Dictionary based Compression Type Classification using a CNN Architecture

Hyewon Song, Beom Kwon, Seongmin Lee and Sanghoon Lee
Electrical and Electronic Department, Yonsei University, Seoul, South Korea
E-mail: {shw3164, hsm260, lseong721, slee}@yonsei.ac.kr

***Abstract*—As digital devices which are capable of viewing contents easily such as mobile phones and tablet PCs have become widespread, the number of digital crimes using these digital contents also increases. Usually, the data which can be the evidence of crimes is compressed and the header of data is damaged to conceal the contents. Therefore, it is necessary to identify the characteristics of the entire bits of the compressed data to discriminate the compression type not using the header of the data. In this paper, we propose a method for distinguishing 16 dictionary-based compression types. We utilize 5-layered Convolutional Neural Network (CNN) for classification of compression type using Spatial Pyramid Pooling (SPP) layer. We evaluate our proposed method on the Wikileaks Dataset, which is a text file database. The average accuracy of 16 dictionary-based compression algorithms is 99%. We expect that our proposed method will be useful for providing evidence for Digital Forensics.**

I. Introduction

With the advent of the digital age, people usually receive and send data through digital devices in daily life. As digital culture permeates a lifetime, the number of digital crimes is increasing day by day. Digital crime is more difficult to punish because of the diverse ways of concealing data which can be evidence. One way to conceal the evidence is to compress the data and delete the header for making it harder to be decoded. Modification or deletion of the header of the compressed data which has information about the compression type prevents the original data to be known. It is necessary to prepare the method of decoding these compressed data to be used for the evidence of digital crimes.

There are two types of compression methods: Lossy compression to achieve high compression rate and Lossless compression to avoid data loss. The compression type which is commonly used in daily life such as .7z and .zip is the lossless compression. It is necessary to use the lossless compression because using lossy compression is not critical on images and videos but it is critical on text files. There are two methods of lossless compression techniques: Entropy-based compression and Dictionary-based compression. Entropy-based compression is the method of compressing overlapped sequences of characters by making them simple codes. The Shannon, Huffman, Golomb algorithms are the representative algorithms of entropy-based compression. On the other hand, dictionary-based compression is the method of compressing original data by expressing certain patterns of data in an index. Lempel-Ziv 77 (LZ77), Lempel-Ziv 78 (LZ78), Lempel-Ziv Storer-Szynanski (LZSS) are the representative algorithms of dictionary-based compression. There are various methods according to how to find the patterns and represent as indices. Although there are various compression types, there have been a few studies to discriminate compression types by analyzing the characteristics of the entire bitstreams [1][2]. Therefore, in this paper, we propose a new Convolutional Neural Network (CNN) which distinguishes the dictionary-based compression type for compressed data.

The previous discrimination methods used Support Vector Machine (SVM) by learning with the feature vector from the compressed data [3][4]. The feature vector which represents the compression type is very different as which feature extraction methods are used such as Frequency test and Runs test. Also, it is very limited to show various compression types using only hand-crafted features. In order to overcome these problems, we propose CNN based compression type discrimination network by learning with feature vectors which represent the whole bitstream of data.

Our proposed classification method targets text files rather than images or videos mainly for data compressed by dictionary encoding. Since we use the bitstream of the compressed text data as inputs, the features of the data are extracted using 1D-CNN network. In addition, we apply Spatial Pyramid Pooling (SPP) layer [5] to 1D-CNN network to obtain fixed-size feature vectors from inputs. In the case of images, cropping or zero-padding is usually used for adjusting the fixed-size. However, in the case of a text file, it is hard to change its arbitrary size to a fixed size because the data is sensitive to transform its original form. Nevertheless, if some editing techniques are applied to text data, it could be possible to change the whole information of the text data. Therefore, in this paper, we attempt to use arbitrary-sized input and extract the fixed-size feature vector by SPP layer. In addition, we apply softmax regression to the feature vectors from SPP layer to classify 16 dictionary-based compression types.

II. Main Algorithm

The overall network for dictionary-based compression type classification is shown in Fig. 1. The proposed network consists of two parts: Feature extraction part and Fixed length feature generation. In the feature extraction part, features for arbitrary-sized input data are extracted through several convolutional layers. In the fixed-length feature generation

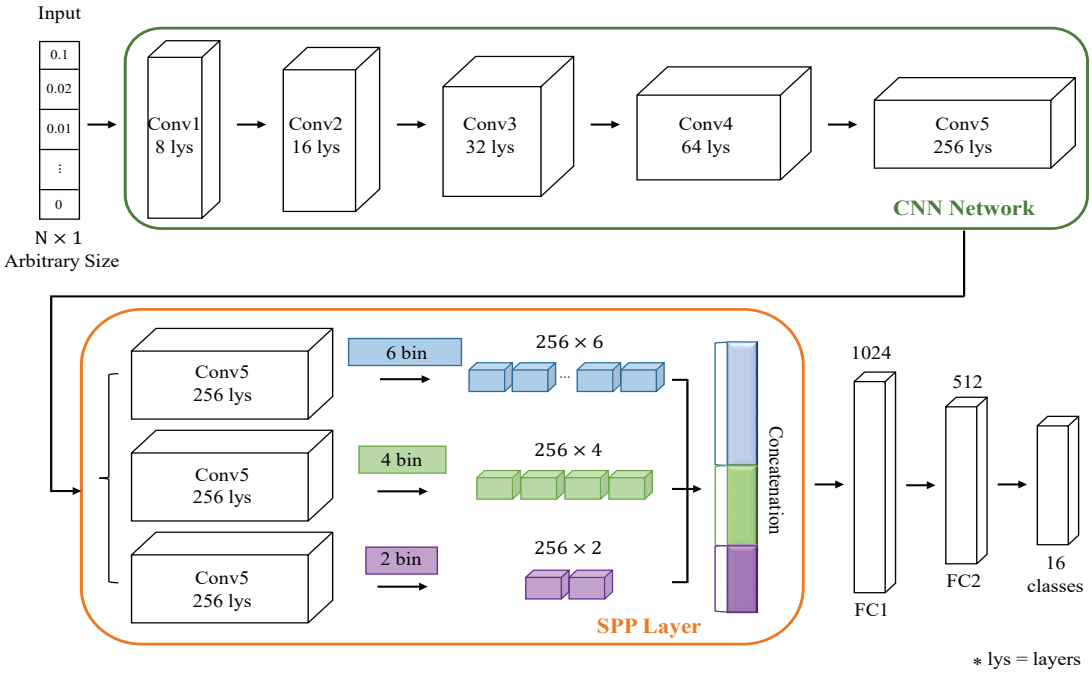

Fig. 1. Main architecture of our proposed network

part, the fixed-size feature vectors are generated by applying SPP layer. Finally, the classification of compression type is performed through fully-connected layers and softmax regression.

A. The type of Dictionary based compression

Dictionary-based compression is a method of reducing the length of data by storing certain patterns of characters by means of the index of the dictionary. Dictionary-based compression is based on the Lempel-Ziv algorithm and developed into several algorithms such as LZ77 and LZ78 algorithms. This compression type is divided into internal dictionary encoding and external dictionary encoding depending on whether information about the dictionary is included in the encoded data.

We explain the concept of dictionary-based compression by introducing the encoding process of the LZ77 algorithm which is one of the most popular dictionary-based compression algorithms. Fig. 2 represents the process of dictionary encoding. As shown in Fig. 2, LZ77 compresses the input data using a sliding window that consists of a search buffer and a lookahead buffer. To implement this algorithm, first, it is necessary to set the length of a search buffer and a lookahead buffer. In the figure, the length of a search buffer and a lookahead buffer are 5 and 3. Moreover, LZ77 finds the longest matched letters in the lookahead buffer and search buffer. These matched letters should be a sequence of consecutive bits from the start point of lookahead buffer. After then, the lookahead buffer and the search buffer are slid together to find the letters that match with bitstreams in

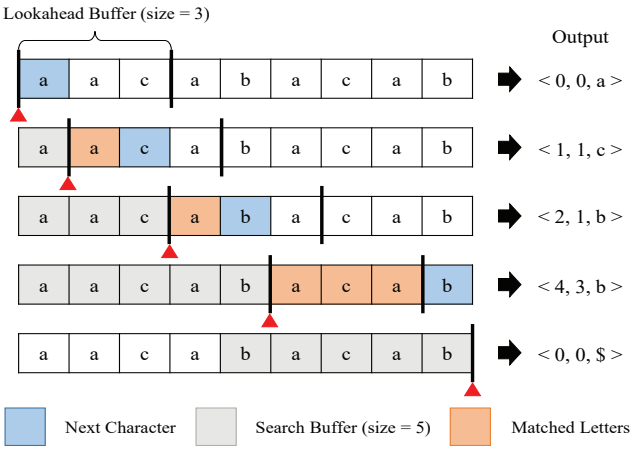

Fig. 2. Encoding process of the LZ77 when the input stream is "aacabacab" and the sizes of the search and lookahead buffers are 5 and 3, respectively.

the buffer. The matched letters are encoded as tuple $\langle i, j, X \rangle$, where i means the start point of the matched letters in the search buffer from the left, j represents the number of matched letters, and X is the next letter after the matched letters in the lookahead buffer. If there are no matched letters in the search and lookahead buffers, the output is $\langle 0, 0, X \rangle$. And if there are no further letters to be compressed, the output is $\langle i, j, \$ \rangle$, which \$ is the signal of the endpoint of input data. After the compression process is completed, LZ77 converts each tuple into binary form.

As described in the LZ77 algorithm, the dictionary-based compression method reduces the data by indexing the repeated characters in the data. We employ 16 dictionary-based

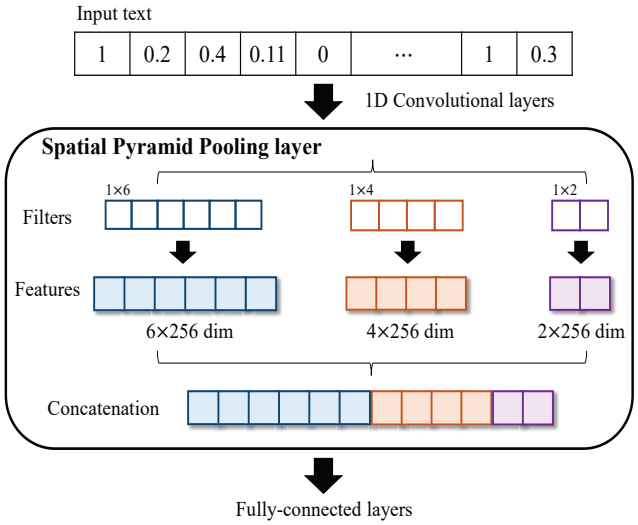

Fig. 3. Concept of Spatial Pyramid Pooling in the case of 1D text data

compression algorithms for compression type classification: Byte Pair Encoding (BPE), DEFLATE, Lempel-Ziv 4 (LZ4), Lempel-Ziv 77 (LZ77), Lempel-Ziv 78 (LZ78), Lempel-Ziv Jeff-Bonwick (LZJB), Lempel-Ziv MArkov chain (LZMA), Lempel-Ziv Oberhumer (LZO), Lempel-Ziv Ross Wiliam (LZRW), Lempel-Ziv Stac (LZS), Lempel-Ziv Storer-Szymanski (LZSS), Lempel-Ziv X (LZX), Run-Length Encoding (RLE), SNAPPY, and Zstandard. A brief description of other dictionary-based compression algorithms is included in the reference [6].

B. Spatial Pyramid Pooling

Spatial Pyramid Pooling is one of the most successful methods used for computer vision, and it is an extension of the Bag-of-Words model [7]. This is often used in the classification and object detection fields by further subdividing the image and extracting its characteristics. It makes the accuracy of recognition increase by enabling to generate representative values from images or windows of arbitrary size.

Fig. 3 shows the basic concept of SPP in the 1-D text data. The concept is simple enough to make fixed-length features to be connected to a fully-connected layer. This is the method to obtain the bin features for each channel of feature maps from convolutional layers and finally get a *(the total number of bins)* $\times$ *(the number of the channel of the last layer of CNN)* dimensional feature vector. By extracting features from the input text data for various bins, it is possible to not only generate fixed-length features but also obtain various scale characteristics, resulting in better classification performance.

In our proposed CNN network, SPP layer is applied after convolutional layers to get fixed length feature vectors for arbitrary-sized compressed data as shown in Fig. 3. There are other methods for making static feature vectors such as cropping the data or zero-padding. However, since the text data can be varied greatly in a small data change unlike other general images or videos, using SPP layer is more suitable for the text data.

C. CNN architecture for compression type classification

Compressed text data is employed as the input in the form of bitstreams represented in hexadecimal codes and each bit value is normalized within 0$\sim$1 range. The overall network consists of 5 convolutional layers, SPP layer, and 2 fully-connected layers. The kernel size is all same as 3 and the number of channels is different for various scale feature maps at each convolutional layer as shown in Fig. 1. Also, the number of the bin for SPP layer is set as 6, 4, 2 as shown in Fig. 3. As a result, CNN generates 256×6, 256×4, 256×2 dimensional feature vectors and concatenates these feature vectors into a fixed-length 256 dimensional feature vector. By creating features through pooling operation on various scales, the data of each algorithm has different characteristics. Finally, through softmax regression, the classification probability for 16 dictionary-based compression algorithms is obtained. The class which has the highest probability value becomes the estimated class of input.

III. Experiments

To validate the proposed network, we used text files from the publicly available database of WikiLeaks [8]. This database contains 1,440 text files. By using the open-source code of each algorithm, all text files were compressed individually with 16 dictionary-based compression algorithms. The number of training data is 18,400 text files (1,150 text files per algorithm) and the number of testing data is 4,640 text files (290 text files per algorithm).

The performance of our proposed network is shown by comparing the classification accuracy between the classification method using SVM and the proposed network. Unlike the feature vector obtained from deep learning in the proposed network, the SVM method uses the 6 extracted features per compressed text file as input. The tests [9] for feature extraction were conducted as below.

1) Frequency Test
   : The ratio of 0 and 1 in the entire bit streams

2) Frequency Test within a Block
   : The ratio of 0 and 1 in a block of M bits

3) Runs Test
   : The probability of appearing 0 or 1 continuously in the entire bit streams

4) Tests for the Longest-Run-of-Ones in a block
   : The probability of the longest appearance of 0 or 1 continuously in the entire bit streams

TABLE I
THE CLASSIFICATION ACCURACY OF THE SVM METHOD AND THE PROPOSED NETWORK

No.	Type	SVM		Proposed	
		Top 1	Top 3	Top 1	Top 3
1	LZ77	71.25%	92.29%	98.33%	100%
2	LZ78	64.79%	91.04%	99.67%	100%
3	LZW	48.13%	95.00%	96.00%	99.00%
4	LZSS	63.54%	95.00%	99.00%	100%
5	BPE	78.13%	98.96%	100%	100%
6	LZ4	65.00%	94.58%	99.67%	100%
7	LZJB	94.38%	96.67%	97.00%	99.67%
8	LZRW	45.42%	88.13%	100%	100%
9	RLE	51.25%	88.13%	99.67%	100%
10	LZS	49.38%	95.00%	100%	100%
11	DEFLATE	57.92%	97.92%	99.33%	100%
12	LZMA	75.42%	98.54%	99.00%	100%
13	LZO	55.21%	98.75%	98.00%	100%
14	LZX	52.08%	99.58%	98.33%	100%
15	SNAPPY	61.67%	93.54%	100%	100%
16	Zstandard	87.92%	97.29%	92.67%	100%
Average		**63.84%**	**95.04%**	**98.54%**	**99.92%**

5) Binary Matrix Rank Test
: The rank of sub-matrices of the entire bit streams

6) Discrete Fourier Transform
: The Discrete Fourier Transform of the entire bit streams

Each test's output was a single value which represents the whole characteristic of a compressed text file. The 1×6 feature vector per text file which is a concatenation of these test values was used for learning SVM.

Table I shows the classification accuracy of the method using SVM and the proposed network. The table has Top 1 and Top 3 accuracies of the 16 dictionary-based compression algorithms. As shown in Tabel I, our proposed network outperforms in both Top 1 and Top 3 accuracies. Especially, the proposed network at Top 1 has the average accuracy of 98.54%, while the method using SVM is only 63.84%. For the SVM method, the classification accuracy depends on which test is used to construct the feature vector. Fig. 4 shows Top 1 and Top 3 accuracies of 6 sets which are composed of 5 tests of the test list to figure out the dependency on the test list for making feature vector. Set 4 is composed of $\{1, 2, 4, 5, 6\}$ tests in the list, and Set 6 is composed of $\{2, 3, 4, 5, 6\}$ tests in the list. Two sets have significantly low accuracy than the other sets. It means that it is important to use what kinds of feature extraction methods for better performance. Although there are several appropriate feature extraction methods, the method which uses the feature vector of concatenating hand-crafted features is limited to contain enough information to distinguish several compression types. Therefore, the proposed network which generates feature vectors using deep learning shows better performance in the compression type classification.

IV. CONCLUSION AND FUTURE WORKS

We proposed a network for dictionary-based compression type classification. It consisted of 1D convolutional layers and Spatial Pyramid Pooling layer. As SPP layer was used,

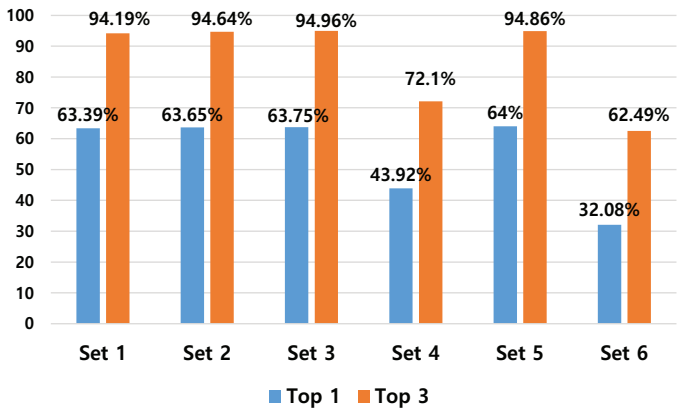

Fig. 4. Average classification accuracy of SVM depending on the different sets of tests

the arbitrary-sized text files could be treated using CNN by making a fixed length of the feature vector. Also, using deep learning for making the feature vector, it overcame the limitations of the method using hand-crafted features by fully understanding the characteristics of compressed data. Since these reasons, the proposed network showed much better performance than the basic classification method using SVM. Even though our method has good results on classification, it is limited to generalize the algorithm. It is necessary to handle various text files to obtain accurate results for future work.

ACKNOWLEDGMENT

This work was supported by the research fund of Signal Intelligence Research Center supervised by Defense Acquisition Program Administration and Agency for Defense Development of Korea.

REFERENCES

[1] B. Kwon, M. Gong and S. Lee, "Novel error detection algorithm for LZSS compressed data," *IEEE Access,* vol. 5, 2017, pp. 8940-8947.
[2] B. Kwon, S. Lee, "Error detection algorithm for Lempel-Ziv-77 compressed data," *Journal of Communications and Networks,* vol. 21, no. 2, 2019, pp. 100-112.
[3] J.A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural processing letters,* vol. 9, no. 3, 1999, pp. 293-300.
[4] B. Kwon, M. Gong, J. Huh and S. Lee, "Identification and Restoration of LZ77 Compressed Data Using a Machine Learning Approach," *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC),* 2018, pp. 1787-1790.
[5] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE transs on pattern analysis and machine intelligence,* vol. 37, no. 9, 2015, pp. 1904-1916.
[6] Y. Rathore, M.k. Ahirwar, and R. Pandey, "A brief of data compression algorithms," *International Journal of Computer Science and Information Security,* vol.11, no. 10, 2013, pp. 86-94.
[7] G. Csurka, C. Dance, L. Fan, J. Willamowski and C. Bray, "Visual categorization with bags of keypoints," *Workshop on statistical learning in computer vision, ECCV,* vol. 1, no. 1-22, 2004, pp. 1-2.
[8] https://911.wikileaks.org/files/
[9] A. Rukhin, J. Soto, J. Nechvatal and M. Smid, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," Booz-Allen and Hamilton Inc Mclean Va, 2001.