2018 IEEE Symposium on Security and Privacy Workshops

File Fragment Classification Using Grayscale Image Conversion and Deep Learning in Digital Forensics

Qian Chen[1], Qing Liao[1], Zoe L. Jiang[1]*, Junbin Fang[2]*, Siuming Yiu[3], Guikai Xi[2], Rong Li[2], Zhengzhong Yi[1], Xuan Wang[1], Lucas C.K. Hui[4], Dong Liu[5], En Zhang[5]
[1]*School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China*
[2]*Guangdong Provincial Engineering Technology Research Center on VLC, Guangzhou Municipal Key Laboratory of Engineering Technology on VLC, and the Department of Optoelectronic Engineering, Jinan University, Guangzhou, China*
[3]*The University of Hong Kong, Hong Kong*
[4]*Hong Kong Applied Science and Technology Research Institute, Hong Kong*
[5]*School of Computer and information Engineering, Henan Normal University, Xinxiang, 453007, China*

***Abstract*—File fragment classification is an important step in digital forensics. The most popular method is based on traditional machine learning by extracting features like N-gram, Shannon entropy or Hamming weights. However, these features are far from enough to classify file fragments. In this paper, we propose a novel scheme based on fragment-to-grayscale image conversion and deep learning to extract more hidden features and therefore improve the accuracy of classification. Benefit from the multi-layered feature maps, our deep convolution neural network (CNN) model can extract nearly ten thousands of features through the non-linear connections between neurons. Our proposed CNN model was trained and tested on the public dataset GovDocs. The experiments results show that we can achieve 70.9% accuracy in classification, which is higher than those of existing works.**

***Keywords*-Digital forensics, file fragments classification, deep learning, grayscale image**

I. Introduction

File fragment classification plays an important role in digital forensics. Evidence could be found in deleted/hidden fragments. File carving technologies are usually applied to reconstruct files from these fragments for further forensic investigation. Correctly classifying these fragments is a necessary step for effective file carving. Otherwise, file carving has to try all the combinations of a huge number of fragments and the computational cost will be massive [1]. Besides, the accuracy of file fragment classification affects also the accuracy of file carving significantly as misclassified fragments represent the noise of the input.

Early researches on file fragment classification utilize the full file extension, the magic number or the metadata of files to classify file fragments. These methods only have high classification accuracy when the metadata can be found and extracted from storage medium with the fragments. Thus, they have less practical application in digital forensics as the metadata of file fragments is usually missing in real cases.

* Corresponding author:zoeljiang@gmail.com junbinfang@gmail.com

In recent years, content-based file fragment classification algorithms extracting the N-gram, Shannon entropy, Hamming weight and statistical regularities of bytes have been proposed for file fragments of incomplete file. In some schemes, traditional machine learning techniques are deployed to improve the performance of these classification algorithms. However, for high entropy files like compressed files (e.g., .zip file or .jpg file) and encrypted file, the accuracy is not as good as expected. For the files generated by the same compression algorithm, the classification result is also not good if the file header or other meta information is lost.

In this paper, a novel file fragment classification scheme using fragment-to-grayscale image conversion and deep learning is proposed. By extracting the high-dimensional features of file fragments and utilizing the advantage of local connection and weight sharing of CNN, the proposed scheme can achieve a high accuracy of fragment classification, even for the files that are not easy to be classified in previous schemes, such as compressed files and composite files.

The proposed scheme was evaluated experimentally using 16 types of files in a public dataset GovDocs, which was partially used in Sceadan [2]. Note that the files under test have no metadata and some hard-to-solve files in the previous researches were added to the dataset. Experimental results show that the proposed scheme could achieve a classification accuracy of 70.9%, which is higher than that of the previous works. Besides, using grayscale images for features extraction, the proposed scheme can distinguish several types of highly similar file fragments correctly. In addition to the texture features of the image, the hidden features of the image can be extracted by our deep learning model. Furthermore, the reasons for some misclassified cases were analyzed using confusion matrix and the recall and precision rates of each type of files.

The rest of the paper is organized as follows. In Section II,

DOI 10.1109/SPW.2018.00029

we review related work about file fragment classification of digital forensics. Section III discusses the concepts of file type and data type, followed with a simple introduction of our CNN model. In Section IV, the proposed scheme is introduced in details including the fragment-to-grayscale image conversion and the further classification based on our CNN model. Section V shows the experimental results with analysis and discussion. We conclude our paper and discuss the future work in Section VI.

II. Related work

File type recognition has been studied for a long time. Early research focused on complete files, where file type can be confirmed by file extension, extracted features of magic numbers, or metadata which are fixed-length binary strings of file header or trailer. However, not all file types have extensions or magic numbers, and file extensions and magic numbers are easily tampered with by criminals.

File-content based method was proposed by Mcdaniel [3], using the byte frequency distribution (BFD) or byte frequency cross-correlation (BFC) to identify the file type. The average classification accuracy is only around 27% with BFD and around 46% with BFC. The accuracy can over 90% when the file headers which contain magic numbers can be found and included. The similar performance can be found in the scheme proposed by Li [4], which can achieve an accuracy of 99% using the first 20 bytes of completed files. However, in practical, it's rare to obtain complete, unmodified and well preserved files. Therefore, although these researches can achieve the accuracy at about 90% (Mcdaniel [3] and Li [4]), it is of no great practical significance in actual digital forensics work.

From the practical perspective, the classification of incompletely file fragments is gradually getting important in digital forensics. In general, some file features such as metadata and file extension are unavailable. Thus, the accuracy of file fragments classification is usually not as high as that of the early research works. To improve the accuracy, researchers tried to extract more different features and use methods of traditional machine learning.

Veenman [5] selected 28 types of file fragments of a private dataset and extracted features such as 1-gram, Shannon entropy and Kolmogorov complexity to classified file fragments with Linear Discriminant Analysis (LDA) which is also called Fisher Linear Discriminant (FLD). The basic idea of this method is to use the Fisher criterion function to choose the extreme value vector as the optimal projection direction so that the projection distances of different classes are the farthest. The classification accuracy is about 45% on average.

Calhoun and Coles [6] extracted more features similar to Veenman's [5]. They carried out experiments with 4 types of files (JPG, BMP, GIF, and PDF) in a small dataset and achieved a relatively high accuracy (about 88.3%) with two algorithms for file type classification: Fishers linear discriminant and longest common subsequences. However, since their experiments only include 4 types of files, it is not known that whether the method can perform well on several types of files that are difficult to distinguish in file fragment classifications.

And, they incline to use public datasets for fair comparison of results [7].

Axelsson [8] selected 28 types of file fragments of a public dataset and extracted the Normalized Compression Distance (NCD) as the feature to classify file fragments. The idea behind NCD is that when data vectors are compressed individually and concatenated, the normalized distance between them can be used as a measure of similarity. They used k-nearest neighbors (k-NN) algorithm and achieved $32\% \sim 36\%$ accuracy of classification.

Fitzgerald [9] selected 24 types of file fragments of a public dataset and extracted features such as 1-gram, 2-gram, Shannon entropy, and Hamming weights. Using support vector machines combined with the bag-of-words model which are widely used in natural language processing to improve the classification accuracy, the method can achieve an accuracy of nearly 47.5%.

Beebe et al. proposed a scheme named Sceadan [2]. In their experiments, both file type fragments and data type fragments were selected from a public dataset and a private dataset. Note that 40% fragments were selected from the private dataset. They used the combination of 1-gram and 2-gram features in the linear kernel of support vector machine (SVM) and compared different kernel functions of SVM algorithm. The accuracy of this scheme is 73.45%.

III. Definitions and basic model

In this section, the difference between the definitions of file type and data type is explained, and the basic model of CNN is introduced.

A. File type vs data type

The classification of file and data types plays an important role in information security and digital forensics. Therefore, the concepts of file type and data type should be correctly distinguished. In this paper, we introduce the notion by Erbacher and Mulholland [10]:

- Data type: Indicative of the type of data embedded in a file.
- File type: The overall type of file. This is often indicated by the application used to create or access the file.

Note that classifying file types is more challenging since different types of files may contain same data type and one type of file may contain a variety of data types, e.g., composite files such as PPT, PDF and other types of files embedded with JPG and other complex data types.

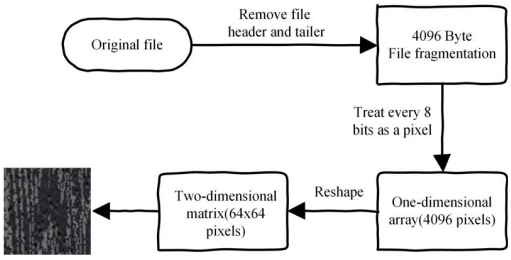


Figure 1. Converting fragments into grayscale images

B. Convolution neural network:

Deep Learning can use the output of the layer in the network as another expression of the raw data so that it can be considered as the feature maps learned through each layer in the network. CNN is one kind of deep learning algorithm which widely used in image classification. Due to shared weight and local perception, CNN has a great advantage in image processing.

Lecun [11] proposed LeNet model for document recognition in 1998, which is the formal formation of CNN. Alex [12] used CNN in the ImageNet contest in 2012 with the accuracy exceeding the second nearly 10%, which laid a solid foundation of CNN. The CNN model that structured by Alex in the Imagenet is later named as Alexnet. From then on, CNN started to shock the world and applied in many fields and achieved very good results. Our CNN model is based on Alex's model, and we made some modifications and optimizations to adapt to our application.

$$x_j^l = f(\sum_{i \in M_j} x_i^{l-1} * k_{ij}^l + b_j^l) \tag{1}$$

Eq. 1 shows the calculation from the previous $l-1$ layer to the convolutional layer. Where M_j represents the selected region of the input layer, while K_{ij} represents the weight parameters, and $f()$ represents the ReLU active function. In addition, feature map shares the same kernel and bias in CNN.

IV. The proposed scheme

In the proposed scheme, file fragments are first converted into grayscale images, and then deep learning is utilized to extract more hidden features of the images to improve the performance of file fragments classification.

A. Fragment-to-grayscale image conversion

Fig. 1 shows the processing procedures of converting file fragments into grayscale images. Conti et al. [13] proved that the texture features are determined by the types of data structures. Therefore, data fragments from different types of files may have different texture features in the corresponding grayscale images. However, Conti only proved the feasibility of classification from intuitive visual expression without deep research on how to distinguish different types of file fragments. Indeed, different file types cannot be simply classified by different texture features since there is little difference in the structural features of the high entropy files like compressed files or composite files.

Nataraj et al. [14] proposed a traditional machine learning method to extract features of the image from the complete files and applied it for malware identification. What we are trying to solve is to classify different types of file fragments in digital forensics where the metadata and magic numbers may be lost, or even be maliciously tampered. That means, we usually do not have complete files, and the fragments are randomly sorted. Unlike traditional machine learning that extract features such as byte-frequency statistics, we try to use deep learning to extract better feature maps which are hidden behind grayscale images. At the same time, although high entropy file fragments are randomly distributed in grayscale images and cannot be easily distinguished, it can be distinguished with randomness using NIST Statistical Test [15]. Extracting features such as 1-Gram, 2-Gram are also the part of NIST's frequency test.

In our scheme, we randomly choose 16 types of file fragments from the public dataset GovDocs to extracting hidden features, such as different texture features, random features, and compressibility for classification.

B. The CNN Network Structure and Approach

Deep learning emphasizes the importance of feature learning. That is, through layer-by-layer feature transformation, the feature representation of the sample in the original map is transformed into a new feature map, which making classification or prediction easier. Compared with the method of constructing features by artificial rules, using big data to learn features makes it possible to describe the inherent information of original data. The convolution of the images is the weighted sum of pixels, and the matrix composed of different weights is called convolution kernel. In the convolutional neural network, it is also called filter. In the neural network, the neurons in the filter are the weight of the convolutional neural network. In each convolutional layer, the convolution operation on the image is actually looking for the output of each neuron. A filter corresponds to a neuron. Applying the same filter to different areas of the image, we can get one of the feature maps. Due to the deep feature learning rather than the manual extraction of features, we can finally obtain better classification accuracy. The specifically CNN network structure and approach will be described below.

1) Loss function: Loss function is used to estimate the degree of inconsistency between the predicted value $f(x)$ of the model and the true value y. It is a real valued function and is usually expressed by $L(y, f(x))$. The smaller the loss function is, the stronger the robustness it is. The loss function is the core part of the empirical risk function

and an important part of the structural risk function. The structural risk function of the model includes empirical risk and regular. This paper uses the Adam (Adaptive Moment Estimation) optimization method to dynamically adjust the learning rate of each parameter to minimize the total loss function.

- Cross-entropy cost function

Cross-entropy is used to evaluate the degree of difference between the current probability distribution and the real probability distribution. Reducing the cross-entropy loss can improve the prediction accuracy of the model. The cross-entropy cost can be calculated with:

$$L_1 = -\frac{1}{n} * \sum_x [y * ln(a) + (1-y) * ln(1-a)] \quad (2)$$

$$a = \sigma(z), z = \sum \omega_j * x_j + b \quad (3)$$

- L2 Regularization

$$L = L_1 + \frac{1}{2} * \beta * \sum ||\omega||^2 \quad (4)$$

Where L_1 represents the original loss function, in this paper is cross entropy loss function, and β is a hyper-parameter use to adjust the proportion of the two losses. In neural networks, regularization networks tend to let weights be smaller. In the case of small weights, the random changes of x do not have much impact on the neural network model, so it will be less affected by noise. Without the regularization of neural networks, the weights are large, and it is easy to adapt to the data through larger changes of the model and will learn the local noise more easily.

2) The optimized CNN network Structure: Fig. 2 shows the CNN network structure we modified and optimized for file fragments classification. The first layer of the convolutional layers uses a convolution kernel with a scale of 1x1. The convolution filter of this scale does not consider the relationship between local information. But filter and non-linear activation function enhance the expression ability of networks. It can also complicate the network structure using many pipelines. Regarding the main features extracted in traditional machine learning: N-Gram, it's concept is consistent with the local connection and weight sharing of CNN. However, the features of traditional machine learning are extracted manually, which depends largely on experience. Deep learning can learn more useful features by constructing hidden layers and training massive of datasets, which ultimately improve the accuracy of classification. There are different sizes of filters in each of layers which can train the best-fit feature maps through gradient descent and reverse derivation.

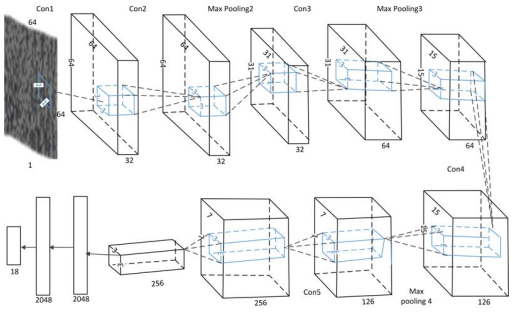


Figure 2. The modified and optimized CNN network structure.

V. Experimental results and discussion

A. Dataset

Garfinkel [7] suggested that the use of standard dataset not only allow researchers to validate other result, but also to build upon them. In our experiments, we use a subset of the public dataset GovDocs [7]. The Govdocs [7] dataset contains a set of 1,000 directories, each of which includes 1,000 files. We randomly select 16 types of these files to create our experimental dataset. 70% of them (811,922 file fragments) are training set and 30% (312,361 file fragments) are testing data. While the size of one physical sector on a hard disk is 512 bytes, files stored on a hard disk is organized as clusters, which size is usually 4,096 bytes. Therefore, in our experiments, we sliced the files under tests into file fragments of 4,096 bytes, and the file header and trailer were removed to simulate the real cases. Note that composite files and several different types of compressed files which used the same compression algorithm are involved in our dataset and classifying these file types is more challenging than simple data type classification.

B. Performance evaluation index

To evaluate the performance of our scheme, we introduce several parameters: accuracy of file fragments classification, recall rate, and precision, which are defined as follows.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (5)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (6)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (7)$$

where,

TP means the forecast category which is positive and actually positive,

TN means the forecast category which is negative and actually negative,

FP means the forecast category which is positive and actually negative, and

FN means the forecast category which is negative and actually positive.

C. Experiment results

We first transform one-dimensional data bytes in file fragments into a specific size of a two-dimensional matrix. The data bytes in the matrix are regarded as the pixels in the grayscale images, as shown in Fig. 3. Some of the grayscale images have obvious texture features different from the others, while some of them look quite similar, such as the grayscale images (c),(d),(e),(h) and (k) in Fig. 3. Indeed, (h) is JPG files which using lossy compression algorithm, while (d) is GIF files based on the LZW algorithm. And, (c), (e) and (k) were produced using Phil Katz's Deflate compression algorithm. Deflate is a lossless data compression algorithm that uses both the LZ77 algorithm and Huffman Coding.

Then, we use CNN to train and test the file fragments which have been converted to grayscale images. We modified and optimized CNN model to a certain extent to make it more suitable for classifying file fragments in digital forensics. In our model, the kernel size of the first layer is 1, which can be seen as unigram feature is extracted, while unigram feature is a particularly important feature in the previous classification of file fragments. And the following layers take other kernel sizes to extract more features by making use of the non-linear connection between neurons. We also optimize the model, including add L2 loss function to prevent overfitting.

The recall and precision of classifying each type of file fragments are shown in Fig. 4 as a scatter diagram. And the average classification accuracy of all 16 types is also shown in Table I with comparison with some previous works.

Since the datasets used in these previous researches are not exactly the same, we list the information including "No. of file types" and "Source of dataset".

Note that all the datasets in these works have compressed or composite files. And, the datasets of Axelsson [8], Fitzgerald [9] and Veenman [5] are all selected from the public dataset GovDocs, the same one used in our experiments.

As shown in Table I, our scheme can achieve better accuracy (70.9%) compared with the existing works except for the result of Sceadan [2] in which nearly 40% of their data were selected from private datasets. In addition, there are many factors affecting the results of the experiments, which will be analyzed below in details.

D. Analysis and discussion

It is very helpful to the future work for analyzing the reasons of misclassification. Table I and Table II report the average classification accuracy of all types and the full confusion matrix, respectively. The classification accuracy in this paper is significantly better than random chance(1/16=6.25%) and has a certain degree of improvement over the previous work. In this paper, we has some highly visual expression similar file fragments like (c),(e),(h),(k),(d) in Fig. 3. They are easily be confused in previous work because they are embedded or compressed high-entropy files. Look at DOC and DOCX files, they are very similar in naming, but their compression algorithms are completely different. In DOC, Microsoft used binary to store, whereas in DOCX Microsoft began using XML. So DOCX actually became a packaged compressed file, so these two files will not be confused with each other. Due to the problem of number distribution is different of each file in GovDocs, this two classes accuracies are not particularly prominent. CNN can extract high-dimensional features, PNG and GZ, this two kinds of high entropy compressed files can be separated in some degree. At the same time, we found that many files are misclassified as GZ or PPT. Because GZ is a compressed file and PPT is a composite file type, they may embed various of different type of file fragments. In the case of no file header or tailer, how to get more prominent results in high entropy files and composite files is our future work.

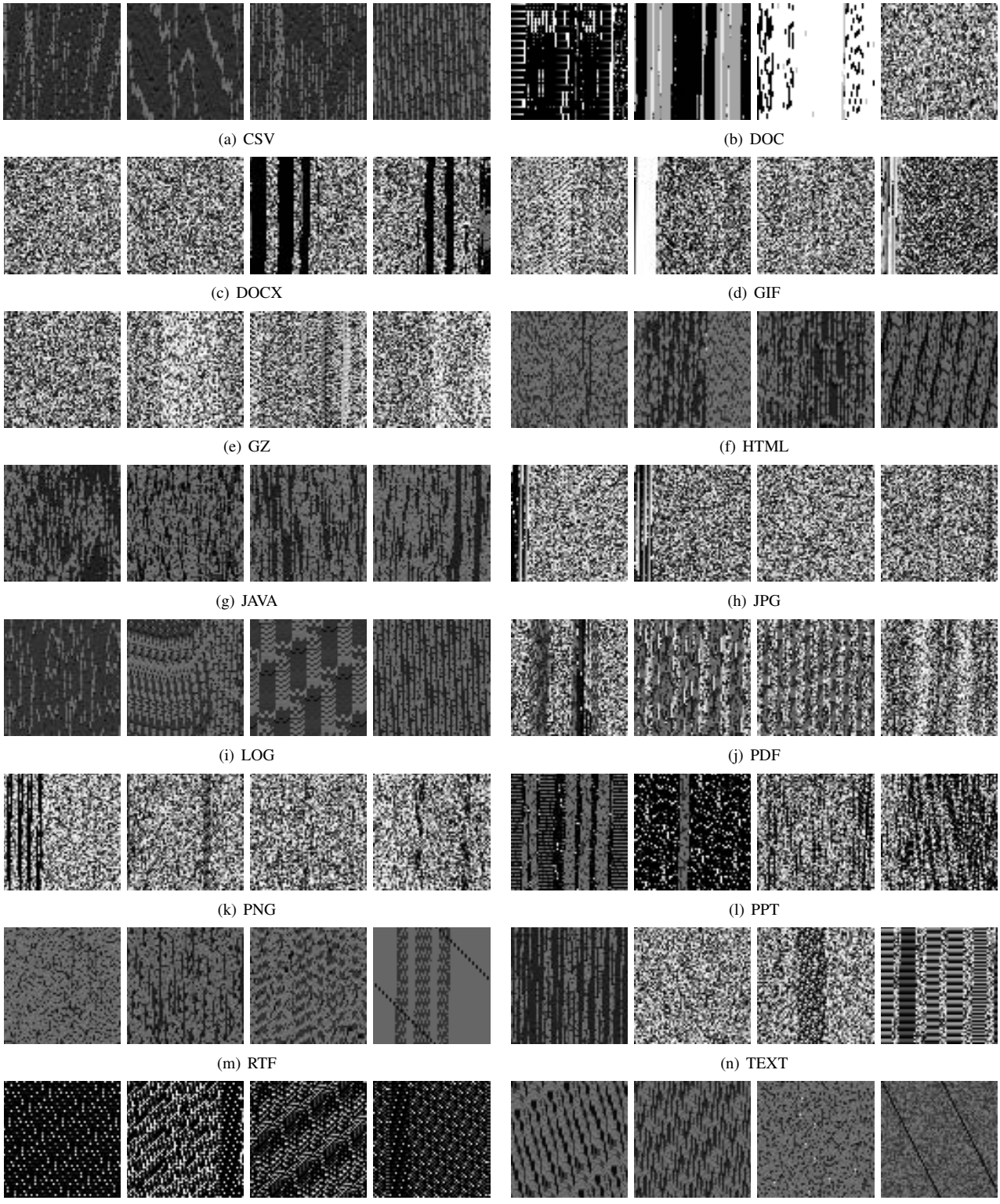
The goal of this paper is to verify the feasibility of the idea that whether grayscale images and deep learning can be used in file fragments classification of digital forensics. The results show that this approach is feasible and can get a better result. However, the number distributions of different types of files are not the same in GovDocs, for example, when the number of files is very small, it will affect the accuracy of the final classification results. This paper does not optimize the dataset itself, the bias may affect the accuracy of our model. In the future work, we can focus on the optimization of datasets and model to further improve the classification accuracy.

VI. Conclusions

Classifying file fragments is an important and difficult problem in file carving. We made several important contributions to this problem. We creatively used grayscale images and deep learning for improving the accuracy of file fragments classification. Comparing our results with several representative results in previous work, we show that our method achieves a much higher accuracy, in general. We also provide a detailed analysis of the effectiveness of using grayscale for file classification in order to understand which file type can provide better results using these techniques. It seems that our approach is promising. It is worth to further optimize the model and techniques as our future work.

Acknowledgment

This work was partially supported by National Natural Science Foundation of China (No. 61771222, 61772233), National Key Research and Development Program of China (No. 2017YFB0803002, 2017YFB0802204), Science and Technology Projects of Guangdong Province (No. 2016A010101017), Project of Guangdong High



(a) CSV (b) DOC

(c) DOCX (d) GIF

(e) GZ (f) HTML

(g) JAVA (h) JPG

(i) LOG (j) PDF

(k) PNG (l) PPT

(m) RTF (n) TEXT

(o) XLS (p) XML

Figure 3. Grayscale with 64*64 matrix of 16 types file fragments with 64*64 matrix

Table I
COMPARISON OF CLASSIFICATION ACCURACY (* 40% OF DATA FROM PRIVATE DATASETS)

Scheme	No. of file types	With compressed or composite file?	Source of dataset	Average accuracy
This paper	16	Yes	public	70.9%
Axelsson. [8]	28	Yes	public	32%-36%
Fitzgerald [9]	24	Yes	public	47.5%
Veenman [5]	28	Yes	private	45%
Xu [16]	29	Yes	public	39.7%-54.7%
Sceadan [2]	38	Yes	public+private*	73.45%

Table II
CONFUSION MATRIX

	CSV	DOC	DOCX	GIF	GZ	HTML	JAVA	JPG	LOG	PDF	PNG	PPT	RTF	TEXT	XLS	XML
CSV	97%															
DOC		36%			28%							25%		7%		
DOCX			27%		27%							33%		13%		
GIF				28%	48%							24%				
GZ					92%							7%				
HTML	12%	6%				46%			7%					12%		6%
JAVA						10%	65%		13%							7%
JPG								15%				81%				7%
LOG									97%							
PDF										38%		29%				
PNG					16%						21%	61%				
PPT					16%							78%				
RTF		15%				13%							71%			
TEXT					18%							67%		3%		
XLS															92%	
XML		8%											14%			92%

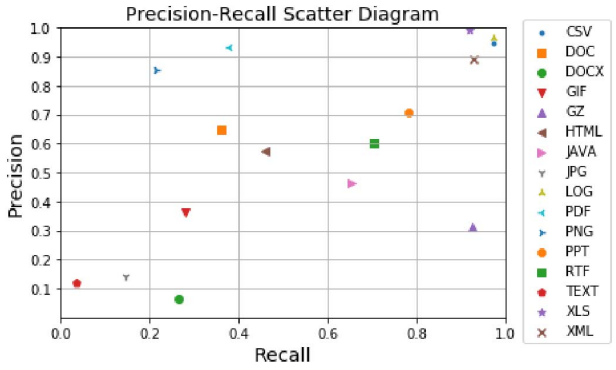


Figure 4. Precesion-Recall Scatter Diagram of 16 Classes.

Education (No. YQ2015018), Science and Technology Project of Guangzhou (No. 201707010253, 201704030105, 201605030002), Key Technology Program of Shenzhen, China (No. JSGG20160427185010977) and Basic Research Project of Shenzhen, China (No. JCYJ20160318094015947, JCYJ20160322114027138).

REFERENCES

[1] V. Roussev and S. L. Garfinkel, "File fragment classification-the case for specialized approaches," in *International IEEE Workshop on Systematic Approaches To Digital Forensic Engineering*, 2009, pp. 3–14.

[2] N. L. Beebe, L. A. Maddox, L. Liu, and M. Sun, "Sceadan: Using concatenated n-gram vectors for improved file and data type classification," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 9, pp. 1519–1530, 2013.

[3] M. Mcdaniel and M. H. Heydari, "Content based file type detection algorithms," in *Hawaii International Conference on System Sciences*, 2003, p. 10 pp.

[4] W. J. Li, K. Wang, S. J. Stolfo, and B. Herzog, "Fileprints: identifying file types by n-gram analysis," in *Information Assurance Workshop, 2005. Iaw '05. Proceedings From the Sixth IEEE Smc*, 2005, pp. 64–71.

[5] C. J. Veenman, "Statistical disk cluster classification for file carving," in *International Symposium on Information Assurance and Security*, 2007, pp. 393–398.

[6] W. C. Calhoun and D. Coles, "Predicting the types of file fragments," *Digital Investigation*, vol. 5, pp. S14–S20, 2008.

[7] S. Garfinkel, P. Farrell, V. Roussev, and G. Dinolt, "Bringing science to digital forensics with standardized forensic corpora," *Digital Investigation the International Journal of Digital Forensics and Incident Response*, vol. 6, pp. S2–S11, 2009.

[8] S. Axelsson, "The normalised compression distance as a file fragment classifier," *Digital Investigation*, vol. 7, no. 8, pp. S24–S31, 2010.

[9] S. Fitzgerald, G. Mathews, C. Morris, and O. Zhulyn, "Using nlp techniques for file fragment classification," *Digital Investigation*, vol. 9, no. 15, pp. S44–S49, 2012.

[10] R. F. Erbacher and J. Mulholland, "Identification and localization of data types within large-scale file systems," in *International Workshop on Systematic Approaches To Digital Forensic Engineering*, 2007, pp. 55–70.

[11] Y. Lcun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *International Conference on Neural Information Processing Systems*, 2012, pp. 1097–1105.

[13] G. Conti, S. Bratus, A. Shubina, A. Lichtenberg, R. Ragsdale, R. Perez-Alemany, B. Sangster, and M. Supan, "A visual study of primitive binary fragment types," 2010.

[14] L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, "Malware images:visualization and automatic classification," in *International Symposium on Visualization for Cyber Security*, 2011, pp. 1–7.

[15] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, and E. Barker, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," *Applied Physics Letters*, vol. 22, no. 7, pp. 1645–179, 2015.

[16] T. Xu, M. Xu, Y. Ren, J. Xu, H. Zhang, and N. Zheng, "A file fragment classification method based on grayscale image," *Journal of Computers*, vol. 9, no. 8, 2014.