# Binary Malware image Classification using Machine Learning with Local Binary Pattern

Jhu-Sin Luo
Department of Computer Science
Kennesaw State University
jluo6@students.kennesaw.edu

Dan Chia-Tien Lo
Department of Computer Science
Kennesaw State University
dlo2@kennesaw.edu

***Abstract*—Malware classification is a critical part in the cybersecurity. Traditional methodologies for the malware classification typically use static analysis and dynamic analysis to identify malware. In this paper, a malware classification methodology based on its binary image and extracting local binary pattern (LBP) features is proposed. First, malware images are reorganized into 3 by 3 grids which is mainly used to extract LBP feature. Second, the LBP is implemented on the malware images to extract features in that it is useful in pattern or texture classification. Finally, Tensorflow, a library for machine learning, is applied to classify malware images with the LBP feature. Performance comparison results among different classifiers with different image descriptors such as GIST, a spatial envelop, and the LBP demonstrate that our proposed approach outperforms others.**

***Keywords*—malware, classification, machine learning, visualization, local binary pattern.**

## I. Introduction

Over the past few years, the Internet usage had experienced an exponential growth. It has became an important part of our daily lifes. The cybersecurity is also playing a role in that the online financial activities such as the online payment and online money transaction become widespread [1]. The users of the Internet face threats from the malware which causes detriment to users of computer and the Internet. AV-TEST, an IT security Institute, registers over 583 million the malware in 2017[2] and based on their reports, the amount of the malware dramatically increases every year.

Traditional methodologies for the malware classification or detection mainly use static analysis and dynamic analysis to identify type of the malware and behavior of the malware. Both methodologies have their advantages and disadvantages. Static analysis examines the executable file without actually executing. It extracts the binary code from the file to generate the patterns or features which could be used to identify whether the file is the malware or not. The static analysis is ineffective against different code obfuscation[3]. On the other hand, dynamic analysis verify the file by executing on the secure environment or virtual environment. By executing file, the behaviors of the malware is able to observe. Nonetheless, dynamic analysis still exist disadvantages. The malware might have different behaviors in two different environments or some behaviors may need to be triggered on specific circumstances.

In this paper, a malware classification approach based on image processing and convolutional neural network is proposed. First, as Figure 1 demonstrates, each pixel in the malware images are reorganized. The pixels of the original malware images are constructed by line by line. We rearrange each pixel of images by 3 by 3 grids. Second, the LBP is applied on the malware image to extract features. Finally, malware images are classified by TensorFlow and the result would be compared with other classifiers.

## II. Related Work

In [1], L.Nataraj, S.Karthikeyan, G. Jacob and B. S. Manjunath visualize malware into grey scale. They applied GIST descriptor on the malware images. The GIST descriptor is useful on scene classification. The K-nearest neighbor is utilized to classify malware images. In [5], Aziz Makandar and Anita Patrot applied Discrete Wavelet Transformation (DWT) on the malware images to extract features. They use Support Vector Machine (SVM) to discriminating the malware classes. In [14], Aziz Makandar and Anita Patrot obtain the global features of the malware images by using gabor wevelet transform and GIST. And Artificial neural network (ANN) is used to train and test malware images.

## III. Our Methodology

In this section, we demonstrate our approach step by step. First, we demonstrate malware visualization and reorganization. Second, we introduce the LBP and how to apply the LBP on our images. Third, we show our TensorFlow architecture which we utilize to train and classify.

### A. Malware Visualization

In [1], L.Nataraj, S.Karthikeyan, G. Jacob and B. S. Manjunath visualized the malware into grey scale image in the range [0, 255]. The width of image is fixed and the height is allowed to vary. In [14], Aziz Makandar and Anita Patrot also convert malware into grey scale in the range [0, 255]. In [5], the malware is also visualized into grey scale image and normalized into 256*256 dimension.

Our methodology is that we reorganized the grey scale malware images which are provided by [1, 4] L.Nataraj, S.Karthikeyan, G. Jacob and B. S. Manjunath. They convert the malware into images with grey scale. The malware images with grey scale are obtained by reading malware in binary. A Malware binary is read as a vector of 8 bit unsigned integers and then arranged into 2D array (Figure 2). We rearrange each pixel in the malware images into 3 by 3 grid (Figure 3). We convert malware images into 3 by 3 grid in that it is suitable for extracting LBP descriptor.

978-1-5386-2715-0/17/$31.00 ©2017 IEEE

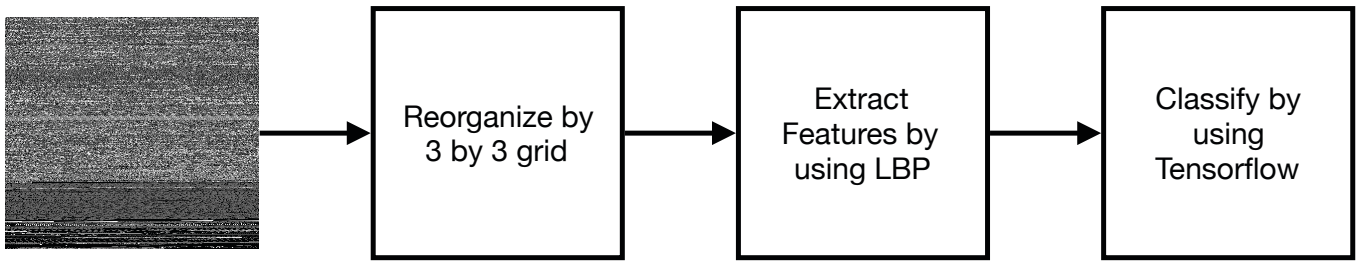


Fig. 1: Overview of entire System

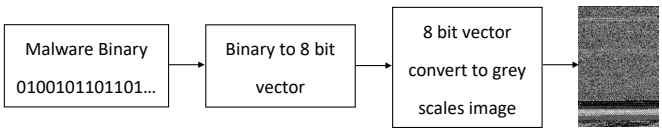


Fig. 2: Reorganized malware image

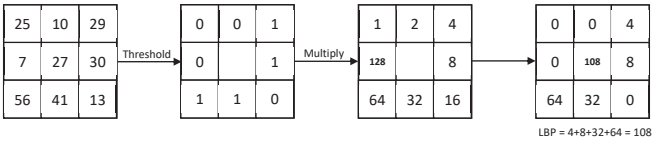


Fig. 4: The LBP operator

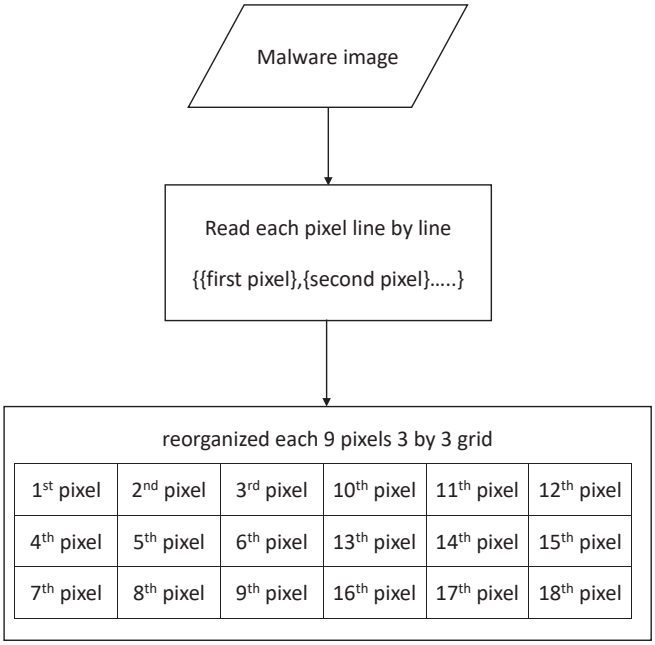


Fig. 3: Reorganized malware image

### B. Local Binary Pattern

The Local Binary Pattern, a visual descriptor, is useful for texture analysis and texture classification [6,7,8]. As Figure 4 demonstrates, the value of central pixel is threshold. The 8 neighbors around a pixel are compared with the central pixel. If a neighbor's value is greater than central pixel, the value of the neighbor is written '1'. The value of neighbor which less then threshold is written '0'. The threshold results are multiplied with weights which are given by power of two. The central value is the sum of the multiplying results. For each pixel in the image do the same process. The final LBP descriptor can be obtained by calculating the histogram of the image.

### C. TensorFlow Architecture

We use TensorFlow [13, 15] for training and testing. As Figure 5 shows, we use 3x3 convolutional filter with ReLU and then perform 2x2 max pooling layer with stride 2 to downsample. The number of first convolutional filter is 16. The size of second convolutional filter is also 3x3 but with 32 filters. The output of max pooling is multi-dimensional. The flattening layer is applied to convert multi-dimensional nodes into one dimensional nodes. After flattening output, the fully connected layer is obtained.

### D. Dataset

The dataset we use is provided by [1, 4]. This dataset includes 32 families and around 12000 malware images with grey scale (table I). The types of malwares mainly belong to trojan, password steeler and virus. We use 20% of each malware family dataset for training and the rest for testing.

## IV. Experimental Results

We evaluate Tensorflow for the LBP features classification, and use LBP features for training Support Vector Machine (SVM) classifier and k-nearest neighbor (KNN) classifier. We also implement GIST [9, 10, 11] features with TensorFlow, KNN and SVM. Table II, table III and table IV are the confusion matrices of Tensorflow, KNN and SVM using LBP feature. According to the confusion matrices, we discover that the malware belong to family 28, 29 and 30 which are Virut.A, Virut.AC and Virut.AT respectively are easy to get confused. As seen in table II, Tensorflow can differentiate these three with higher accuracy than others. Table V displays the accuracy of different methodologies over 32 malware families.

## V. Pros and Cons

Our approach run with GPU, which is significantly shorter the execution time (Figure 6). Moreover, this method doesn't have to run on a virtual machine or virtual environment

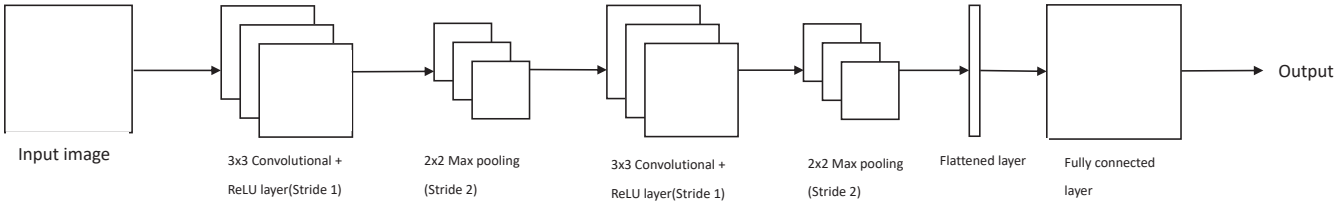


Fig. 5: Architecture of TensorFlow

| Malware Family | Type of malware | Amount of malware |
|---|---|---|
| Adialer.C.UPX | Adialer | 188 |
| Agent.FYI | Backdoor | 116 |
| Aliser.7825 | Trojan | 256 |
| Allaple.A | Worm | 4540 |
| Alueron_Gen_J | Trojan | 198 |
| Autorun.A | Worm | 106 |
| Azero.A | Trojan | 121 |
| Backdoor.Agent.AsPack | Backdoor | 180 |
| C2Lop | Trojan | 692 |
| Dialplatform.B | Dialer | 177 |
| Dontovo.A | TrojanDownloader | 162 |
| Fakerean | Rogue | 381 |
| Farfli.I | Backdoor | 94 |
| Instantaccess | Dialer | 431 |
| Lolyda.AA1 | PasswordSteeler | 213 |
| Lolyda.AA2 | PasswordSteeler | 184 |
| Lolyda.AA3 | PasswordSteeler | 123 |
| Lolyda.AT | PasswordSteeler | 159 |
| Luder.B | Virus | 509 |
| Malex.gen!J | Trojan | 136 |
| Nuwar.A | Virus | 51 |
| Obfuscator.AD | TrojanDownloader | 142 |
| Rbot.gen | Backdoor | 158 |
| Sality.AM | Virus | 127 |
| Skintrim.N | Trojan | 80 |
| Swizzor.gen | TrojanDownloader | 520 |
| VB.AT | Worm | 408 |
| Virut.A | Virus | 133 |
| Virut.AC | Virus | 269 |
| Virut.AK | Virus | 571 |
| Wintrim.BX | TrojanDownloader | 97 |
| Yuner.A | Worm | 800 |

TABLE I: Malware Family

to observe the behavior of malware. Additionally, because our approach is based on image processing, we can apply other image descriptors to do the voting to acheive higher classification accuracy. Although malware images can be analyzed with our approach based on local binary pattern and machine learning, there still have countermeasures. Because our approach converts the malware into binary and reorganizes. Therefore, if a rival who rewrites whole the program in other way or uses other instructions instead of original one result in changing whole the pattern of malware image, our approach may fail.

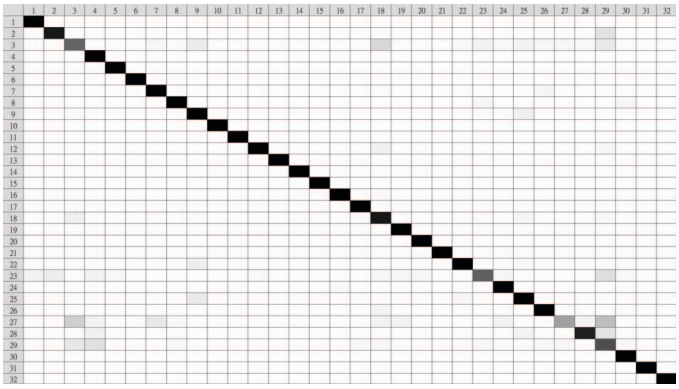

TABLE II: Confusion Matrix of Tensorflow using LBP feature

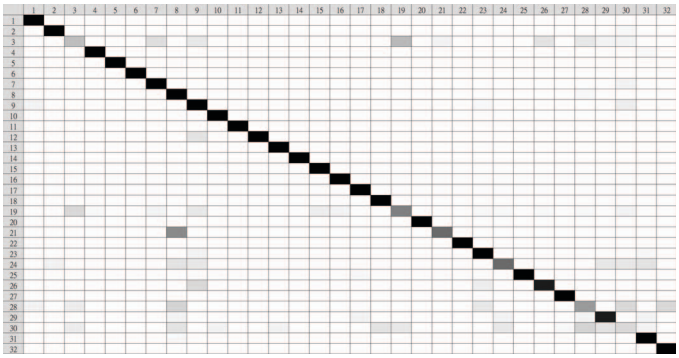

TABLE III: Confusion Matrix of KNN using LBP feature

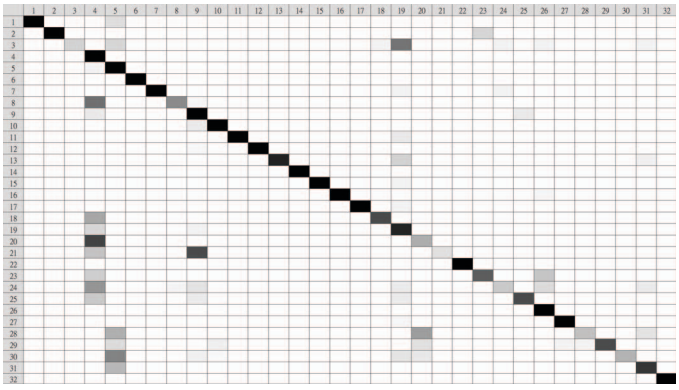

TABLE IV: Confusion Matrix of SVM using LBP feature

| Classification Method | Accuracy |
|---|---|
| TensorFlow+LBP | 93.17% |
| SVM+LBP | 87.88% |
| KNN+LBP | 85.93% |
| TensorFlow+GIST | 87.88% |
| SVM+GIST | 81.23% |
| KNN+GIST | 82.83% |

TABLE V: Experiment Result over 32 Malware Family

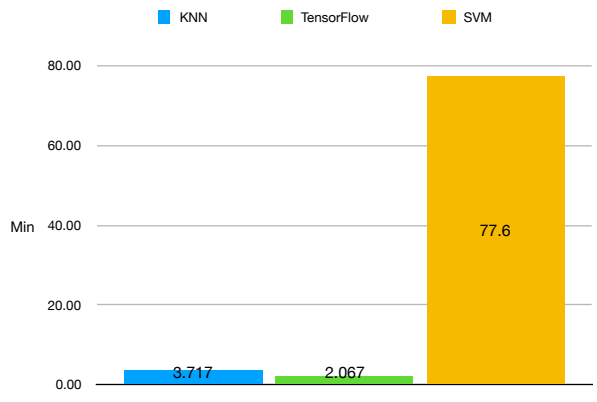


Fig. 6: Average Execution Time of each Methodology

## VI. Future Work

While our experimental results demonstrate that the accuracy using LBP as feature is slightly higher than other methodologies, there are ways of how the experiment could be improved. The first priority would be to extend the malware family, which means that increases the size and classes of dataset. At the meantime, converting malware file into image uses different approaches such as converting to RGBA color space instead of grey scale and using color-LBP[7, 8, 12] as feature, which is one possible future work. Additionally, we plan to design a different architecture of Tensorflow and examine more different image descriptor to increase the accuracy and reduce time consumption.

## Acknowledgment

This material is based in part upon work supported by the National Science Foundation under Grant Numbers 1623724, 1438858, 1244697, and 1241651. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. Furthermore, we would like to thank the authors [1] for providing the malware image dataset.

## VII. Conclusion

An experimental result shows that the accuracy based on our approach is 93.17%. The experiment is performed to classify malware images over 32 families around 12000 malware images. We reorganize malware images and utilize Local Binary Pattern as descriptor to extract features and classify the results with TensorFlow library. The comparison over different classifiers and features demonstrates that using LBP with TensorFlow obtains higher accuracy than others approaches. Furthermore, extending dataset of malware, converting malware to RGBA color space, designing different architectures of TensorFlow and testing more image descriptors is our future works, which may improves the research and obtains more comprehensive methodology.

## References

[1] Nataraj L., Karthikeyan S., Jacob G., Manjunath B. S.,"The malware Images: Visualization and Automatic Classification,"International Symposium on Visualization for Cyber Security (VizSec) ,July 20, 2011, Pittsburg, PA, USA.

[2] Malware statistic from: https://www.av-test.org/en/statistics/the malware/

[3] A. Moser, C. Kruegel and E. Kirda, "Limits of Static Analysis for Malware Detection," Twenty-Third Annual Computer Security Applications Conference (ACSAC 2007), Miami Beach, FL, 2007, pp. 421-430.

[4] Malware Images from http://vision.ece.ucsb.edu/~lakshman/malware_images/album

[5] Aziz Makandar and Anita Patrot, "Wavelet Statistical Feature Based Malware Class Recognition and Classification using Supervised Learning Classifier," Oriental Journal of Computer Science and Technology, ISSN: 0974-6471, June 2017, Vol. 10, No. (2): Pgs. 400-406

[6] T. Ojala, M. Pietikainen, and D. Harwood, "A Comparative Study of Texture Measures with Classification Based on Feature Distributions," Pattern Recognition, vol. 29, pp. 51-59, 1996.

[7] Chao Zhu, Charles-Edmond Bichot and Liming Chen, "Multi-scale Color Local Binary Patterns for Visual Object Classes Recognition," 2010 20th International Conference on Pattern Recognition, Istanbul, 2010, pp. 3065-3068.

[8] Chao Zhu, Charles-Edmond Bichot and Liming Chen, "Image region description using orthogonal combination of local binary patterns enhanced with color information," Pattern Recognition, Volume 46, Issue 7, 2013, Pages 1949-1963, ISSN 0031-3203

[9] Aude Oliva, Antonio Torralba, "Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope," International Journal of Computer Vision, Vol. 42(3): 145-175, 2001.

[10] A. Oliva and A. Torralba, "Building the gist of a scene: the role of global image features in recognition," Prog. Brain Res. Vis. Percept., vol. 155, pp. 2336, 2006.

[11] 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, pp. 273-280 vol.1.

[12] Chao Zhu, Charles-Edmond Bichot and Liming Chen, "Color orthogonal local binary patterns combination for image region description," Rapport technique RR-LIRIS-2011-012, LIRIS UMR, vol. 5205, p. 15, 2011

[13] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," arXiv preprint arXiv:1603.04467, 2016

[14] Aziz Makandar and Anita Patrot, "Malware Analysis and Classification using Artificial Neural Network," 2015 International Conference on Trends in Automation, Communications and Computing Technology (I-TACT-15), Bangalore, 2015, pp. 1-6.

[15] R. Pilipovi and V. Risojevi, "Evaluation of convnets for large-scale scene classification from high-resolution remote sensing images," IEEE EUROCON 2017 -17th International Conference on Smart Technologies, Ohrid, Macedonia, 2017, pp. 932-937.