Malware Detection with Malware Images using Deep Learning Techniques

Ke He

Dong-seong Kim

Visualize a malware file as an image and use CNN to classify the image

Recurrent Neural Network in MDS

- Most frequently used deep learning technique in malware detection
- Works very well in text classification
- Programs can be thought of as text with API calls instead of words.
- Can be bypassed by adding redundant API calls which tricks the RNN[1]
- RNN remembers the language environment

1. Hu, W., & Tan, Y. (2017). Black-box attacks against RNN based malware detection algorithms. arXiv preprint arXiv:1705.08131.

Convolutional Neural Network in MDS

- Often requires static/dynamic analysis to extract features.
  - doesn't take advantage of CNN's innate ability to extract high level features
  - We want end-to-end classification.
- Redundant API are equivalent to shifts/distortion of features in image, which a CNN is designed to recognise.
- Need to solve two major problems
  - Numerically represent malware files
  - Deal with variable size input

System Overview

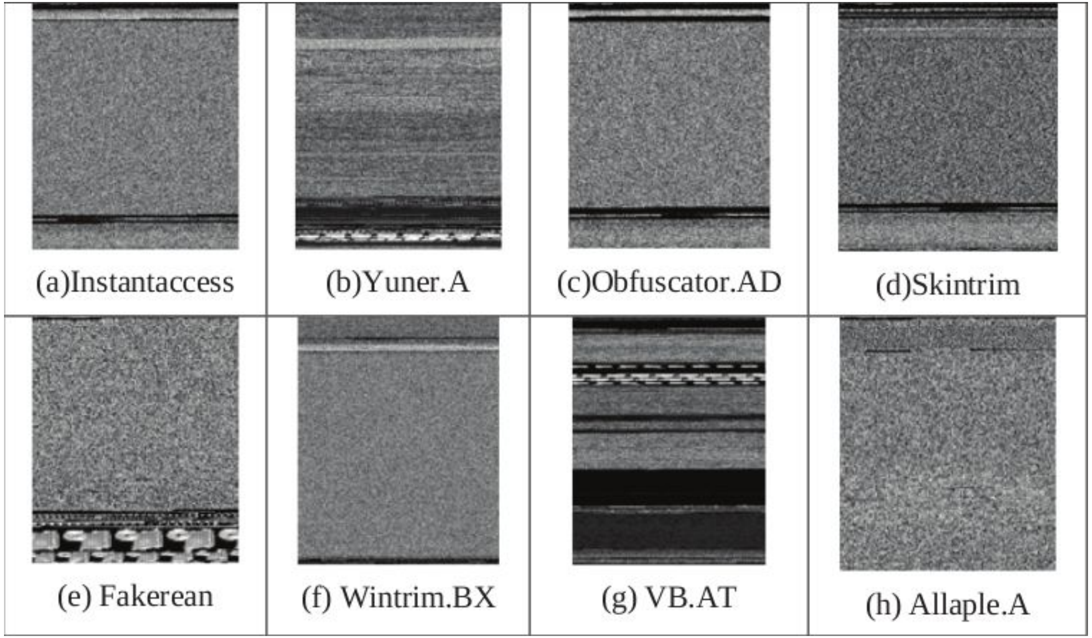
Preprocessor
- converts binary files to grayscale/RGB images

Classifier
- trains the CNN with malware image and apply spatial pyramid pooling (SPP)

Evaluator
- Evaluates the performance of trained model

Preprocessor

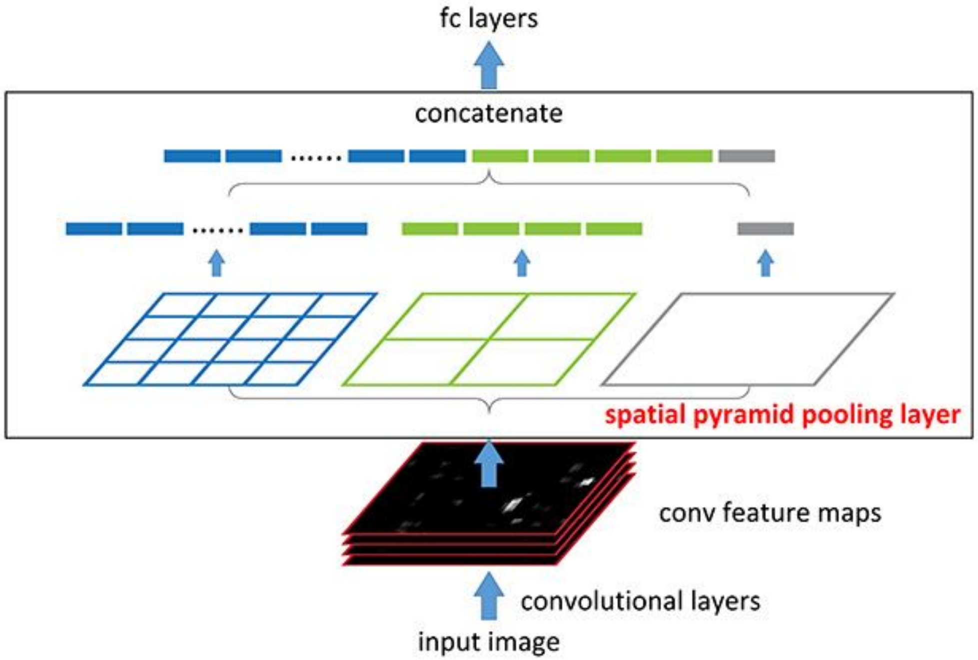
- Converts each byte (0~255) to a pixel (0~255) for grayscale images
- Converts a group of 3 bytes to a pixel for RGB images
- Malware belonging to the same family exhibits similar texture[1]
- All images have fixed width of 1920



1. Nataraj, L., Karthikeyan, S., Jacob, G., & Manjunath, B. S. (2011, July). Malware images: visualization and automatic classification. In Proceedings of the 8th international symposium on visualization for cyber security (p. 4). ACM.

Classifier

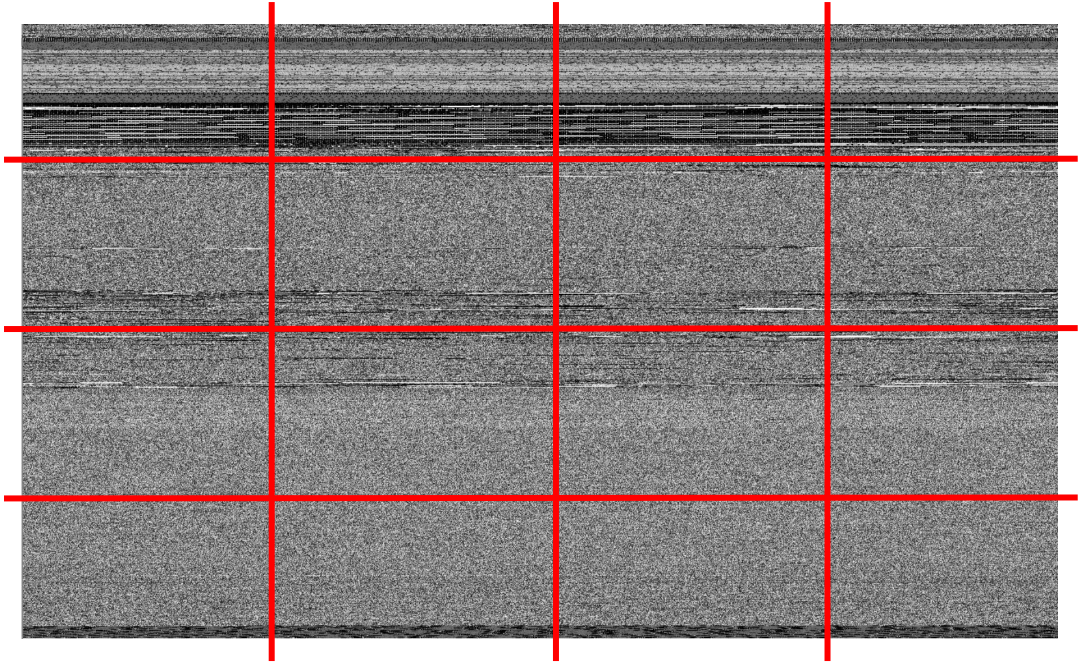
- Use resnet50[1] or a plain network with 3 layers
- To deal with variable size inputs, spatial pyramid pooling[2] is applied
- With SPP, all files are resized to (256, 256), (256, 151), (256, 433) representing the 3 main sizes of malware for training.
- Without SPP, all samples are resized to a fixed size (256,256) using bilinear interpolation



1. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
2. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2014. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition.

Evaluator

- Classifies with given threshold.
- Calculates the confusion matrix
- In practice, some files are extremely large, thus we use the “divide and conquer” method.



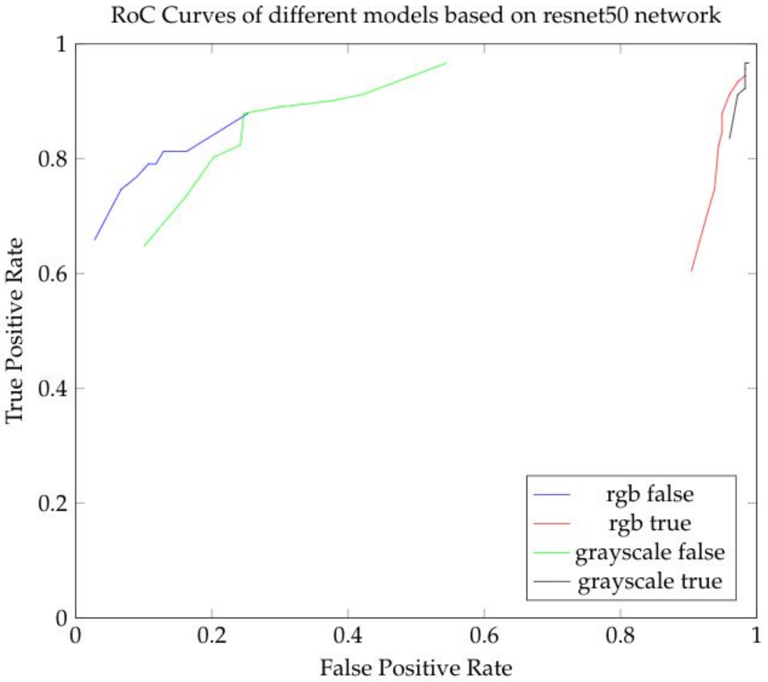
Summary of Experiment

- Dataset Andro-dumpsys[1]: Android platform, 906 malware and 1776 benign file.

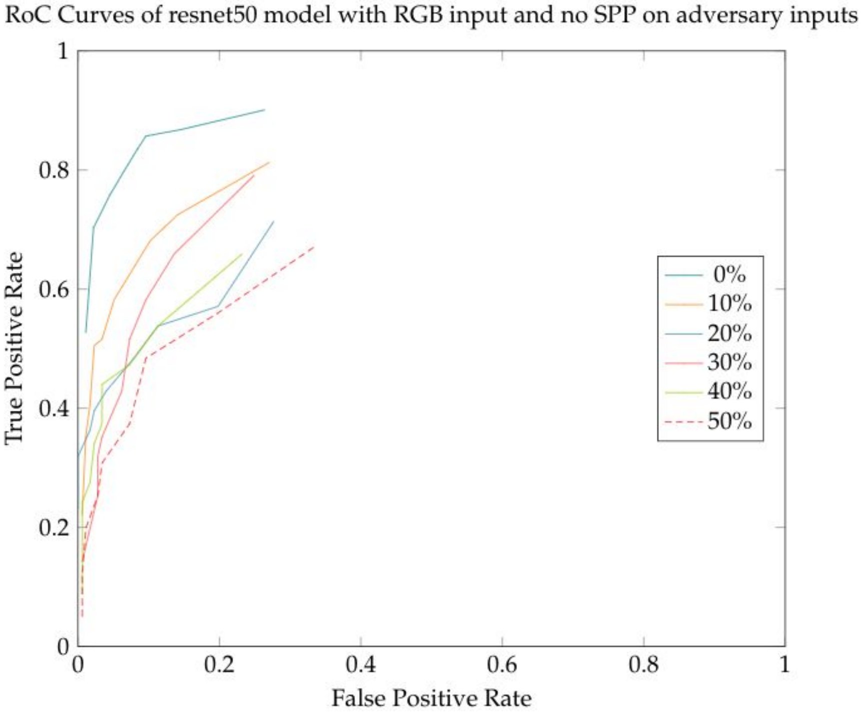
Experimental Parameters		
Parameter	Level 1	Level 2
Preprocessing	Greyscale	RGB
Classifier	3 layer Plain	Resnet50
SPP	No SPP	SPP with bin size (1, 2, 4)

1. Jang, J. W., Kang, H., Woo, J., Mohaisen, A., & Kim, H. K. (2016). Andro-dumpsys: anti-malware system based on the similarity of malware creator and malware centric information. computers & security, 58, 125-138.

Results

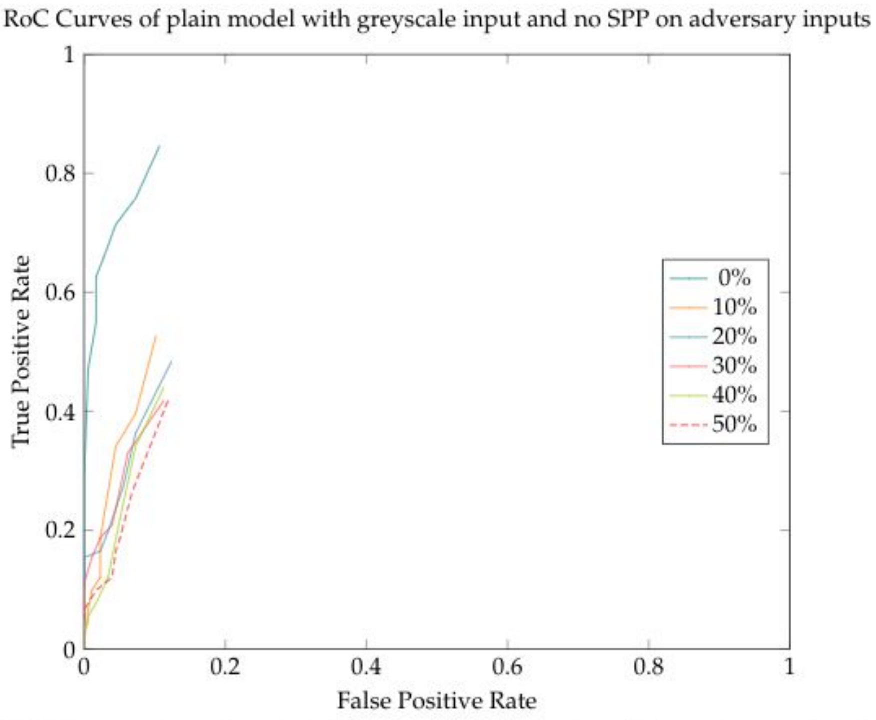


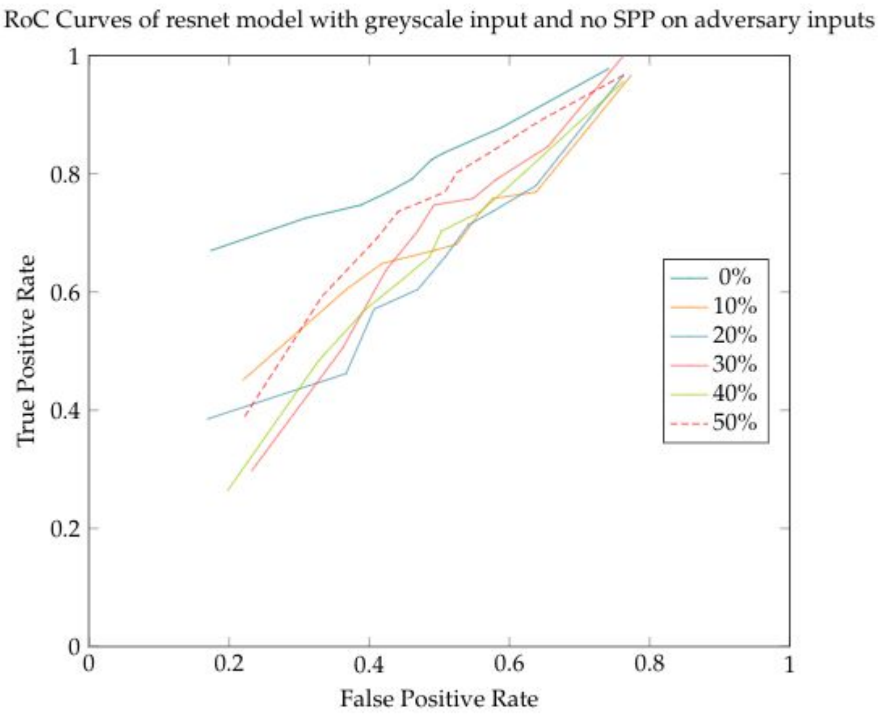
RoC Curves of different models based on resnet50 network
True Positive Rate
False Positive Rate
0
0.2
0.4
0.6
0.8
1
rgb false
rgb true
grayscale false
grayscale true



RoC Curves of resnet50 model with RGB input and no SPP on adversary inputs
True Positive Rate
False Positive Rate
0
0.2
0.4
0.6
0.8
1
0%
10%
20%
30%
40%
50%

Results


RoC Curves of plain model with greyscale input and no SPP on adversary inputs
True Positive Rate
False Positive Rate
0
0.2
0.4
0.6
0.8
1
0%
10%
20%
30%
40%
50%


RoC Curves of resnet model with greyscale input and no SPP on adversary inputs
True Positive Rate
False Positive Rate
0
0.2
0.4
0.6
0.8
1
0%
10%
20%
30%
40%
50%

Findings

- SPP with divide and conquer works poorly
  - Features are being splitted, which implies unlike traditional image classification where features are focused at one spot, malware images have features scattered around the image.
  - Images are varying too much in its size and aspect ratio that SPP cannot handle.
- Greyscale is more resistant to redundant API calls
  - Redundant APIs shifts subsequent pixels in RGB, can be fixed by introducing a padding scheme
- Resizing with bilinear interpolation shows promising results against redundant API injection.

Future Work

- Make SPP great again
- Other types of resize/compression method
- Fine tune the network
- Add distortions in input image/adversarial training
- API level image transformation

~ END ~