Journal of Advanced Zoology

ISSN: 0253-7214

*Volume **43** Issue 1 **Year 2022** Page **979-989***

Malware Detection Using Tlenet On Image Data

V S Jeyalakshmi[1*], Krishnan Nallaperumal[2]

[1]Research Scholar, Centre for Information Technology and Engineering, Manonmaniam Sundaranar University, Abishekapatti, Tirunelveli-627012, Tamil Nadu, India E-mail: vsjeyalakshmiap@gmail.com*
[2]Senior Professor and Head, Centre for Information Technology and Engineering, Manonmaniam Sundaranar University, Abishekapatti, Tirunelveli-627012, Tamil Nadu, India E-mail: krishnan17563@gmail.com

Corresponding Author: V S Jeyalakshmi
**Research Scholar, Centre for Information Technology and Engineering, Manonmaniam Sundaranar University, Abishekapatti, Tirunelveli-627012, Tamil Nadu, India E-mail: vsjeyalakshmiap@gmail.com*

The prevalence of malicious software is referred to as malware and has seen a notable increase today. It is a significant threat to the overall integrity of internet security in modern time is significant. Malware is a dangerous threat to the whole internet users due to its unauthorized data acquisition and inflict damage upon computer systems. Malware detection has gathered a significant attention within academic fields due to the growing prevalence of malware. Threats pose risks to the individual computer users, corporations and governmental entities, etc., to detect the unidentified malware in real-time is difficult. The malware detection systems depend on the examination of the malware signatures and behavioral patterns is by the combination of dynamic and static analysis. The problem is to identify and categorise the malicious software by image analysis techniques. Deep learning (DL) are used for image recognition after the conversion of executable files into image formats in the mailimg and ImageNet datasets. Transfer learning is used to train deep learning models for large-scale datasets. The problem within this method is slow and laborious. In pursuit of the objective, the study investigates the implementation of deep convolutional neural networks to classify the malware. Additionally, it provides methodologies for effectively using transfer learning techniques to compromise the challenges associated with detecting and categorizing the different types of malware. To enhance the classification of malware, a pre-trained convolutional neural network (CNN) in transfer learning is used to categorize the malware images from the Malimg and ImageNet datasets are proposed. The Malimg dataset consists of malicious programs in the form of images are classified using the portable executable files. EfficientNet3 classifier has scored an impressive accuracy of 99.60 percentage.

Keywords: Malicious Software, Deep Learning, Convolutional Neural Networks, classifier, Malimg, Imagenet.

I. INTRODUCTION

Malware, a malicious software involves any computer programs that are intentionally designed to carry out harmful activities. The prevalence of such programs has significantly exaggerated in recent years, owing to the widespread possession of many devices makes the malware families are susceptible to attacks. Therefore, it is crucial to develop efficient and dependable methodologies for the detection and mitigation of emerging malware[1]. Contemporary instances of malware are equipped with defensive measures to avoid study of computer code through backward attacks. Viruses are often focus to have the modifications like obfuscation techniques [2]. The syntactic code transformations include the process of taking input programme and generate an improved version with more features are giving complexity for analysis [3, 4]. Combining methods of concealment with code optimization are including in compilers for the investigation of malware. The frequent results provides slowdown or obstruction of the disassembly procedure [5, 6]. Therefore, it is crucial to identify methodologies to address the fundamental binary data rather than depending on derived characteristics from reverse engineering endeavors.

To categorize the new malware into their respective families is referred to as malware classification and it is accomplished by the use of deep learning methodologies. The variety of deep learning models are available than machine learning models for making the transition from shallow to deep models, whereas machine learning needs human feature development before training can begin and deep learning models can work with raw data without any preprocessing. One drawback of shallower algorithms is that they need domain expertise, which means extra time and effort must be spent for analyzing the dataset samples earlier the learning phase can begin. The rapid proliferation of new viruses has resulted in an impractical expense associated with human work [7, 8]. Features may be extracted from database samples without the need for traditional feature engineering or domain expertise by deep learning techniques. The above-mentioned study establishes deep learning as the preferred model for the classification of malware [9]. One limitation associated with deep learning approaches in contrast to shallower designs is the inclination to exhibit overfitting behavior when trained using small size datasets. The issue may provide a challenge in several domains such as program analysis and particularly malware classification due to the significant resources and time required to collect an adequate number of samples with accurate ground-truth information [10]. The issue is prevalent in several other domains including recognizing images and sorting them into categories [11, 12]. Concerns about a lack of appropriate information for training may be effectively addressed within the computer vision, it is possible to produce additional data points by performing semantics-preserving changes to the existing images. The transformations may include manipulations like rotation, spatial interpreting, and cropping. The practice of producing more data by manipulating existing images is often referred to as data augmentation is a fundamental technique in deep learning. This is an important component in the methodology to discover the malware patterns crisply [13].

The creation of malware has a significant increase in recent times and presenting a serious risk to the security of enterprises, organizations and community. To mitigate the malware spread, it is essential to develop novel approaches for promptly detecting and categorizing malware instances as well as the investigation of their behavioral patterns. While AI-based techniques in the malware categorization are gaining grip, it's important to remember that the vast majority of currently-used malware categories are quite straightforward. Moreover, conventional artificial intelligence methods need significant resource allocation for building the features manually. Convolutional neural networks (CNNs) have shown enhanced accuracy in the field of malware image classification using very effective tools and techniques. In comparison to the conventional learning approaches, CNNs have exhibited superior performance [14].

As a result of computer vision tasks, convolutional neural networks (CNNs) are generating considerable interest in the field of deep learning. Video analysis, obstacle detection for autonomous vehicles, natural language processing, localization and segmentation problems are some examples in computer vision. In contrast to the feature extraction manually, transfer learning is used to classify distinct families of malware automatically. To do this, a pre-trained convolutional neural network (CNN) framework is used. Both the Malimg and ImageNet classification datasets were used in the experiment [15, 16]. The objective of the work is to focuses on deep convolutional neural networks with transfer learning in malware family classification. A total of 9,340 malware samples were categorized by the EfficientNet deep learning models in Malimg and Imagenet datasets. Furthermore, Convolutional Neural Network pretrained models to enable the transfer learning for malware classification [17,18].

II. Related works

In computer security, particularly the identification of specific kind of dangerous software is the intended safeguard for the society. The phenomenon of developing the diverse strategies of various virus families by hackers are also increasing [19]. The categorization of malware types have often done in terms of three regular approaches such as static, dynamic and image-based. Static analysis does not need the active execution of a binary code to get information from it. Dynamic analysis used to examine the dangerous software by actively analyzing its behavior in an appropriately controlled setting in real time. The malware categorization through images is a significant and growing area of research and implementation. Deep learning is a significant driving force for the investigation of artificial intelligence combination of processing images and other visual methods. Many deep convolutional neural networks (DCNNs) have been developed for use in the area of processing images have shown encouraging outcomes. In the first phases of malware classification, a method applied for the classification of gray-scale images. The raw executable malware files may be analyzed to derive grayscale representations of their harmful features is potentially feasible. Malware analysis may be conducted by extracting visual components from malware images.
Deep learning technique is widely used for the analysis of vast quantities of information. Artificial neural networks (ANNs) algorithms are at the heart of the deep learning field's to facilitate the success in the training of computer machines to achieve ongoing growth and development. It involves the ability to categorize and classify data and images analogous to the functioning of human brain. Academicians have proposed the use of Convolutional Neural Networks (CNNs) in the field of categorizing malicious software. Cui et al. used Convolutional Neural Networks (CNN) to do an in-depth analysis of the code variations [20]. The codes were deciphered by transforming them into black and white images. Makandar and Patrot use the wavelet transform to create very effective feature vector based texture analysis of malware images. The virus was then categorized by using several classes of support vector machines where the input consisted as malware images. Both its complexity and the number of dimensions of the feature characteristics were decreased [21]. Torralba et al. used the methodology of extracting GIST features from grayscale images and classify them using the distance calculated by Euclidean geometry as a metric to establish the correlation between the two variables. Nevertheless, the methodology used by the researchers incurs a substantial computational burden. Nataraj et al. used a dataset containing a collection of 9,342 distinct malware images, each representing one of 25 distinct forms of malware [22]. The aforementioned people were exploring the possibility of employing byte graphs as grayscale images for the automated malware classification [23].

III. Dataset Description

To categorize the malware images using transfer learning particularly EfficientNets on the Malimg and ImageNet datasets, the transformation of malware from binary code into image files is an important part of malware classification. MalImg dataset comprises a comprehensive collection of 9400 samples of malicious software which are classified into 25 separate malware family. Malicious software feature samples are not have shown in the image formats discovered on the storage media. The bytes that constitute the executable files are directly transformed into floating-point values. Subsequently, the numerical values are construed as pixel intensities that correspond to grayscale images.
As predicted, there is a notable imbalance seen among the classifications in the sample. The class with the highest number of counts are 'Allaple. A', has a total of 2940 samples, whereas the class with the lowest number of 75 malware samples are Skintrim.N. Figure 1 illustrates a visual depiction of eight samples have been randomly picked from two separate classes within the dataset. Differentiating between samples from each family is made easier by the distinctive structures shown by the images from each class [25]. The validity of the work persists for data have been subjected to bicubic interpolation, as seen in Figure 2. The number of identified malware variants is shown in Table 1.

Malware Families	Malware Variant	Malware Counts	Malware Families	Malware Variant	Malware Counts
Adialer.C	dialer	122	Lolyda.AA1	passwordstealer	213
Agent.FYI	bdr	116	Lolyda.AA2	passwordstealer	184
Allaple.A	worm	2,940	Lolyda.AA3	passwordstealer	123
Allaple.L	worm	1,591	Lolyda.AT	passwordstealer	159
Alueron.gen!J	trojan	198	Malex.gen!J	trojan	136

Autorun.K	worm	106	Obfuscator.AD	Troj-dlr	142
C2LOP.gen!g	trojan	200	Rbot!gen	bdr	158
C2LOP.P	trojan	146	Skintrim.N	trojan	75
Dialplatform.B	dialer	177	Swizzor.gen!E	Troj-dlr	128
Dontovo.A	Troj-dlr	162	Swizzor.gen!I	Troj-dlr	132
Fakerean	rogue	381	VB.AT	worm	408
Instantaccess	dialer	431	Wintrim.BX	Troj-dlr	97

Table 1: Malware types and its count

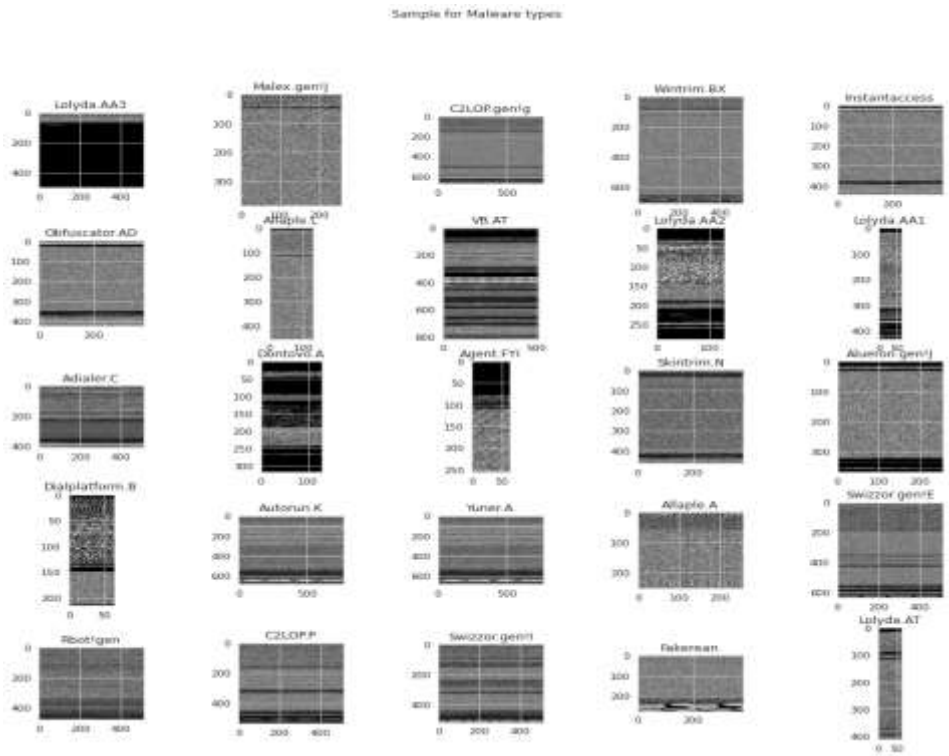

Fig. 1. Malware Types

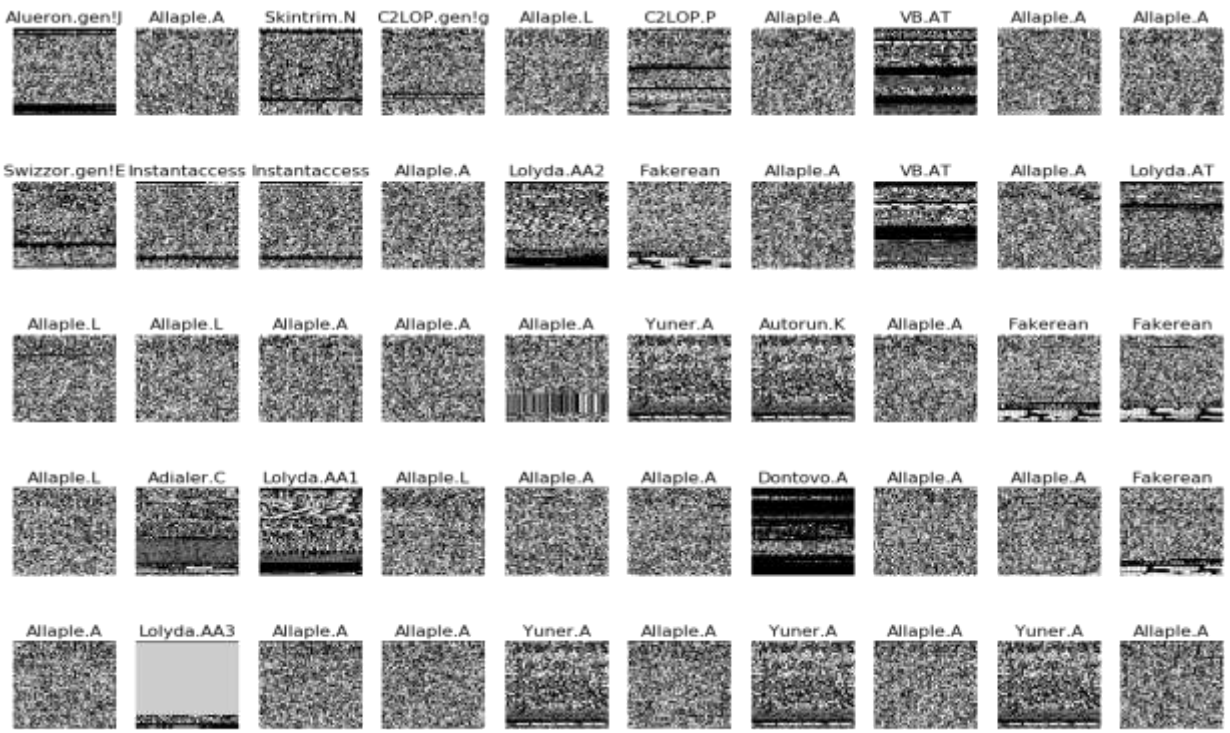

Fig. 2. Malware Binary Image samples

The research discovered the visual representation of gray scale malware images from binary files increasingly appear. Images may be analysed for patterns using deep learning. It is possible to identify related malware by understanding the features across malware images. To identify the meaningful patterns through a network of deep learning algorithms to categorize the group of malicious programs by segmenting a same malicious belonging characteristics to an image.

Deep learning models have shown significant efficacy in the image classification to extract features from within an image using processes such as subsampling, pooling and various other computational techniques. Consequently, CNN models has proven to be especially advantageous in the field of image classification. Convolutional Neural Networks (CNNs) are used in the process of classification to identify the fundamental components of an image that belongs to a certain malware family [26]. Figure 2 illustrates a method for changing a binary representation into a sequence of 8-bit vectors or hexadecimal symbols. The process enables the transformation of malware binaries into visual representations. Figure 3 illustrates the representation of an 8-bit vector encompasses numerical values ranging from 0 to 255. The 8-bit vectors are numerical representations in the malware images that may be converted into pixels of gray scale malware images.

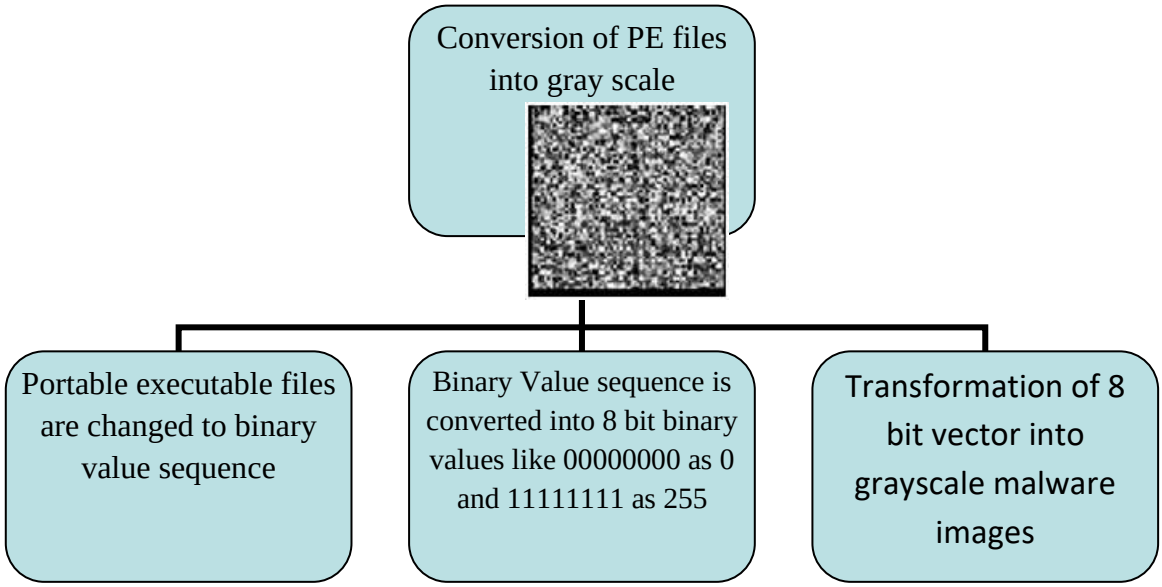

Fig. 3. Malware Binaries to Gray scale Images

IV. Proposed Methodology

Transfer learning refers to a computational technique in deep learning that enables the adaptation of a pre-existing model which has been trained on a specific task to a different and distinct task [27]. Nevertheless, the process of training deep neural networks is time-consuming and requires significant computer resources. Therefore, the primary motivation for using transfer learning is to use existing knowledge and aids. In summary, the EfficientNets pre-trained models trained on the ImageNet dataset are used for reutilization as tested on the Malimg dataset to classify malware families. The implemented technique of transfer learning is shown in Figure 4.

The convolutional neural network architecture and the scaling method are referred to as EfficientNet which was a collection of models (B0 to B7) to demonstrate the remarkable performance in terms of accuracy and efficiency over a broad spectrum of scales ranging from small too big, scaling up from B0 to B7 [28]. The architectural structure of the fundamental model B0 is seen in Figure 5.

EfficientNet methodology uses the compound scaling process starting from B0, where the methodology entails simultaneously adjusting the input parameter size, layers length, dimension and the number of channels based on a predetermined formula. Figure 5 depicts the architecture, whereby the normal scaling (b-d) only amplifies the network's breadth, depth or resolution. To precise three-dimensional scaling utilising a constant ratio, the compound scaling strategy(e) is recommended. The process of determining the appropriate depth, breadth and resolution for each gives good classification accuracy. The EfficientNet model has been undertaken with careful deliberation and these parameters have shown the potential outstanding outcomes. To comply with the standards indicated in the EfficientNet architecture (see figure 5) the malware images

undergoes resize for each model to align with the prescribed dimensions to extract the features of malicious content in the image.

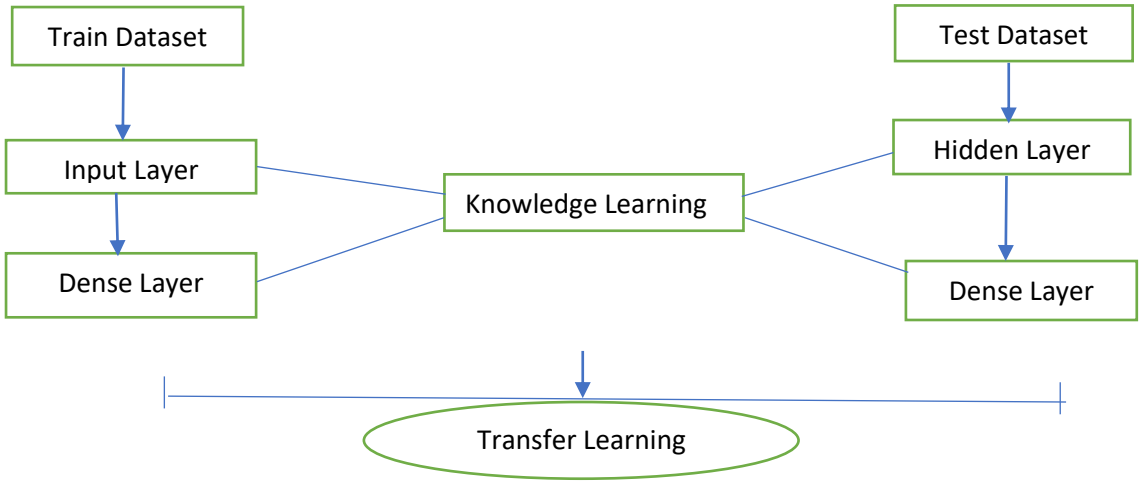

Fig. 4. Proposed Arcitecture

Fig. 5. EfficientNet [28]

A. Experimental Study

A comprehensive account of the experiments includes repeatability and generalizability, model design and strategies for effectively presenting the obtained data. To save time and effort, employ trained EfficientNet models on the ImageNet dataset. The classification layer is then stacked on top by a dense (or completely linked) architecture of 20 output values and SoftMax activations. Training EfficientNet has shown a time-consuming procedure but sometimes extending many days when executed on a high-performance NVIDIA graphics processing unit (GPU). A total of 32,000 distinct parameters may be subjected for training. The models were trained using Adam's optimizer. When the validation loss of the model did not exhibit any improvement for two consecutive epochs. Adam optimizer using an early stopping callback and assigned a learning rate of 0.01 with the minimum attainable rate of efficientnetb0 is shown in Figure 6.

B. Dimension for Train, Test and Validation Models

In the next part, the physical dimensions of the images being examined for classification have been determined as 64×64 and 256×64. The objective of the optimization is to increase the efficacy of training duration and boost the precision of classifying malware for both the models. The training procedure involves evaluating the accuracy of various image sizes resulting in the development of multiple time step quantities and features. The empirical results indicate the selected values exhibit high levels of accuracy in classification, positioning them as optimal choices.

To ensure a certain level, the datasets were split into training, validation and testing batches to ensure adequate generalization. While the first phases of the learning process were conducted many experiments to determine the optimal ratios for the splits. Our objective was to achieve maximum fairness in generalization while also ensuring that a significant number of samples were retained for the period of training. In the tests carried out here, a hold-out test collection having a ratio of 0.2 is used as 20% of the information being tested is kept aside for the testing stage alone and it is not utilized for parameter adjustment throughout the training process. A portion of the sample pool that remains in the dataset is the percentage of data used to validate the model is 10%. The validation loss is calculated using the set at the end of each training iteration. The rest of the data as a "training set," which is utilized to train the network's weights. The findings obtained were derived employing the last remaining test group. The findings were averaged after 20 cycles of testing. The inclusion of numerous random divides in the dataset helps to enhance the generalizability of the technique to minimize the biased division.

C. Models

Both models under consideration have been implemented in Python using the Keras library and have been made accessible on Jupyter notebooks which are freely available online. This has been done to facilitate the research with the MalImg datasets to the algorithms. A network of CNN is implemented using the Keras framework using the TensorFlow library in its basic form. Each convolution layer in the network will be followed by a two maximum pooling layers. There are five convolutional layers in the model, and each one has a kernel with a count of 5 and an activation function of the rectified linear unit (ReLU). The first layer is composed of 32 filters, while the second layer consists of 64 filters. To avoid downsizing the input image, set the padding value of both max-pooling layers to "same" and make them 2x2. A dense layer serves as the network's "brain," followed by a layer of dropouts that is linked to the final compact layer. The experiments were run with an initial batch size of 30 and an NVIDIA Tesla GPU. A standard deviation of 25 batches was used to determine the inference time stated, with each batch repeated 15 times. Figure 7 displays the results of training EfficientNet models covering ranges B0–B7. Since the precision of all models is very near to 100, the accuracy error is utilised for representational purposes only. As seen in Figure 8, varying time intervals among the various iterations of EfficientNet models see Table 2.

Model	Classification types	Malware Type	OS	ML / DL Techniques	Layers	Accuracy Value
Bo, B1	Binary	Mailing dataset	WINDOWS 10 / 11	EfficientNet	32 x 32	98.00 %
B2	Binary	Mailing dataset	WINDOWS 10 / 11	EfficientNet	64 x 32	95.00 %
B3, B4	Binary	Mailing dataset	WINDOWS 10 / 11	EfficientNet	64 x 128	97.61 %
B5, B6	Binary	Mailing dataset	WINDOWS 10 / 11	EfficientNet	128 x 64	92.20 %
B7	Binary	Mailing dataset	WINDOWS 10 / 11	EfficientNet	64 x64	

Table 2. EfficientNet Model Accuracy List

V. Results

The experiments using various malware image sizes and the methodology such as zero-padding and cropping to resize the images using interpolation. EfficientNet-CNN has been implemented in the Keras framework. The malware is predicted to binary images (see Figure 6 and 7). The architecture comprises 151 and 84 units in the network to built an impatience period of 25 epochs to help prevent overfitting. This suggests that the education process of the model will cease after 25 epochs if there is no further decrease in the validation loss (see Figure 8 and 9).

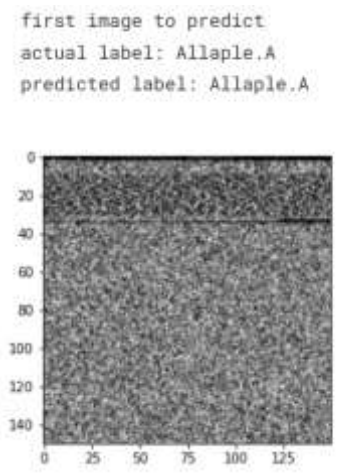

Fig. 6. Malware Image Prediction

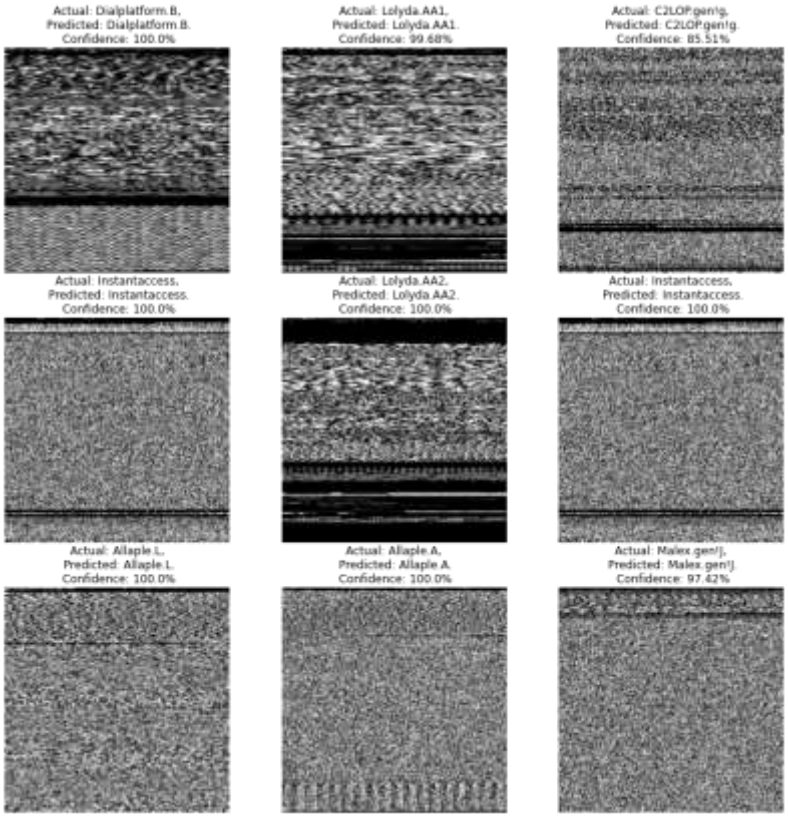

Fig. 7. Prediction of Malware Binary Images

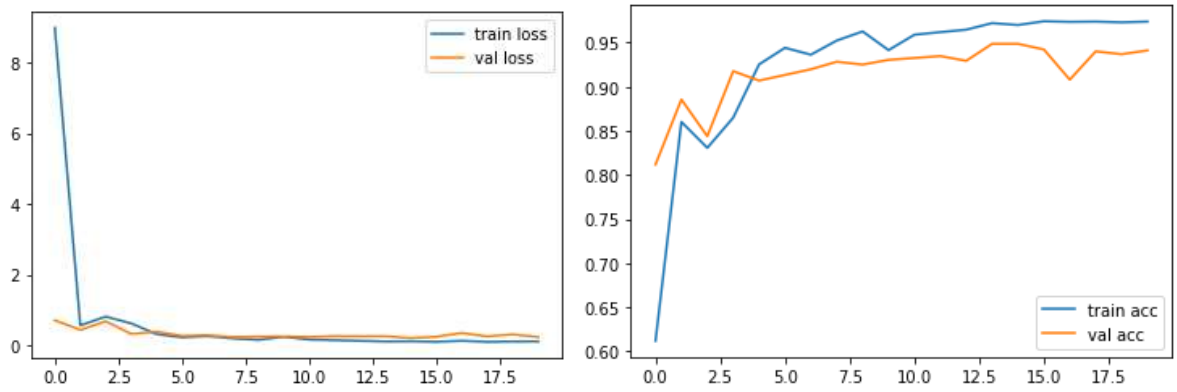

Fig. 8. Train and Test - Accuracy and Loss

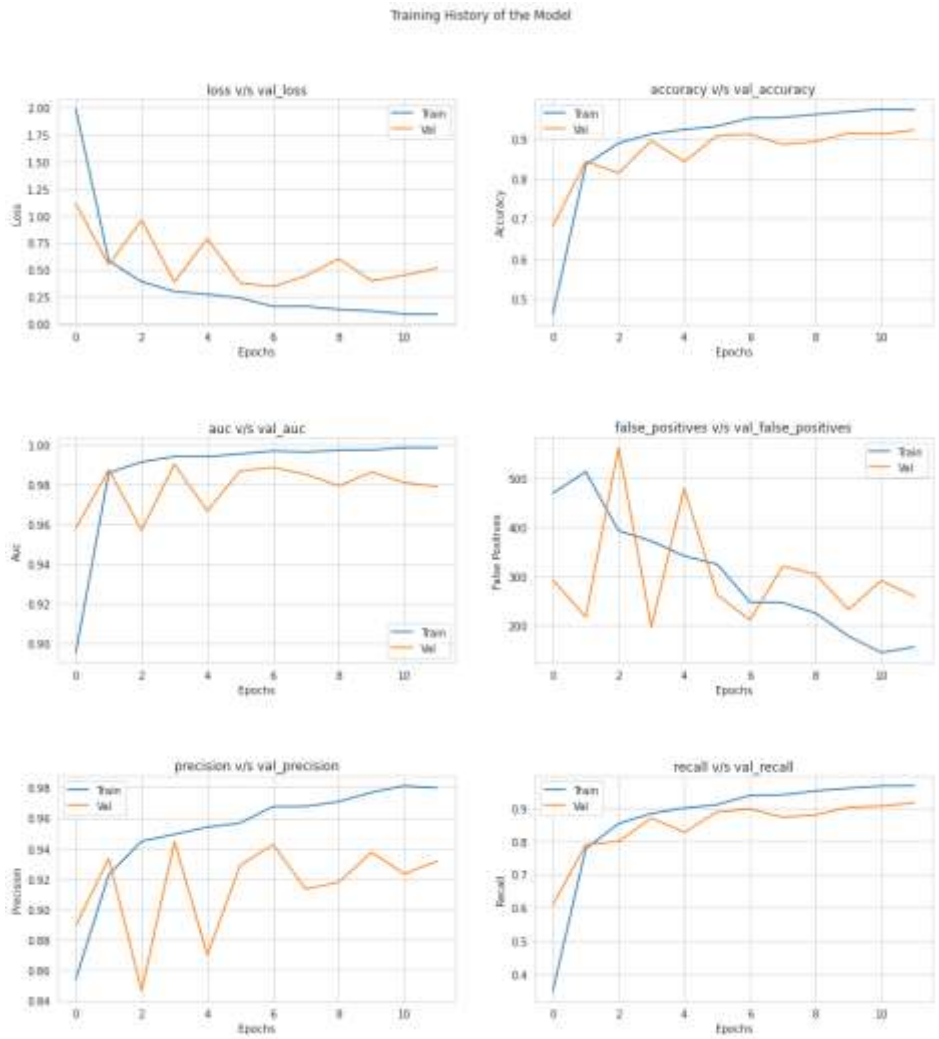

Fig. 9. Recall, Precision, F1-Score and Accuracy Plots

VI. Conclusion and future work

Several antivirus packages are using deep learning approaches to classify malware. The study presents an analysis by employing DL-based classification algorithms to focus on the approach of treating malware as images and using image processing techniques. Architectures that use deep learning techniques have shown proficiency in the identification and classification of malware. EfficientNetB0-B7 convolutional neural network models are used to classify the grayscale images into several categories of malware. The models used are pre-trained on both the Datasets like ImageNet and Malimg. According to the findings, the EfficientNet model is superior than the competition. To classify grayscale malware images, EfficientNet is a groundbreaking innovational performance in malware detection. While using an image processing methodology for malware analysis proves to be a commendable approach that relies on comprehensive image-based characteristics to acknowledge an adversary possessing and thorough understanding of the technology might use countermeasures to undermine the efficacy of the system.

To mitigate possible security breaches, research endeavors will prioritize the exploration of localized feature extraction methodologies that effectively consider the distinctions among malware program files and the basic binary segments. Future research is the segmentation of malware executable patterns and the characterization of local texture patterns in the malware-images.

VII. References

1. McAfee: McAfee Labs Threats Report 2020. https://www.mcafee.com/enterprise/en-us/assets/reports/rp-quarterly-threats-nov 2020.pdf; 2020

2. Schrittwieser, S., Katzenbeisser, S., Kinder, J., Merzdovnik, G., Weippl, E.: Protecting software through obfuscation: Can it keep pace with progress in code analysis? ACM Comput. Surv. (CSUR) 49(1), 1–37; 2016.
3. Collberg, C., Thomborson, C., Low, D.: A taxonomy of obfuscating transformations; 1997.
4. Andriesse, D., Chen, X., Van Der Veen, V., Slowinska, A., Bos, H.: An in-depth analysis of disassembly on full-scale x86/x64 binaries. In: 25th USENIX Security Symposium (USENIX Security 16), pp. 583–600; 2016.
5. Shorten, C., Khoshgoftaar, T.M.: A survey on image data augmentation for deep learning. J. Big Data 6(1), 60; 2019.
6. Perez, L., Wang, J.: The effectiveness of data augmentation in image classification using deep learning. arXiv preprint arXiv:1712.0462; 2017.
7. Marastoni, N., Giacobazzi, R., Dalla Preda, M.: A deep learning approach to program similarity. In: Proceedings of the 1st International Workshop on Machine Learning and Software Engineering in Symbiosis, pp. 26–35; 2018.
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. 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.
14. 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.
15. ImageNetMarch 2022, https://image-net.org/.
16. 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.
17. A. Çayır, U. Ünal, and H. Dağ,"Random CapsNet forest model for imbalanced malware type classification task," Computers & Security, vol. 102, p. 102133; 2021.
18. K. Kosmidis and C. Kalloniatis, "Machine learning and images for malware detection and classification," [Proceedings of the 21st Pan-Hellenic Conference on Informatics, Larissa, Greece]; 2017.
19. L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, "Malware images: visualization and automatic classification," [Proceedings of the 8th international symposium on visualization for cyber security, Pittsburgh, Pennsylvania, USA]; 2011.
20. A. Torralba, K. P. Murphy, W. T. Freeman, and M. A. Rubin, "Context-based vision system for place and object recognition," [Proceedings Ninth IEEE International Conference on Computer Vision, Nice, France]; 2003.
21. A. Makandar and A. Patrot, "Malware class recognition using image processing techniques," [International Conference on Data Management, Analytics and Innovation (ICDMAI), pp. 76–80, Pune, India]; 2017.
22. 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.
23. 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.
24. R. Ronen, M. Radu, C. Feuerstein, E. Yom-Tov, and M. Ahmadi, "Microsoft malware classification challenge," [https://arxiv.org/abs/1802.10135]; 2018.
25. 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.

26. 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.
27. 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.
28. M. Tan and Q. Le, “Efficientnet: rethinking model scaling for convolutional neural networks,” [International conference on machine learning, Long Beach, California]; 2019.