Scientific Research Publishing

Journal of Computer and Communications, 2018, 6, 153-158
http://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

Malware Images Classification Using Convolutional Neural Network

Espoir K. Kabanga, Chang Hoon Kim*

Department of Computer and Information Engineering, Daegu University, Gyeongsan-si, Korea
Email: hopekab@gmail.com, *kimch@daegu.ac.kr

How to cite this paper: Kabanga, E.K. and Kim, C.H. (2018) Malware Images Classification Using Convolutional Neural Network. *Journal of Computer and Communications*, **6**, 153-158.
https://doi.org/10.4236/jcc.2018.61016

Received: November 9, 2017
Accepted: December 26, 2017
Published: December 29, 2017

Abstract

Deep learning has been recently achieving a great performance for malware classification task. Several research studies such as that of converting malware into gray-scale images have helped to improve the task of classification in the sense that it is easier to use an image as input to a model that uses Deep Learning's Convolutional Neural Network. In this paper, we propose a Convolutional Neural Network model for malware image classification that is able to reach 98% accuracy.

Keywords

Malware, Convolutional Neural Network, Malware Classification

1. Introduction

Malicious Software, commonly called malware, is one the most dangerous threats in information technology society. Annual report from antivirus companies shows that thousands of new malware are created every single day. These new malwares become more sophisticated that they could no longer be detected by the traditional detection techniques such as signature-based detection, heuristic detection or behavior-based detection [1].

Signature-based detection searches for specified bytes sequences into an object so that it can identify exceptionally a particular type of a malware. Its drawback is that it cannot detect zero-day or new malware since these malware signatures are not supposed to be listed into the signature database. Heuristic-based detection was developed to basically overcome the limitation of the signature detection technique, in the way that it scans the system's behavior in order to identify the activities which seems to be not normal, instead of searching for the malware signature. Because of this, heuristic-based detection method can be applied to

*Corresponding author.

newly created malware whose signature has not yet been known. The limitation of this technique is that it affects the system's performance and requires more space. Behavior-based detection technique is more about the behavior of the program when it is executing. If a program executes normally, then it is marked as benign, otherwise it is marked as a malware. By analyzing this definition of the behavior-based detection, we can directly conclude that the drawback of this technique is the production of many false positives and false negatives, considering the fact that a benign program can crashed and be marked as a virus or virus can execute as if it was a normal program and simply be marked as benign.

Therefore, the need for much more strong algorithms was required and according to researches and investigation made, machine learning algorithms were found to be very efficient and reliable. [2] gave an overview of different machine learning techniques that were previously proposed for malware detection.

We propose a model that uses machine learning's convolution neural network to classify images extracted from malware binaries and it happens to be robust as it achieves 98% accuracy for testing.

The rest of the paper is structured as follows: We briefly describe some related works in Section 2; Section 3 describes our model; in Section 4 we talk about the dataset; Section 5 describes the result and finally section concludes this paper.

2. Related Work

Visualizing malware as a gray-scale image is a great achievement for malware classification task. Nataraj *et al.* in [3] came up first with the approach of visualizing malware as gray-scale images. In that work, they represented a malware as a gray-scale image of in the range of [0, 255], where 0 being for black and 255 for white. They could observe that the obtained images presented different sections which in turn represented different information about the malware. They used GIST to compute texture features from malware images and K-nearest neighbors for classification.

The work done in [3] has incredibly helped Xiaozhou *et al.* in [4] to win the first place of Microsoft Malware Classification Challenge in 2015. For that challenge, a malware dataset of 500 GB belonging to 9 different families was provided. The winner's solution relied on the extraction of three types of features which are: opcode 2, 3 and 4-grams, segment line count and asm file pixel intensity (instead of binary file).

Other techniques have been used for malware classification. Kolter *et al.* proposed the use machine learning and data mining to detect and classify malicious executables [5]. They evaluated a several methods that include naïve Bayes, decision trees, SVM and boosting and realized that boosted decision trees outperformed other methods with an area under the ROC curve of 0.996. Islam *et al.* in [6] proposed a framework that combined the static features of a function length and printable string information extracted from malware samples into a single test and used k-fold cross validation on the malware that included Trojans, viruses and clean files and achieved an overall classification accuracy of over 98%.

Mansour *et al.* presented a classification system that was characterized by a limited complexity in feature design and in the classification mechanism that they employed [7]. They used a number of novel features to represent discriminant characteristics between different families and the obtained result allowed them to assess the effectiveness of those features with respect to the classification accuracy and to the impurity.

3. Proposed Method

We have used convolutional neural networks because it is reliable and it can be applied to the entire image at a time and then we can assume they are best to use for feature extraction.

Convolutional neural network is a feed-forward neural network where the connectivity pattern between neurons is inspired by the structure of an animal visual cortex and that has proven great value in the analysis of visual imagery.

Our model is described in the following steps:

1) All the images are reshaped into a size of 128 × 128 × 1 (1 being the channel width).

2) Since all the models of deep learning accept data in form of numbers, we have used image library from PIL package of Python to generate vectors of images and further processing are done on these vectors.

3) We have then designed a three-layers deep Convolutional Neural Network for the classification task, which has the following properties: On the Rectified Linear Units (ReLU) layers, we first apply a two-dimensional convolutional layer and after each layer, we applied a nonlinear later also known as activation layer. In convolutional layer, we have operations like element wise multiplication and summations. The ReLU adds nonlinearity to the system. We have used the ReLU instead of nonlinearity function because it is faster than tanh or sigmoid and help in vanishing gradient problem which arises in lower layers of the network. We have also used max pooling layer instead of other layers. It takes a filter and a stride of the same length then applies it to the input volume and outputs the maximum number in sub region that the filter convolves around.

4) The output that we want is a single class in which the given malware belongs to. After applying all the layers, we have a three-dimensional vector of arrays. To convert this vector into a class probability, we convert these vectors into a single layer of one dimension, known as fully connected layer. Downsampling all the vectors to a one-dimensional vector may lead to loss of data. For that reason, we have used two fully connected layers.

5) Cross entropy loss function that is commonly used for multi class classification was used for this work as well as Adam optimizer for optimization task.

The overall architecture of the model is shown in **Figure 1**.

Initially, all the images were of different sizes and had to be converted into 128 by 128 pixels before they are used as input to the model. The output layer is of 25 neurons corresponding to the 25 families of malwares available in the dataset used.

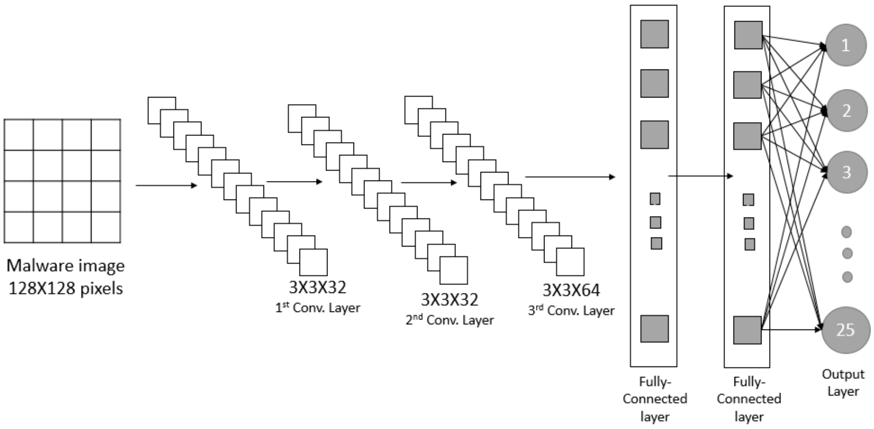

Figure 1. Overview architecture of proposed method.

4. Dataset

We have used the Malimg Dataset [8] that had been used in [3]. It consists of 9458 grayscale images of 25 malware families among which 90% of the total data is used for training and 10% is used for testing. **Table 1** describes the datasets.

As explained in [3], a given malware binary can be read as a vector of 8 bits unsigned integers and organized into a 2-dimensional array and this can be visualized as a grayscale image in the range of [0, 255], where 0 represent back and 255 for white. The size of the image is different depending to their families.

It is important to note that images belonging to the same family appear to look similar to one another and each image presents different textures which represent different information about the malware, as it is shown in **Figure 2**.

5. Experimental Result

The result obtained in the experiment shows an accuracy of 98%, which is same with that obtained in [3] using k-nearest neighbor (KNN) approach of classification. The technique is better compared to some of the traditional methods of classification, for example the one presented by Rieck *et al.* in [9]. However, using image features alone did not overcome the result achieved by the winner of the Microsoft malware classification challenge in 2015, which also used convolutional neural network approach and achieve over 99% accuracy by using three kinds of features extracted from almost half a terabytes of malware sample.

6. Limitation and Conclusion

We have presented a Convolutional Neural Network model that classified images extracted from malware samples. The result is quite competitive.

Being able to visualize malware as gray-scale images has been a great achievement. Many researchers have been using this technique for the task of malware classification and detection. However, other works have shown that this technique can be easily vulnerable to adversarial attack and produce erroneous results. Work done in [10] [11] [12] have shown how a small change in the image

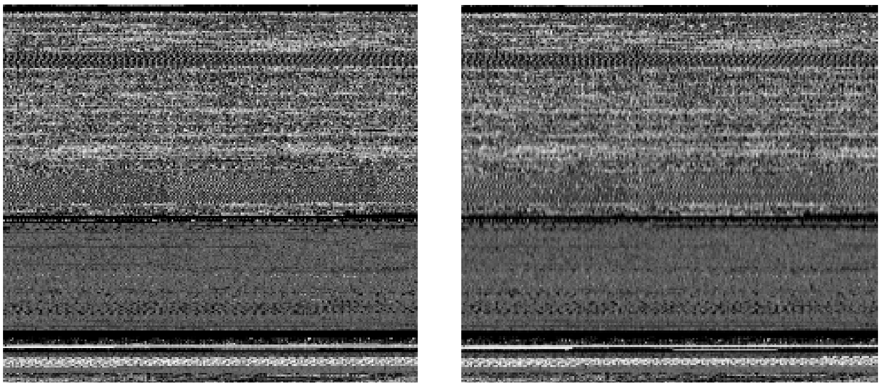

Figure 2. Images extracted from malware of Adialer.C family.

Table 1. Dataset description.

No.	Class	Family Name	No. of samples
1	Worm	Allaple. L	1591
2	Worm	Allaple. A	2949
3	Worm	Yuner. A	800
4	PWS	Lolyda. AA 1	231
5	PWS	Lolyda. AA 2	184
6	PWS	Lolyda. AA 3	123
7	Trojan	C2Lop. P	146
8	Trojan	C2Lop.gen!G	200
9	Dialer	Instantaccess	431
10	Trojan Downloader	Swizzor.gen!l	132
11	Trojan Downloader	Swizzor.gen!E	128
12	Worm	VB.AT	408
13	Rogue	Fakerean	381
14	Trojan	Aluron.gen!J	198
15	Trojan	Malex.gen!J	136
16	PWS	Lolyda. AT	159
17	Dialer	Adialer. C	125
18	Trojan Downloader	Wintrim. BX	97
19	Dialer	Dialplatform. B	177
20	Trojan Downloader	Dontovo. A	162
21	Trojan Downloader	Obfuscator. AD	142
22	Backdoor	Agent. FYI	116
23	Worm: AutoIT	Autorun. K	106
24	Backdoor	Rbot!gen	158
25	Trojan	Skintrim. N	80

(that might not be visible to a human) could lead to misclassification of images. Therefore, using images features alone for the task of malware classification can be dangerous in the way that a small accident while extracting images from malware can produce erroneous results.

Acknowledgements

This research has been supported by the Technology Transfer and Commercialization Program through INNOPOLIS Foundation and funded by the Ministry of Science and ICT (2017DG0018/Intelligence CCTV Using 3D Camera).

References

[1] Al Amro, S. and Alkhalifah, A. (2015) A Comparative Study of Virus Detection Techniques. *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, **9**.

[2] Gavrilut, D., Cimpoesu, M., Anton, D. and Ciortuz, L. (2009) Malware Detection Using Machine Learning, *Proceedings of the International Multiconference on Computer Science and Information Technology*, 735-741. https://doi.org/10.1109/IMCSIT.2009.5352759

[3] Nataraj, L., Karthikeyan, S., Jacobs, G. and Manjunath, B.S. Malware Images.

[4] Wang, X., Liu, J. and Chen, X. (2015) First Place Team: Say No to Overfitting, Winner of Microsoft Malware Classification Challenge (BIG 2015).

[5] Zico Kolter, J. and Maloof, M.A. (2006) Learning to Detect and Classify Malicious Executables in the Wild. *Journal of Machine Learning Research*, **7**, 2721-2744.

[6] Islam, R., Tian, R., Batten, L. and Versteeg, S. (2010) Classification of Malware Based on String and Function Feature Selection. 2010 *Second Cybercrime and Trustworthy Computing Workshop*.

[7] Ahmadi, M., Ulyanov, D., Semenov, S., Trofimov, M. and Giacinto, G. (2016) Novel Feature Extraction, Selection and Fusion for Effective Malware Family Classification. Cornell University Library. arXiv:1511.04317

[8] http://old.vision.ece.ucsb.edu/spam/malimg.shtml

[9] Rieck, K., Holz, T., Willems, C., Dussel, P. and Laskov, P. (2008) Learning and Classification of Malware Behavior. *Fifth Conference on Detection of Intrusions and Malware and Vulnerability Assessment* (*DIMVA*), 108-125. https://doi.org/10.1007/978-3-540-70542-0_6

[10] Nguyen, A., Yosinki, J. and Clune, J. (2015) Deep Neural Networks Are Easily Fooled: High Confidence Predictions for Unrecognizable Images. *IEEE*, *Computer Vision and Pattern Recognition* (*CVPR*).

[11] Papernot, N., McDaniel, P. and Goodfellow, I. (2017) Practical Black-Box Attacks against Machine Learning. *Proceedings of the* 2017 *ACM on Asia Conference on Computer and Communication Security*, 506-519. https://doi.org/10.1145/3052973.3053009

[12] Goodfellow, I., Shlens, J. and Szegedy, C. (2015) Explaining and Harnessing Adversarial Examples. 2015 *International Conference on Learning Representations.*