# Visual Malware Detection by Deep Learning Techniques in Windows System

Hussein Almusawi
*Department of Computer Engineering*
*University of Karabuk*
Karabuk, Turkey
2028150034@ogrencikarabuk.edu.tr

***Abstract*— The number of malwares is increasing dramatically day by day with the development of modern technologies such as the Internet and electronic banks, so we need advanced technology to detect this malicious software more effective than anti-virus programs that rely on the signature of malicious software, which have proven its failure in some cases. In this study, the deep learning technique which is one of the branches of artificial intelligence through Convolutional Neural Network (CNN) has been proposed and applied to dataset called Malimg which consist of 25 families and 9339 samples of malware grey scale images, and these images converted from malware binary files. The result of our method has proven its efficiency by obtaining an accuracy of 96.76% in malware detection.**

***Keywords—malware, convolutional neural network, deep learning, malware image, CNN***

## I. INTRODUCTION

Malware is malicious software such as viruses, Trojan, worms, spyware, and other types that are manufactured with a specific purpose such as destroying devices, spying, stealing accounts, or other malicious activities. Malware is considered one of the most important dangers to the internet's security, and it has been found that this malicious software increases dramatically in the last ten years, as shown in Fig.1 [1], and 430 million malicious programs were discovered in 2015 and they increased by 36% over the previous year [2]. It has also started to become more dangerous in recent years, and one of the most dangerous examples of malware is the WannaCry ransomware worm, which appeared on 12 May 2017, and infected more than 10 thousand individuals and 200 thousand organizations [3]. So we need a very effective and accurate way to detect harmful files, and the best way at the moment is deep learning, which is one of the branches of artificial intelligence, and there are several algorithms in this field.

Nowadays Convolutional Neural Network (CNN) is one of the deep learning algorithms that has become popular in numerous computer works and is effective use of information in a wide range of fields, multiple building blocks like convolutional layer, pooling layer and fully connected layer make up a CNN [4]. It is possible to extract features and classify images by category. To make deep learning technology detect malware, we first train it according to a model called Dataset. There are several methods in deep learning to detect malware, such as opcode serial, API call serial, and image samples [5].

The signature-based methods for detecting malware used by antivirus companies cannot identify attacks for which no corresponding signature is recorded in the repository [6]. This causes a problem in computers being exposed to a lot of new malwares, and another problem that is difficult to solve is how to classify malicious software into the same family (class).

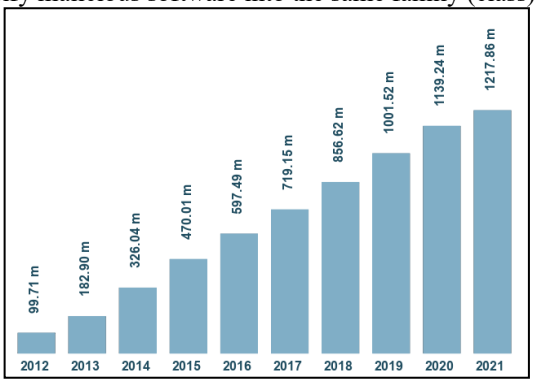


Fig. 1. Last 10 Years Malware Statistics

In this study, we focus on the application of CNN techniques in malware images, which is a highly effective technique for manipulating images of malware samples after converting them from binary files of each type. Firstly, the studies related with our method that use different algorithms to detect malware were explained. The dataset, structure of the proposed method and the experimental result of our model are described. The model was tested, and the results were discussed.

The Proposed system to detect malware that combined malware image as well as deep learning. Initially, gray scale images of the benign and malware were generated. After that to identify Malicious software, used deep learning technique by CNN algorithm to extract features from the images. This model has three convolutions layers, three pooling layers, two fully connected layers and before that the dataset images were resized to 32*32 pixel [5].

The cuckoo sandbox has been used to apply malware identification by deep learning based on API calls. Cuckoo sandbox was constructed by the researchers of [7] in order to extract the malicious code runtime's API call data. For malicious code identification, natural language processing systems are combined with context information from API call information. For malware identification, the BLSTM approach was suggested as a deep learning technique. [7].

A malware structure system has been proposed in [8] to convert malware into colour image and use CNN to classify the image representation. Spatial pyramid pooling layers (SPP) with CNN have been used to deal with different size inputs [8].

Implemented two algorithms for deep learning, CNN and Recurrent Neural Network (RNN), a method for analysing static malicious software using visual images have been

proposed and in this experiment, RNN is used to link data with concerned information and enhance anti interference capability and To extract features from images, CNN has been used [9].

The proposed DenseNet model (Densely Connected Convolutional Networks) is one of the most recent neural networks for object detection, it was implemented on four data sets(which have big samples), three for training and the fourth for testing, to significantly increase malware classification efficiency by addressing data imbalance issues [10]. It was discovered that Extreme Learning machine (ELM) and CNN perform almost as well together when trained on one-dimensional data as they do when trained on two-dimensional data, ELMs are better in training than CNN [11]. Ensemble learning-based way introduced to detect malware and initially is categorized by many of fully connected and one dimension of CNN, after that is categorized by 5 algorithms for machine learning: Decision Tree, Naive Bayes, Gradient Enhancement, Random Forest and AdaBoost, optimal performance was achieved by using ensemble 7 neural networks and ExtraTrees as classifier [12].

## II. THEROICAL BACKGROUND

This section gives an overview of the many forms of malware analysis, followed by a discussion of deep learning methods in artificial intelligence for malware detection and categorization.

### *A. Malware Analysis*

Static and dynamic malware analyses are the two basic methods of malware analysis. Static Analysis examines a malware file's binary code without running it. Since Schultz et al. [13] proposed data mining framework based on Naive Bayes method one of the machine learning methods for detecting harmful executable that have never been seen before, precisely and automatically. With static analysis, authors proposed an approach for identifying Trojans based solely on the lengths of their functions that is quick, simple, and scalable [14], and a methodology for automatically classifying malware according to structural data (graph by function call) [15]. Though each program execution will be mirrored in the code, analysis by static method is not always a simple operation since attackers utilize code obfuscation methods like encryption, binary packers and self-modifying mechanism to avert static analysis.

The obfuscations will have no effect on dynamic analysis, which examines the behaviour of malware while it is being executed in a sandbox such as TTAnalyzer [16] or CWSandbox [17]. A sandbox is a virtual machine that simulates the environment to shield the operating system from damage. A sandbox is a controlled environment in which a malware sample is executed. It may actively monitor and record system calls and behaviour, which may be used to determine if the program is benign or malicious[18][19]. However, dynamic analysis only runs the program for a brief period, and it may not be able to trigger all of the program's execution. Nataraj et al [20]. A malware visualization method was proposed in 2010, the malware binary can be read as an 8-bit vector, which can then be organized into a two-dimensional array. A Gray scale image can be used to represent this. This form of visualization naturally captures the features of many forms of malware and introduces a new approach to malicious analysis. The format and texture of images belonging to the same family of malware are often very similar. In response to this visual similarity, a classification approach based on traditional image attributes has been introduced, as well as several visualization tools for malware investigation such as image processing techniques with Support Vector Machine (SVM) algorithm [21]. The malware binaries were turned into opcode sequences, which were subsequently translated into images. The malware images are then enhanced using image processing techniques and detected using an SVM classifier [22].

### *B. Deep Learning Techniques*

Deep learning which is considered one of the important branches of artificial intelligence has gained a lot of successes in recent years in many fields like natural language processing, computer vision, item detection, voice recognition and the one of the most important is the composition of non-linear functions for modelling input features and output labels [23]. Neural networks, which are like human neurons, were created to mimic the human nervous system for machine learning purposes. The main goal of neural networks is to construct artificial intelligence by designing devices that imitate the computations in the human nervous system. [24]. When it comes to infrastructure, there are several layers that must be learned, Techniques like CNN are commonly used. The network goes through two stages of training: forward and backward. The aim of the forward step is to process each layer's input picture using the current parameters (bias and weights). The first layers look for low-level features like boundaries, angles, and corners, while the other layers look for structures, artifacts, and shapes at the mid to high-level. [25][26]. Backward phase's goal is to count the error and updates the network parameters. CNN consists of several layers which are layers for input, layers for convolutional, layers for pooling, and layers for output. In convolutional layers, the features of the images are extracted by creating new images called feature maps which are highlight the specific features in the images while in the layer of pooling, the size of image is reduced by merging the pixel units adjacent to a specific area into one representative division such max pooling which take maximum pixel in the selected pixels [27]. Since the convolutional layer and pooling layers of a CNN are not clearly visible from the system's inputs and outputs, they are referred to as hidden layers as shown in Fig.2. The main equation in neural network for fully connected:

$$\gamma = b + \sum_{i=1}^{n} (x_i w_i) \quad (1)$$

Where $b$ is referred to the bias, $x$ denotes neuron's input, $w$ represents weights, $n$ denotes the number of inputs from the arriving layer and $i$ denotes a counter from 1 to $n$.

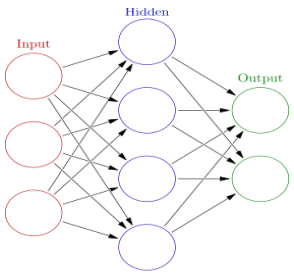


Fig. 2. Neural Network Model

TABLE I. MALIMG MALWARE FAMILIES DATASET

| No. | Family name | Family | Samples |
|---|---|---|---|
| 1 | Skintrim.N | Trojan Horse | 80 |
| 2 | Wintrim.BX | Trojan Downloader | 97 |
| 3 | Autorun.K | Worm AutoIT | 106 |
| 4 | Agent.FYI | Backdoor | 116 |
| 5 | Adialer.C | Dialer | 122 |
| 6 | Lolyda.AA3 | Password Stealer | 123 |
| 7 | Swizzor.gen!E | Trojan Downloader | 128 |
| 8 | Swizzor.gen!I | Trojan Downloader | 132 |
| 9 | Malex.gen!J | Trojan Horse | 136 |
| 10 | Obfuscator.AD | Trojan Downloader | 142 |
| 11 | C2LOP.gen!g | Trojan Horse | 146 |
| 12 | Rbot!gen | Backdoor | 158 |
| 13 | Lolyda.AT | Password Stealer | 159 |
| 14 | Dontovo.A | Trojan Downloader | 162 |
| 15 | Dialplatform.B | Dialer | 177 |
| 16 | Lolyda.AA2 | Password Stealer | 184 |
| 17 | Alueron.gen!J | Trojan Horse | 198 |
| 18 | C2LOP.P | Trojan Horse | 200 |
| 19 | Lolyda.AA1 | Password Stealer | 213 |
| 20 | Fakerean | Rogue | 381 |
| 21 | VB.AT | Worm | 408 |
| 22 | Instantaccess | Dialer | 431 |
| 23 | Yuner.A | Worm | 800 |
| 24 | Allaple.L | Worm | 1591 |
| 25 | Allaple.A | Worm | 2949 |

## III. METHODOLOGY AND RESULTS

### A. Dataset

In this study, a dataset from the University of California, consisting of 9,339 models in the form of gray images, was used, categorized into the 25 malware types prevalent in the world [20], as shown in Table I. It has been found that the malware images associated with a particular family are relatively similar, but different from other families.

In this dataset the binary malware was converted into an image by converting the Portable Executable (PE) file into an 8-bit vector, and after that it is converted to image as shown in Fig. 3 that illustrates the conversion process. The resulting image contains numbers between (0-255) because every pixel in the image consists of 8 bits where 0 is black and 255 is white. Gray scale images shown in Fig.4.

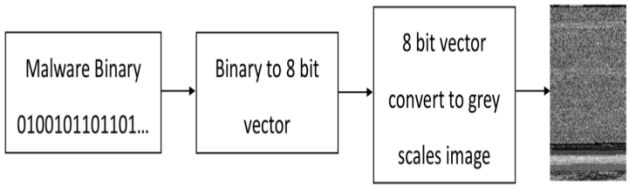


Fig. 3. Convert Malware to Images.

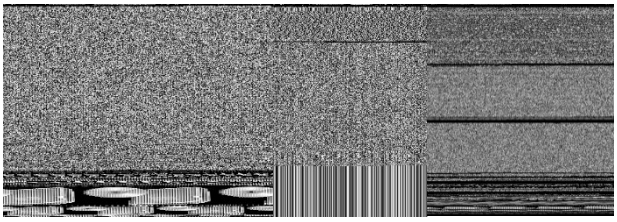

Fig. 4. Malware Images Samples

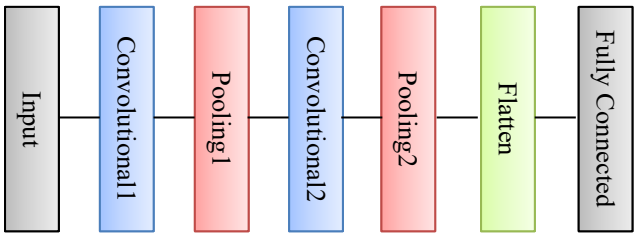


Fig. 5. Proposed Method Structure

### B. Proposed Method

Since the input of CNN receives constant size entries so we resized the images to 64 x 64 pixels, which is the process of changing the file size of image without cutting any part of it then define and build a deep learning structure based on CNN which deals with images of malware samples and include of one layer for input, two layers for convolutional, two layers for Max-Pooling, and one layer for flatten, which converts two-dimensional images to one-dimensional and one layer for output consists of 25 categories depending on the number of malware as shown in Fig.5.

### C. Results

The study model was applied practically on a computer with the following specifications to measure the accuracy:

- windows 7 System.
- CPU Core 2 Duo.
- Python 3.7.9 with Jupiter editor.
- Intel GPU 1 GB.

In our model, 70% of the dataset was used for training, and 30% was used for testing and the accuracy reached to 96.76%. Fig.6 shows the result of the detection of the malware samples. We notice the increase in accuracy with the increase in the number of times of training known as epochs in neural network in our model in the program after designing and writing the codes in Python.

$$Accuracy = (TP + TN) / (TP + TN + FP + FN)$$

Where true positive (TP) and false positive (FP) apply to the number of image tests that were correctly classified as malware and those that were falsely classified as malware. The number of images tests correctly and wrongly labelled as benign is referred to as true negative (TN) and false negative (FN) respectively.

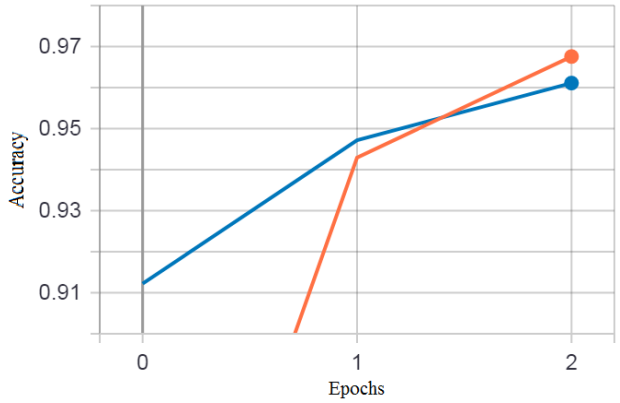


Fig. 6. The Accuracy of the model

## IV. Conclusion

In this study, CNN technology which is one of the most efficient deep learning techniques was applied to a dataset called Malimg from the University of California, which consists of 25 families of malware. The efficiency of this technique has been proven to detect malware and classify it into the family of each type with a high accuracy of 96.76% in our experiment.

In the future, we will work with similar algorithms represented by one shot learning technology to detect malware even in case the hacker changes parts of the malicious file's binary code.

## Acknowledgment

This work is supported by Dr. Muhammet Tahir Guneser, one of the professors of Karabuk University in Turkey, through our teaching of the scientific research course.

## References

[1] AV-TEST, "Last 10 years malware statistics." https://www.av-test.org/en/statistics/malware/ (accessed May 13, 2021).

[2] Symantic, "Internet Security Threat Report 2016." https://docs.-broadcom.com/doc/istr-21-2016-en (accessed Apr. 30, 2021).

[3] A. Liptak, "The WannaCry ransomware attack has spread to 150 countries." https://www.theverge.com/2017/5/14/15637888/authori-ties-wannacry-ransomware-attack-spread-150-countries (accessed May 01, 2021).

[4] A. Patil and M. Rane, "Convolutional Neural Networks: An Overview and Its Applications in Pattern Recognition," *Smart Innov. Syst. Tech-nol.*, vol. 195, pp. 21–30, 2021, doi: 10.1007/978-981-15-7078-0_3.

[5] S. Choi, S. Jang, Y. Kim, and J. Kim, "Malware detection using malware image and deep learning," in *2017 International Conference on Information and Communication Technology Convergence (ICTC)*, 2017, pp. 1193–1195.

[6] N. Idika and A. P. Mathur, "A Survey of Malware Detection Techniques," *SERC Tech. Reports*, no. October, 2007, [Online]. Available: http://www.serc.net/report/tr286.pdf.

[7] Y. Liu and Y. Wang, "A robust malware detection system using deep learning on API calls," *Proc. 2019 IEEE 3rd Inf. Technol. Networking, Electron. Autom. Control Conf. ITNEC 2019*, no. Itnec, pp. 1456–1460, 2019, doi: 10.1109/ITNEC.2019.8728992.

[8] F. D. Q. Ghdo, Z. Odujh, and V. Uhfrjqlwlrq, "Identification of malicious code variants based on image visualization," no. 978, pp. 581–585, 2020.

[9] G. Sun and Q. Qian, "Deep Learning and Visualization for Identifying Malware Families," *IEEE Trans. Dependable Secur. Comput.*, vol. 18, no. 1, pp. 283–295, 2021, doi: 10.1109/TDSC.2018.2884928.

[10] J. Hemalatha, S. A. Roseline, S. Geetha, S. Kadry, and R. Damaševičius, "An Efficient DenseNet-Based Deep Learning Model for Malware Detection," *Entropy*, vol. 23, no. 3, p. 344, 2021, doi: 10.3390/e23030344.

[11] M. Jain, W. Andreopoulos, and M. Stamp, "Convolutional neural networks and extreme learning machines for malware classification," *J. Comput. Virol. Hacking Tech.*, vol. 16, no. 3, pp. 229–244, 2020, doi: 10.1007/s11416-020-00354-y.

[12] N. A. Azeez, O. E. Odufuwa, S. Misra, J. Oluranti, and R. Damaševičius, "Windows PE Malware Detection Using Ensemble Learning," *Informatics*, vol. 8, no. 1, p. 10, 2021, doi: 10.3390/informatics8010010.

[13] M. G. Schultz, E. Eskin, E. Zadok, and S. J. Stolfo, "Data mining methods for detection of new malicious executables," *Proc. IEEE Comput. Soc. Symp. Res. Secur. Priv.*, pp. 38–49, 2001, doi: 10.1109/secpri.2001.924286.

[14] R. Tian, L. M. Batten, and S. C. Versteeg, "Function length as a tool for malware classification," *3rd Int. Conf. Malicious Unwanted Soft-ware, MALWARE'08*, pp. 69–76, 2008, doi: 10.1109/MALWARE.-2008.4690860.

[15] D. Kong and G. Yan, "Discriminant malware distance learning on structural information for automated malware classification," *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, vol. Part F1288, pp. 1357–1365, 2013, doi: 10.1145/2487575.2488219.

[16] U. Bayer, C. Kruegel, and E. Kirda, "TTAnalyze: A tool for analyzing malware," *15th Ann. Conf. Eur. Inst. Comput. Antivirus Res.*, pp. 180–192, 2006, [Online]. Available: https://www.auto.tu-wien.ac.at/~chris/research/doc/eicar06_ttanalyze.pdf.

[17] G. Willems, T. Holz, and F. Freiling, "Toward automated dynamic malware analysis using CWSandbox," *IEEE Secur. Priv.*, vol. 5, no. 2, pp. 32–39, 2007, doi: 10.1109/MSP.2007.45.

[18] J. Zico Kolter and M. A. Maloof, "Learning to detect and classify malicious executables in the wild," *J. Mach. Learn. Res.*, vol. 7, pp. 2721–2744, 2006.

[19] B. Anderson, D. Quist, J. Neil, C. Storlie, and T. Lane, "Graph-based malware detection using dynamic analysis," *J. Comput. Virol.*, vol. 7, no. 4, pp. 247–258, 2011, doi: 10.1007/s11416-011-0152-x.

[20] L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, "Malware images: Visualization and automatic classification," *ACM Int. Conf. Proceeding Ser.*, no. July, 2011, doi: 10.1145/2016904.2016908.

[21] A. Makandar and A. Patrot, "Malware class recognition using image processing techniques," *2017 Int. Conf. Data Manag. Anal. Innov. ICDMAI'17*, pp. 76–80, 2017, doi: 10.1109/ICDMAI.2017.8073489.

[22] T. Wang and N. Xu, "Malware variants detection based on opcode image recognition in small training set," *2017 2nd IEEE Int. Conf. Cloud Comput. Big Data Anal. ICCCBDA 2017*, pp. 328–332, 2017, doi: 10.1109/ICCCBDA.2017.7951933.

[23] J. Fan, C. Ma, and Y. Zhong, "A selective overview of deep learning," *arXiv Prepr. arXiv1904.05526*, 2019.

[24] C. C. Aggarwal, "Neural networks and deep learning," *Springer*, vol. 10, pp. 973–978, 2018.

[25] B. B. Traore, B. Kamsu-Foguem, and F. Tangara, "Deep convolution neural network for image recognition," *Ecol. Inform.*, vol. 48, pp. 257–268, 2018.

[26] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[27] P. Kim, *MATLAB Deep Learning: With Machine Learning, Neural Networks and Artificial Intelligence*. 2017.