Image. This study found that viewing malware binaries as images become more evident. Deep learning can detect patterns inside photos. Malware families can also be identified using the essential patterns of features in malware images. To reveal significant patterns by automatically extracting features, a deep learning network uses images from a certain malware family that all have a similar pattern. CNN models are highly effective at classifying pictures because they can extract important characteristics from inside an image by subsampling, pooling, and other computations, making them particularly useful for image classification. For the aim of classification, CNNs hunt for the essential elements inside an image from a specific malware family [32]. Figure 2 depicts a way of converting a binary

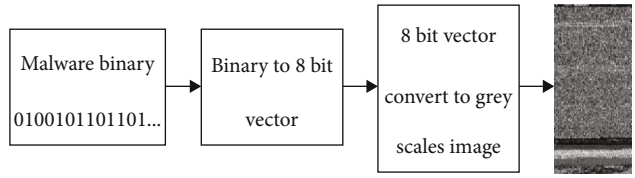

Figure 2: Converting malware binary to an image [16].

PE file into a series of 8-bit vectors or hexadecimal values, which may then be used to transform malware binaries into pictures. In Figure 2, an 8-bit vector can be represented by the numbers 00000000 (0) to 11111111. (255). 8-bit vectors represent numbers that can be transformed into pixels in the malware picture.

3.2. Transfer Learning. Transfer learning is a machine language that helps repurpose a pretrained model for one job to a unique type of work [33]. However, deep network training takes a long period and a lot of computational power. Thus, the main idea is behind using transfer learning. To sum up, we use the EfficientNets pretrained models on ImageNet to reuse on the Malimg dataset for malware family classification. Figure 3 shows the approach of transfer learning that we implemented.

3.3. EfficientNets 0-7 Models. EfficientNet introduced by Tan and Le [34] is an architecture of convolutional neural networks and a scaling methodology; EfficientNet has a group of models (B0 to B7) that are excellent at combining precision and effectiveness on a range of different scales, from small to large. The paper gives strict guidelines on how to scale from B0 to B7. Figure 4 shows the architecture of the base model B0.

EfficientNet scales from B0 following a method called compound scaling where input size, layer width, depth, and the number of channels are all scaled simultaneously following a given formula.

In Figure 5, conventional scaling (b–d) in the diagram above only improves one dimension of network breadth, depth, or resolution. (e) is a suggested compound scaling approach that uses a set ratio to scale all three dimensions consistently. Each EfficientNet model's depth, width, and resolution have been carefully selected since they have been shown to produce excellent outcomes. To respect the guidelines of the EfficientNet architecture, we resize the images for each model to its corresponding recommended size in the original paper.

4. The Experiment Result

We use pretrained EfficientNet models on the ImageNet dataset to save time and effort, and then, we add a classification layer, which is a dense (or fully connected) layer with 25 output units and SoftMax activations. Training EfficientNet from scratch can take days on a powerful NVIDIA GPU. There are 32025 different parameters that can be trained. Adam's optimizer was utilized in the training of the models. When the model validation loss did not improve for three

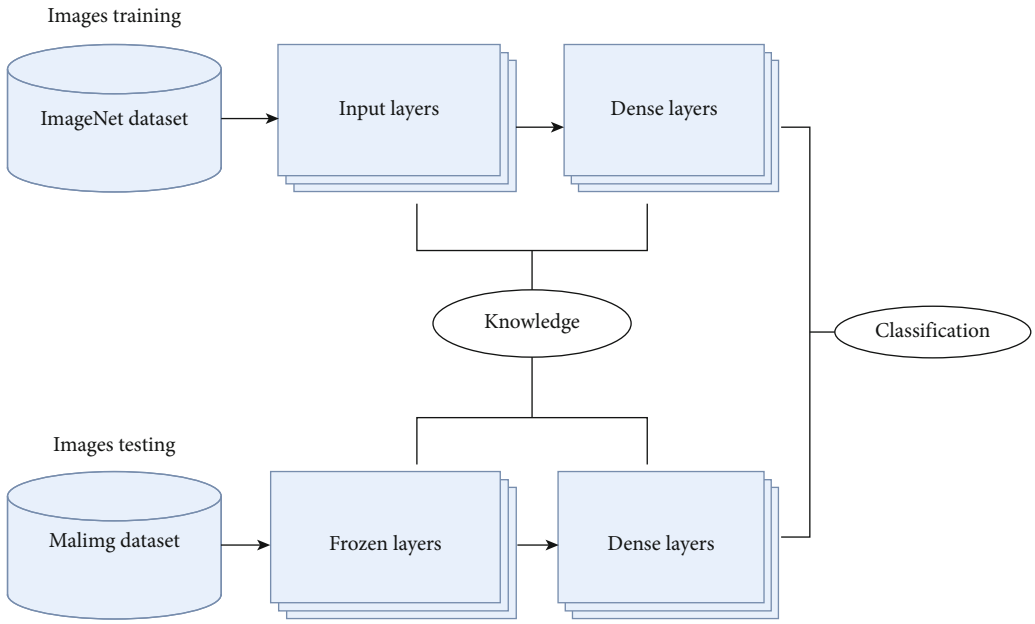

Figure 3: The approach of transfer learning.

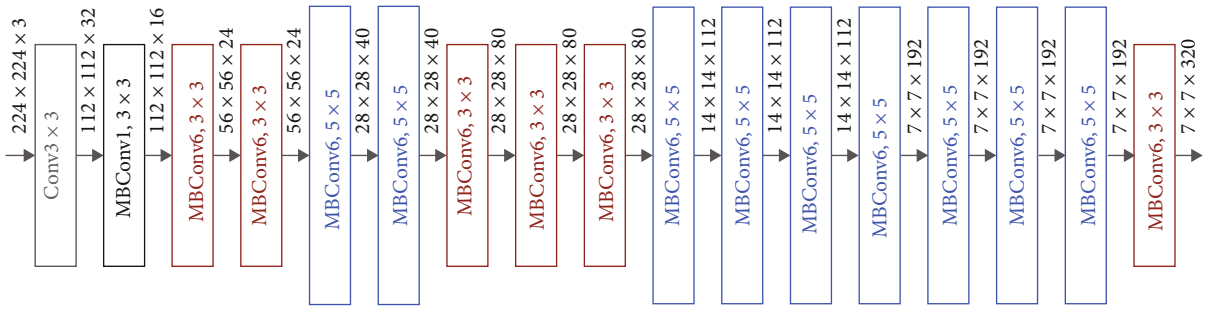

Figure 4: EfficientNet B0 architecture, by Tan and Le [34].

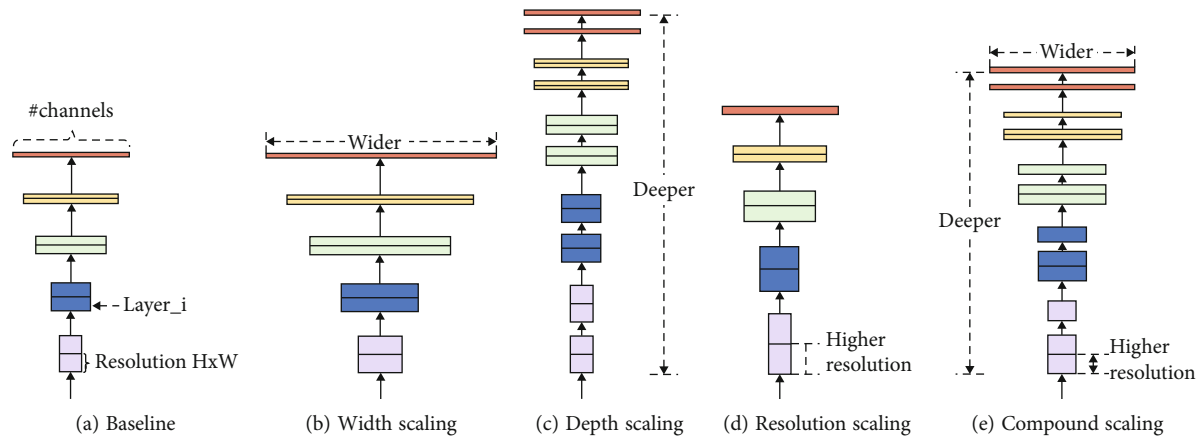

Figure 5: Compound scaling vs. other methods, from the original paper [34].

epochs in a row, we terminated Adam's training using an early stopping callback and set Adam's learning rate to 0.01, which is the lowest possible value. Figure 6 depicts the relative importance of the inputs and outputs at the three different levels: input 2, efficientnetb0, pooling, and dense.

The amount of time it takes to complete the training is determined by the batch size. Larger batch sizes result in more images being loaded into the random access memory (RAM) while the training is being completed. All of the models were trained with Adam's optimizer at a learning

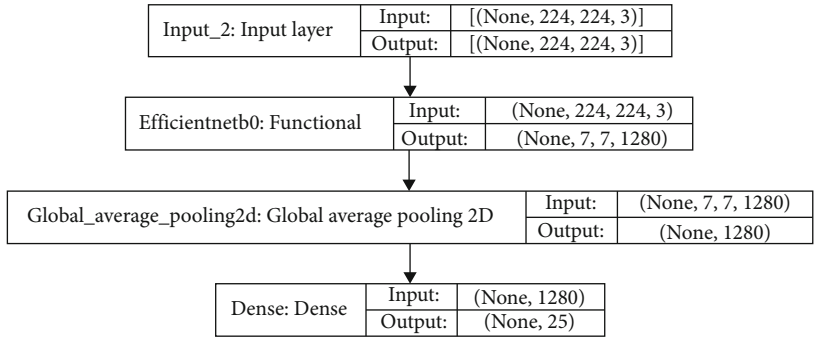

Figure 6: Final B0 model using transfer learning of EfficientNet weights.

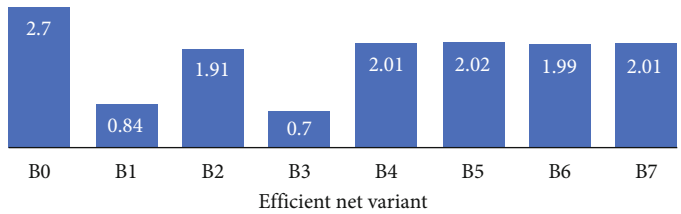

Figure 7: Accuracy error (100-accuracy) of EfficientNet variants.

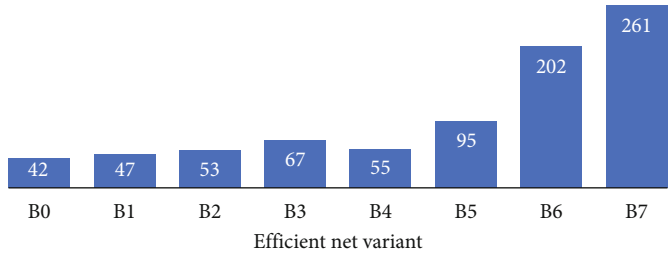

Figure 8: Training time in minutes of EfficientNet variants.

Table 3: Training time in minutes with accuracy error percentage and the inference time.

	Model	Inference time (ms)	Accuracy error %	Training time in minutes
0	***EffNet B0***	**4.91**	2.07	**42.0**
1	***EffNet B1***	5.55	0.84	47.0
2	***EffNet B2***	6.50	**0.03**	53.0
3	***EffNet B3***	8.77	0.70	67.0
4	***EffNet B4***	15.12	2.01	55.0
5	***EffNet B5***	25.29	2.02	95.0
6	***EffNet B6***	40.45	1.99	202.0
7	***EffNet B7***	61.62	2.01	261.0

rate of 0.001 and with an early stopping callback to stop the training when the model validation loss stopped decreasing for three consecutive epochs. As a result, the training time offers a reliable estimation of the amount of time required to complete one development iteration. In addition, the inference time is extremely important in the context of cybersecurity. An ideal model would provide an accurate diagnosis of the malware family in real time, allowing us to respond appropriately and limit the amount of damage that is caused.

We used the reported results of inference on ImageNet that was performed by the karas team, despite the fact that this study did not cover inference time in a real-time setup. The experiments were run on NVIDIA Tesla A100 GPU with a batch size of 32 and reported the inference time as the average of 30 batches and 10 repetitions. Figure 7 illustrates the results of training EfficientNet models from B0 to B7. We use accuracy error for visualization since the accuracy of all models is close to 100. Figure 8 shows the different times between all variants of EfficientNet models.

Table 3 provides information the training time in minutes, the accuracy error percentage, and the inference time in milliseconds. EfficientNet B3 records the best accuracy while EfficientNet B0 takes only 42.0 minutes to train to make it the fastest in terms of development time, as for inference time, EfficientNet B0 is by far the best recording only 4.91 MS per inference step.

5. Conclusion and Future Work

Deep learning methods are being utilized by multiple antivirus software packages for the purpose of malware

classification. This paper described the deep learning malware classification models that involve viewing and processing malware as images. Architectures based on deep learning are good at identifying and classifying malware. In this paper, grayscale images were classified according to different types of malware using EfficientNetB0-B7 CNN models. All models that will be used are pretrained models on ImageNet and Malimg datasets. The findings show that the EfficientNet model beats every additional effort. EfficientNet is a state-of-the-art performance in grayscale malware image categorization for malware detection.

Although an image processing-based approach to malware analysis is an excellent strategy based on global image-based features, an adversary who understands the technique can use countermeasures to defeat the system. To counter potential attacks, future studies will focus on more localized feature extraction approaches that account for the differences between malware executables and their primitive binary segments. Segmenting the image portions and characterization of the local texture of the texture patterns is one possible future expansion.

Data Availability

The data used in this research can be obtained from the corresponding authors upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

No Funds were received.

References

[1] S. Afzal, M. Asim, A. R. Javed, M. O. Beg, and T. Baker, "URLdeepDetect: a deep learning approach for detecting malicious URLs using semantic vector models," *Journal of Network and Systems Management*, vol. 29, no. 3, p. 21, 2021.

[2] S. ur Rehman, M. Khaliq, S. I. Imtiaz et al., "Diddos: an approach for detection and identification of distributed denial of service (ddos) cyberattacks using gated recurrent units (gru)," *Future Generation Computer Systems*, vol. 118, pp. 453–466, 2021.

[3] S. Mohurle and M. Patil, "A brief study of wannacry threat: ransomware attack 2017," *International Journal of Advanced Research in Computer Science*, vol. 8, no. 5, pp. 1938–1940, 2017.

[4] J. Hernandez-Castro, A. Cartwright, and E. Cartwright, "An economic analysis of ransomware and its welfare consequences," *Royal Society Open Science*, vol. 7, no. 3, p. 190023, 2020.

[5] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: a comprehensive review," *Neural Computation*, vol. 29, no. 9, pp. 2352–2449, 2017.

[6] A. R. Javed, M. Usman, S. U. Rehman, M. U. Khan, and M. S. Haghighi, "Anomaly detection in automated vehicles using multistage attention-based convolutional neural network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4291–4300, 2021.

[7] A. R. Javed, M. O. Beg, M. Asim, T. Baker, and A. H. Al-Bayatti, "Alphalogger: detecting motion-based side-channel attack using smartphone keystrokes," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–14, 2020.

[8] F. Shah, Y. Liu, A. Anwar et al., "Machine learning: the backbone of intelligent trade credit-based systems," *Security and Communication Networks*, vol. 2022, Article ID 7149902, 10 pages, 2022.

[9] S. S. Ullah, S. Hussain, A. Gumaei, and H. AlSalman, "A secure NDN framework for Internet of Things enabled healthcare," *Computers, Materials & Continua*, vol. 67, no. 1, pp. 223–240, 2021.

[10] S. Hussain, S. S. Ullah, M. Uddin, J. Iqbal, and C. L. Chen, "A comprehensive survey on signcryption security mechanisms in wireless body area networks," *Sensors*, vol. 22, no. 3, p. 1072, 2022.

[11] S. Hussain, S. S. Ullah, I. Ali, J. Xie, and V. N. Inukollu, "Certificateless signature schemes in Industrial Internet of Things: a comparative survey," *Computer Communications*, vol. 181, pp. 116–131, 2022.

[12] J. Iqbal, M. Adnan, Y. Khan et al., "Designing a healthcare-enabled software-defined wireless body area network architecture for secure medical data and efficient diagnosis," *Journal of Healthcare Engineering*, vol. 2022, Article ID 9210761, 19 pages, 2022.

[13] H. Salman, J. Grover, and T. Shankar, *Hierarchical Reinforcement Learning for Sequencing Behaviors*, vol. 2733, pp. 2709–2733, 2018.

[14] A. S. Parihar, S. Kumar, and S. Khosla, "S-DCNN: stacked deep convolutional neural networks for malware classification," *Multimedia Tools and Applications*, vol. 81, no. 21, pp. 30997–31015, 2022.

[15] L. Nataraj, V. Yegneswaran, and P. Porras, "A comparative assessment of malware classification using binary texture analysis and dynamic analysis categories and subject descriptors," *4th ACM Workshop on Security and Artificial Intelligence*, 2011.

[16] ImageNetMarch 2022, https://image-net.org/.

[17] S. Miyawaki, E. A. Hoffman, and C. L. Lin, "Effect of static vs. dynamic imaging on particle transport in CT-based numerical models of human central airways," *Journal of aerosol science*, vol. 100, pp. 129–139, 2016.

[18] A. Çayır, U. Ünal, and H. Dağ, "Random CapsNet forest model for imbalanced malware type classification task," *Computers & Security*, vol. 102, p. 102133, 2021.

[19] K. Kosmidis and C. Kalloniatis, "Machine learning and images for malware detection and classification," in *Proceedings of the 21st Pan-Hellenic Conference on Informatics*, Larissa, Greece, 2017.

[20] L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, "Malware images: visualization and automatic classification," in *Proceedings of the 8th international symposium on visualization for cyber security*, Pittsburgh, Pennsylvania, USA, 2011.

[21] A. Torralba, K. P. Murphy, W. T. Freeman, and M. A. Rubin, "Context-based vision system for place and object recognition," in *Proceedings Ninth IEEE International Conference on Computer Vision*, Nice, France, 2003.

[22] A. Makandar and A. Patrot, "Malware class recognition using image processing techniques," in *2017 International Conference on Data Management, Analytics and Innovation (ICDMAI)*, pp. 76–80, Pune, India, 2017.

[23] L. Alzubaidi, J. Zhang, A. J. Humaidi et al., "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *Journal of big Data*, vol. 8, no. 1, pp. 1–74, 2021.

[24] Z. Cui, F. Xue, X. Cai, C. Yang, G.-g. Wang, and J. Chen, "Detection of malicious code variants based on deep learning," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. 3187–3196, 2018.

[25] M. Kalash, M. Rochan, N. Mohammed, N. Bruce, W. Yang, and F. Iqbal, "Malware classification with deep convolutional neural networks," in *9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pp. 1–5, Paris, France, 2018.

[26] R. Ronen, M. Radu, C. Feuerstein, E. Yom-Tov, and M. Ahmadi, "Microsoft malware classification challenge," 2018, https://arxiv.org/abs/1802.10135.

[27] D. Gibert, C. Mateu, J. Planes, and R. Vicens, "Using convolutional neural networks for classification of malware represented as images," *Journal of Computer Virology and Hacking Techniques*, vol. 15, no. 1, pp. 15–28, 2019.

[28] S. Kumar, "MCFT-CNN: malware classification with fine-tune convolution neural networks using traditional and transfer learning in Internet of Things," *Future Generation Computer Systems*, vol. 125, pp. 334–351, 2021.

[29] M. Schofield, G. Alicioglu, R. Binaco et al., "Convolutional neural network for malware classification based on API call sequence," in *Proceedings of the 8th International Conference on Artificial Intelligence and Applications (AIAP 2021)*, Zurich, Switzerland, January 2021.

[30] W.-C. Lin and Y.-R. Yeh, "Efficient malware classification by binary sequences with one-dimensional convolutional neural networks," *Mathematics*, vol. 10, no. 4, p. 608, 2022.

[31] B. Kolosnjaji, A. Demontis, B. Biggio et al., "Adversarial malware binaries: evading deep learning for malware detection in executables," in *2018 26th European signal processing conference (EUSIPCO)*, pp. 533–537, Rome, Italy, 2018.

[32] A. Bensaoud, N. Abudawaood, and J. Kalita, "Classifying malware images with convolutional neural network models," *International Journal of Network Security*, vol. 22, no. 6, pp. 1022–1031, 2020.

[33] L. Alzubaidi, O. al-Shamma, M. A. Fadhel, L. Farhan, J. Zhang, and Y. Duan, "Optimizing the performance of breast cancer classification by employing the same domain transfer learning from hybrid deep convolutional neural network model," *Electronics*, vol. 9, no. 3, p. 445, 2020.

[34] M. Tan and Q. Le, "Efficientnet: rethinking model scaling for convolutional neural networks," in *International conference on machine learning*, Long Beach, California, 2019.