ICIC Express Letters
Part B: Applications
Volume **11**, Number **8**, August **2020**

ICIC International ©2020 ISSN 2185-2766
pp. 761–766

ANDROID MALWARE CLASSIFICATION USING FLATTENED IMAGE REPRESENTATION AND SENET

Yuta Otani, Atsushi Ueno and Tomohito Takubo

Graduate School of Engineering
Osaka City University
3-3-138, Sugimoto, Sumiyoshi-ku, Osaka-shi 558-8585, Japan
otani@kdel.info.eng.osaka-cu.ac.jp; { ueno; takubo }@info.eng.osaka-cu.ac.jp

Received February 2020; accepted April 2020

Abstract. *A smartphone is an inevitable life base in modern society. Android is the most popular operating system of mobile devices and most mobile malware targets Android. A technology to analyze and classify malware efficiently is needed because of vast amount of malware. In this paper, we propose a malware classification approach using SE-ResNet34 deep neural network with input image flattened. Android malware samples are represented as byte plot images and height is converted to 1. These flattened images are used to train SE-ResNet34. As the result of experiments on datasets consisting of 18,963 samples from 7 families, we obtain an average accuracy of 98.32%. This result shows that using flattened images to train SE-ResNet34 is efficient for malware classification.*

Keywords: Malware classification, Malware visualization, Deep learning, Image classification

1. **Introduction.** A smartphone is an indispensable life base in modern society. It can be widely used not only for voice communication, emails and web browsing, but also for Internet banking. With the spread of mobile devices such as smartphones, malware targeting mobile devices is increasing. Malware is a generic term for software that is specifically designed to disrupt, damage, or gain unauthorized access to a computer system such as computer virus and Trojan horse.

Android is the most popular operating system of mobile devices. The IDC report [1] showed that Android dominated about 87% of the world market share in the first quarter of 2017. Android devices are exposed to malware threats because they are open source operating systems, have high market share and are accumulating import information. According to major security vendor, McAfee [2], the number of mobile malwares discovered by the third quarter of 2017 reached 21 million, an increase of 56% compared to 2016. The amount and variety of malware are vast, but most of them are variants created by modifying existing malware a little. Therefore, a technology to analyze and classify malware efficiently is needed. After identifying variants of malware, it is possible to cope with malware threats by appropriately applying existing approach against malware.

In recent years, deep learning technology shows its usefulness in a wide range of problems. Deep learning technology shows remarkable performance especially in the field of image recognition since the method using a Convolutional Neural Network (CNN) called AlexNet won the ImageNet Large Scale Visual Recognition Competition (ILSVRC), an image recognition competition, in 2012 with a major difference to the conventional method.

DOI: 10.24507/icicelb.11.08.761

In also malware classification task, the use of visualization and image recognition techniques has been studied in the literature. Nataraj et al. [6] proposed a method for classifying Windows executable malware represented as grayscale images. They use Gabor filter to extract GIST features from gray scale images, and then use a k-nearest neighbors (kNN) classifier. They obtained an accuracy of 97.18% in a dataset consisting of 25 malware families, totaling 9,458 malware samples. Kabanga and Kin [3] and Rezende et al. [4] made classifier using CNN and showed high accuracy. Kabanga and Kin [3] constructed a CNN model using a three-layer convolution layer and two-layer full connection layer. They trained the model using 128 × 128 grayscale images and achieved an average of 98% in a dataset consisting of 9,458 malware samples from 25 different families. Rezende et al. [4] used the VGG16 pre-trained model on the ImageNet dataset to extract feature from 224 × 224 grayscale byte plot images. Extracted features are used to train an SVM classifier for classifying malware family. They obtained an accuracy of 92.97% in a dataset consisting of 20 malware families, totaling 10,136 samples. When CNN is used for malware classification using image representation, the size of the malware image is converted to a square which one side is depending on the file size. Kabanga and Kin [3] and Rezende et al. [4] also used this measure.

When classifying malware using a CNN model for computer vision, the filter is usually square. However, the vertical direction of the image is only accidentally lined up by the folding, and there are few features that are connected in the vertical direction. Furthermore, when the image representation of malware is input directly in a CNN model, there is a problem of how to handle the image folding position in the convolution layer. Malware is time-series data, but when raw data of malware is converted to an image, the data of the folding position exist far away. Therefore, if an important feature shows a folding position, the feature will not be captured well of that position.

In this research, we propose an approach that converts a square image to a height 1 image and performs convolution using a filter whose height is 1. In this paper, we call an image whose height is 1 flattened image. We use SENet as a CNN model, which is a novel deep learning model proposed by Hu et al. [5]. The experiments comparing the classification accuracies between when using flattened images and when using square images showed the former images were more effective than the latter.

The remaining of this paper is organized as follows. Section 2 describes the method proposed in this paper. Section 3 discusses our experimental setting and results. The conclusions follow in Section 4.

2. Methodology.

An overview of the method is given in Figure 1 and Figure 2. In the first step, we extract the classes.dex file from Android malware and convert it to an RGBA image. Second, we flatten the generated image with height of 1. Finally, we extract feature and classify malware family using SENet whose convolutional filter height is 1.

2.1. Malware visualization.

The image representation of N channel malware is generated in the following procedure. The given malware binary is read as a vector of 8N-bit unsigned integers and then organized into a 2D array, then visualized as an N-channel image, where the width is defined by the file size. The imaging methods ignore a remainder of dividing the file size by the number of channels. After that, to unify each image size, all the images are resized to 96 × 96 squares by the bilinear method which is one of the pixel interpolation methods, which refers to the luminance value of 2 × 2 pixels around the pixel to be obtained and complements it using its weighted average value. Finally, row components of an image are connected in a series for each channel.

2.2. SENet architecture.

The Squeeze-and-Excitation Networks (SENet) architecture was proposed by Hu et al. [5] and is the winning model of ILSVRC in 2017. In this

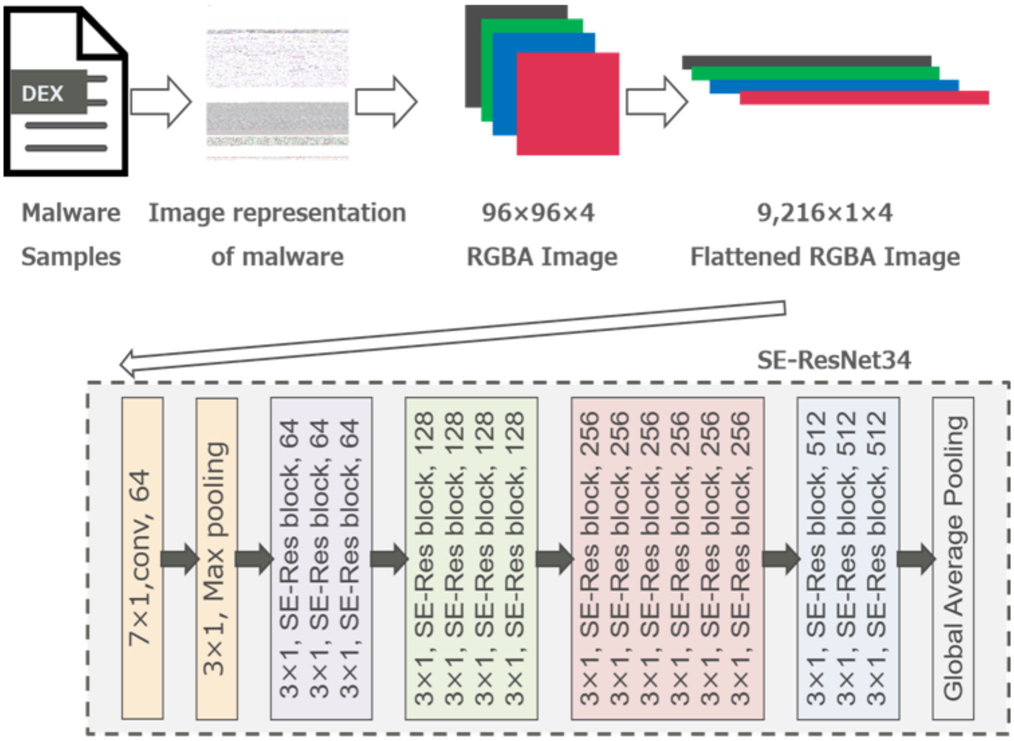


Figure 1. Overview of the proposed method

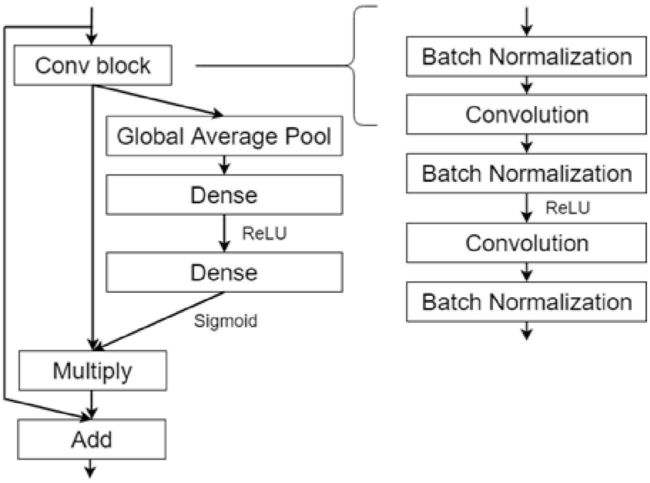


Figure 2. The architecture of the SE-Res block

network, attention mechanism can be added to a given network architecture by a module called SE block. SE block has two steps: Squeeze step and Excitation step. In Squeeze step, global average pooling is applied to the H × W × C feature map to calculate 1 × 1 × C-dimensional features. In the next Excitation step, the dependency between the channels of the feature map is extracted by 1 × 1 convolution. SE block enables feature selection to match the context of the entire image by scaling the feature map using the weight for each channel. According to Hu et al [5], they introduced SE block to ResNet [8] which is the winning model of ILSVRC in 2015 and improve classification accuracy.

In our research, we make a classifier using SE-ResNet34 which introduces SE block to ResNet with 34 layers. Figure 2 illustrates the architecture of the SE-Res block. Conv block includes Batch Normalization layer, Convolution layer, and activation function.

Batch Normalization enables stabilization and speedup of learning. There are many proposal and discussion on which layer to apply batch normalization. He et al. [9] showed that the accuracy can be improved by placing a batch normalization layer before the convolutional layer. Han et al. [10] improved the accuracy by reducing the number of activation functions in Conv block and adding batch normalization layer at the end of the method of He et al. [9]. Thus, we use the method of Han et al. [10] for our Conv block. The remaining SE-Res block layers are like the structure described above. Our model differs from original SE-ResNet34 in a point. We change the height of all filters in the model to 1.

3. **Experiments.** In this section, we conduct experiments comparing classification accuracies between when using flattened representation and when using square representation.

3.1. **Dataset.** In this paper, we use Android Malware Dataset (AMD) [7] provided by Argus Cyber Security Labs of the University of South Florida for research purpose. This dataset consists of 71 families of 24,533 samples of Android malware collected between 2010 and 2016. In our experiments, we use the top seven families which have over 1,000 samples, a total of 18,963 samples because there is a great difference in the number of samples for each family. Table 1 shows the numbers of samples of the top seven families.

Table 1. The numbers of samples included in top seven families in AMD dataset

Family name	Total samples
Airpush	7,843
Fusob	1,275
FakeInst	2,168
Kuguo	1,199
Mecor	1,820
Youmi	1,300
Dowgin	3,358

3.2. **Malware classification.** Using SE-ResNet34, we classified 18,963 samples of malware into seven families. We converted malware files to image representation by the method described in 3.1, resized them to 96 × 96 × 4, and used them as inputs to SE-ResNet34. The flattened 9,216 × 1 × 4 images was also used as inputs to SE-ResNet34. To evaluate the performances of the two cases, we used a stratified 10-fold cross validation and calculated accuracy, precision, recall, and F-measure. Definition 3.1 shows how to calculate accuracy, precision, recall, and F-measure. TP, TN, FP, FN mean, True positive, True negative, False positive, False negative respectively. During training, the network parameters were optimized using Adam optimizer, using a mini-batch size of 64 and using categorical cross entropy as the loss function. When computing the cross entropy, each class is weighed according to the number of malware samples in each family because there is a big difference in the number of malware samples by family.

Definition 3.1.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

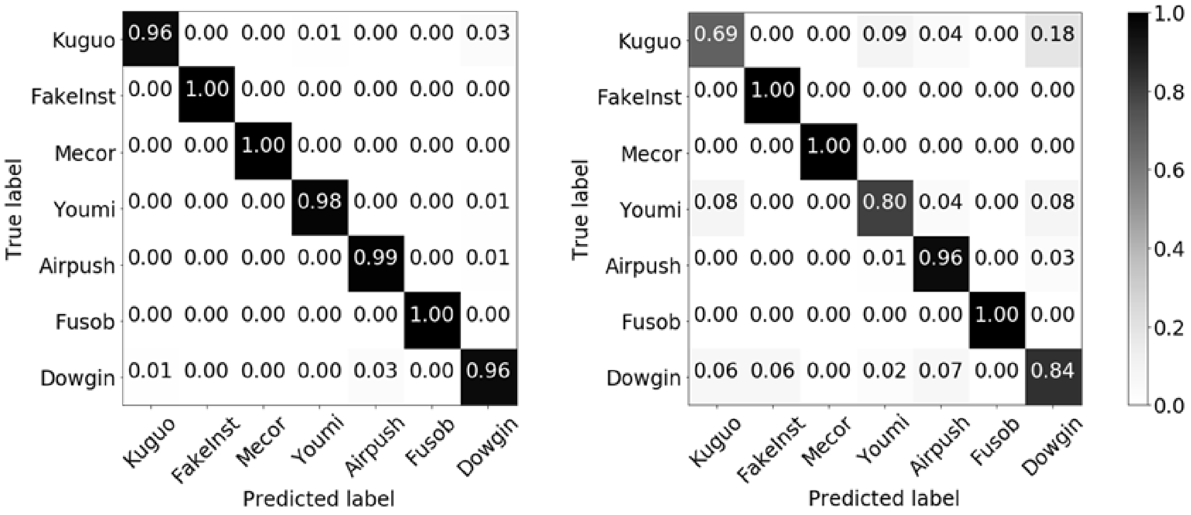
$$Recall = \frac{TP}{TP + FN}$$

$$F\text{-}measure = \frac{2 \times Precision \times Recall}{Precision \times Recall}$$

Table 2 shows the classification results using the square 96 × 96 × 4 RGBA images and the 9,216 × 1 × 4 ones. All four scores (accuracy, precision, recall and F-measure scores) are higher when using the flattened images than using the square ones. Figure 3 represents the confusion matrix obtained by the two cases. In Figure 3(a), Kuguo, Youmi and Dowgin have similar behavior and their classification accuracies are very low compared to other families. Whereas, Figure 3(b) shows that even Kuguo, Youmi and Dowgin families are classified with high accuracies. From these results, we conclude that using fattened images improves the performance of malware classification by comparing the classification accuracies between when using flattened images and when using square ones.

Table 2. Average classification report with SE-ResNet34

	Accuracy	Precision	Recall	F-measure
Square RGBA images	0.9212	0.9206	0.9212	0.9209
Flattened RGBA images	0.9832	0.9832	0.9832	0.9832



(a) 96 × 96 × 4 RGBA images

(b) 9,216 × 1 × 4 flattened RGBA images

Figure 3. Normalization of confusion matrix

4. **Conclusions.** In this paper, we proposed the approach which converts a square image to a height 1 image when classifying malware using image processing technology. We made a classifier using SE-ResNet34 which was a novel deep neural network model and conducted experiments comparing classification accuracies between when using flattened representation and square representation. The results showed that the former representation was more effective than the latter. In the future work, we would like to compare classification accuracies between a deep neural network for computer vision and other deep neural network for time-series data like a recurrent neural network.

REFERENCES

[1] International Data Corporation, *Smartphone OS Market Share, 2017 Q1*, https://www.idc.com/promo/smartphone-market-share/os, Accessed in May 2017.

[2] McAfee Labs, *McAfee Labs Threats Report December 2017*, https://www.mcafee.com/enterprise/en-us/assets/infographics/infographic-threats-report-dec-2017.pdf, Accessed in June 2018.

[3] E. K. Kabanga and C. H. Kin, Malware images classification using convolutional neural network, *Journal of Computer and Communications*, vol.6, no.1, pp.153-158, 2018.

[4] E. Rezende, G. Ruppert, T. Carvalho, A. Theophilo, F. Ramos and P. de Geus, Malicious software classification using VGG16 deep neural network's bottleneck features, in *Information Technology – New Generations. Advances in Intelligent Systems and Computing*, S. Latifi (ed.), Cham, Springer, 2018.

[5] J. Hu, L. Shen and G. Sun, Squeeze-and-excitation networks, *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.7132-7141, 2018.

[6] L. Nataraj, S. Karthikeyan, G. Jacob and B. S. Manjunath, Malware images: Visualization and automatic classification, *Proc. of the 8th International Symposium on Visualization for Cyber Security*, 2011.

[7] F. Wei, Y. Li, S. Roy, X. Ou and W. Zhou, Deep ground truth analysis of current Android malware, in *Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA 2017). Lecture Notes in Computer Science*, M. Polychronakis and M. Meier (eds.), Cham, Springer, 2017.

[8] K. He, X. Zhang, S. Ren and J. Sun, Deep residual learning for image recognition, *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.770-778, 2016.

[9] K. He, X. Zhang, S. Ren and J. Sun, Identity mappings in deep residual networks, in *Computer Vision – ECCV 2016. Lecture Notes in Computer Science*, B. Leibe, J. Matas, N. Sebe and M. Welling (eds.), Cham, Springer, 2016.

[10] D. Han, J. Kim and J. Kim, Deep pyramidal residual networks, *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.5927-5935, 2017.