

# 바이트 평균의 Gray-Scale화를 통한 Signature가 존재하지 않는 멀티미디어 데이터 조각 파일 타입 분류 연구\*

윤 현 호,<sup>†</sup> 김 재 현, 조 현 수, 원 종 은, 김 견 우, 조 재 현<sup>‡</sup>  
한국정보기술연구원 베스트 오브 더 베스트(교육생)

## Classification of Non-Signature Multimedia Data Fragment File Types With Byte Averaging Gray-Scale\*

Hyun-ho Yoon,<sup>†</sup> Jae-heon Kim, Hyun-soo Cho,  
Jong-eun Won, Gyeon-woo Kim, Jae-hyeon Cho<sup>‡</sup>  
Best of the Best, Korea Information Technology Research Institute(Trainee)

### 요 약

일반적으로 시그니처와 파일 메타정보가 없는 파편화된 파일은 복구가 어렵다. 특히 멀티미디어 파일은 파편화 가능성이 크고 높은 엔트로피를 가지고 있으므로 현재 시그니처 기반의 카빙으로는 복구하는 것이 거의 불가능하다. 이러한 문제를 해결하기 위해 파편화된 파일에 대한 연구가 진행되고 있지만 멀티미디어 파일에 대한 연구는 부족한 실정이다. 본 논문은 시그니처(Signature)와 파일 메타정보가 없는 파편화된 멀티미디어 파일의 타입을 분류하는 연구이다. 파일 타입에 따라 특정 바이트 값의 빈도 차이를 통해 각 파일 타입의 특징값을 추출하며, 그에 맞는 Gray-Scale 테이블을 설계하고 CNN(Convolutional Neural Networks) 모델을 이용하여 JPG, PNG, H.264, WAV 총 4 가지 멀티미디어의 파일 타입을 분류하는 방법을 제시한다. 본 논문을 통해 시그니처와 파일 메타정보가 없는 파편화된 파일 타입의 분류 연구를 촉진하여 다양한 파일의 복구 가능성을 높일 수 있을 것으로 기대된다.

### ABSTRACT

In general, fragmented files without signatures and file meta-information are difficult to recover. Multimedia files, in particular, are highly fragmented and have high entropy, making it almost impossible to recover with signature-based carving at present. To solve this problem, research on fragmented files is underway, but research on multimedia files is lacking. This paper is a study that classifies the types of fragmented multimedia files without signature and file meta-information. Extracts the characteristic values of each file type through the frequency differences of specific byte values according to the file type, and presents a method of designing the corresponding Gray-Scale table and classifying the file types of a total of four multimedia types, JPG, PNG, H.264 and WAV, using the CNN (Convolutional Natural Networks) model. It is expected that this paper will promote the study of classification of fragmented file types without signature and file meta-information, thereby increasing the possibility of recovery of various files.

**Keywords:** CNN, Digital Forensics, Data Fragment Classification

Received(02. 03. 2020), Modified(04. 09. 2020),  
Accepted(04. 09. 2020)

\* 본 논문은 문화체육관광부와 한국저작권보호원이 주관한 2019년 디지털포렌식 아이디어/논문 공모전에서 수상작으로 선정된 논문(을 수정 보완한 논문)입니다.

This work has selected for award at the 2019 Digital Forensics Idea / Thesis Contest administered by Ministry of Culture, Sports

<sup>†</sup> 주저자, hho7735@gmail.com

<sup>‡</sup> 교신저자, jehyun9027@gmail.com(Corresponding author)

## I. 서론

본 논문의 순서로 II. 관련 연구, III. 기존 연구, IV. 파일 조각의 Gray-Scale 테이블 분석, V. 멀티미디어 데이터 파일 조각 파일 타입 분류, VI. 결론에 대해 기술하며 본 논문을 마친다.

파일 크기가 큰 멀티미디어 파일의 특성상 파일이 파편화되어 저장될 가능성이 크며, Signature와 파일 메타정보가 없는 파편화된 파일의 경우 복구가 힘든 실정이다. 이러한 문제를 해결하기 위해서는 파편화된 파일의 파일 타입을 우선으로 식별하여 분류해야 하며, 본 논문에서는 JPG, PNG, H.264, WAV 총 4가지 대표적인 멀티미디어 파일의 파일 타입을 분류하는 방법을 제시한다.

## II. 관련 연구

본 장에서는 파일 복구 흐름도를 통해 파일 복구 방법과 한계를 소개하고 본 논문을 통해 개선할 수 있는 점에 대해서 기술한다.

파일 복구에 대한 전체적인 흐름은 Fig.1.과 같다. 삭제된 파일을 복구하고자 할 때, 가장 먼저 파일의 다양한 속성값을 가지고 있는 파일 메타정보를 확인한다. 파일 메타정보가 남아있다면, 비 할당 영역에서 삭제된 파일의 데이터들을 찾아낼 수 있어 파일을 쉽게 복구할 수 있다. 삭제된 파일에 대한 파일 메타정보가 덮어써진 경우에는 크게 두 가지 상황으로 나누어진다. 첫 번째는 파편화되지 않은 채로 저장된 파일이 삭제된 후 파일 메타정보가 덮어 쓰인 경우이다.

다. 이 경우에는 김진국 등[1]이 연구한 파일 카빙 기법인 ‘RAM-Slack Carving’, ‘File-Structure Carving’ 기법을 사용하여 삭제된 파일을 복구해낼 수 있다. 두 번째는 파편화된 채로 저장된 파일이 삭제된 후 파일 메타정보가 덮어써진 경우이다. 이 경우 파편화된 파일을 모두 수집하여 온전한 파일로 만들어야 파일 복구가 가능하다. 하지만 저장 매체에 Signature가 존재하지 않는 수많은 파일 조각들이 존재하므로 삭제된 파일 조각을 식별하기 힘들어 복구되지 않는 경우가 많다. 따라서 본 논문의 연구를 통해 Fig.1.의 Identifying the file types of non-signature file fragments 단계를 개선한다면 파일 조각들의 파일 타입을 분류하고 삭제된 파일을 복구해낼 가능성이 향상될 수 있다.

## III. 기존 연구

본 장에서는 파일 타입 분류를 위한 기존 연구에 대해 소개하고 본 논문과의 차이점에 대해서 기술한다.

### 3.1 바이트 기반 파일 타입 분류 연구

M.McDaniel, et al. 는 BFA(Byte Frequency Analysis), BFC(Byte Frequency Cross-correlation), FHT(File Header / Trailer) 바이트 기반 분석 방법을 제안하였다[2]. BFA는 파일 특징마다 바이트 개수가 비슷한 특성을 이용하여 파일 타입의 지문을 주파수 형태로 생성한다. 비교하려는 파일의 상관계수를 측정된 뒤, 결과값이 가장 비슷한 값으로 파일 특징을 결정한다. BFC는 위와 비슷하게 주파수 형태로 변환하지만, 특정 파일 타입에 나타나는 바이트 값을 중점으로 비교한다. 예를 들어, HTML에 ‘<.’ 문자가 자주 등장하므로 해당 문자를 나타내는 바이트 값의 상관계수를 측정하는 방법이다. FHT는 헤더(Header)/푸터(Footer)에 존재하는 Signature를 탐색하여 파일 타입을 식별한다. 해당 논문은 JPG, WAV를 포함한 30개의 파일 타입 분류를 시도하여, 각 분석 방법마다 평균 27.50%, 45.83%, 95.83%의 정확도가 나왔다.

하지만 해당 연구에서는 파일을 파편화시키지 않고 분석하였으며, 파일 타입의 Signature를 형성하기 어려운 점과 상관계수의 계산 복잡성 때문에 BFA, BFC 분석 방법은 매우 두드러진 특징을 가진 파일

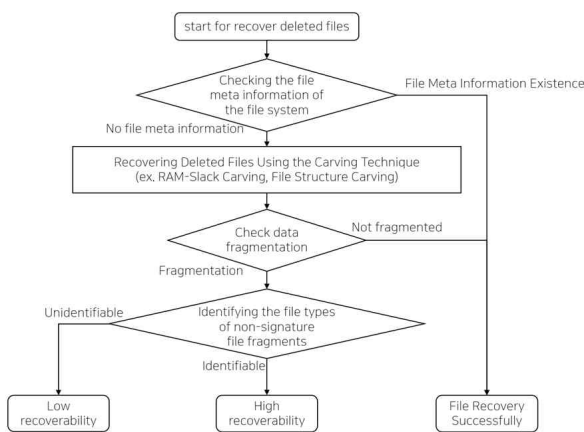


Fig. 1. File Recovery Flow

타입만 분류할 수 있다. 또한, FHT 분석 방법은 Header/Footer에 존재하는 Signature를 분석하기 때문에, Header/Footer가 없는 조각 데이터의 경우 분류할 수 없게 된다.

### 3.2 AI를 이용한 파일 타입 분류 연구

Qiming Li, et al. 의 BFA의 상관계수의 복잡성을 SVM(Support Vector Machine)으로 해결한 연구도 존재한다[3]. Header/Footer를 제거하고 4096 bytes로 조각 데이터를 생성한 뒤, 비교 대상을 JPEG로 기준을 잡아 나머지 파일 타입을 1 대 1로 비교하였다. 각 파일 타입마다 학습 셋 720개, 시험 셋 80개 조각으로 연구를 진행하였다.

하지만 데이터 구조체에 존재하는 Signature를 완벽히 제거하지 않았고, 학습과 시험을 위한 조각 개수가 적었다. 또한, JPG와 각각 파일 타입에 1 대 1 비교를 하였기 때문에 실무에서 사용하긴 어려우며 해당 연구의 정확도는 Fig.2.과 같다.

	DLL	EXE	PDF	MP3
Linear	<b>98.25%</b>	<b>86.88%</b>	81.13%	<b>89.13%</b>
Polynomial	97.63%	47.38%	49.44%	61.38%
RBF	96.94%	48.06%	<b>91.56%</b>	44.25%
Sigmoid	96.44%	46.50%	87.69%	69.19%

Fig. 2. Qiming Li, et al. Classification Results(3)

### 3.3 Gray-Scale을 이용한 파일 타입 분류 연구

Qian Chen, et al. 의 파일 데이터를 Gray-Scale 이미지로 변환하여 CNN(Convolutional Neural Networks)으로 분류한 연구도 존재한다[4]. 1

6개 파일 타입을 선정하여 Header/Footer를 제거하고 4096 bytes로 조각 데이터를 생성한 뒤, Gray-Scale로 변환하였다. 학습 셋에 약 81만 개, 시험 셋에 약 31만 개 조각으로 나눠 연구를 진행하였다. 하지만 위 연구는 데이터 구조체에 존재하는 Signature를 완벽히 제거하지 않았고, 파일 타입별로 정확도의 편차가 크다. 특히 멀티미디어 파일인 JPG, PNG의 정확도는 각 15%, 21%로 낮은 정확도를 보였으며, 전체적인 결과는 Fig.3.과 같다.

따라서 본 연구는 위의 연구들을 보완하여 데이터 구조체의 Signature를 완벽히 제거하여 오직 파일의 데이터만으로 파일 타입 분류를 시도했으며, BFA의 한계점을 보완하여 파일 타입 분류 방법을 제시하고 분류 정확도를 높여 멀티미디어 파일의 복구 가능성을 높이고자 한다.

## IV. 파일 조각의 Gray-Scale 테이블 분석

본 장에서는 파일 타입 선정부터 Gray-Scale 테이블 설계까지 본 연구에서 진행된 연구 절차에 대해서 상세히 설명한다.

### 4.1 파일 타입 선정

파일 크기가 큰 멀티미디어 파일의 특성상 파일이 파편화되어 저장될 가능성이 크며, Signature와 파일 메타정보가 없는 파편화된 파일의 경우 복구가 쉽지 않다. 따라서 본 연구에서 식별해내고자 하는 파일 타입은 멀티미디어의 대표 분야인 이미지, 비디오, 오디오에서 JPG, PNG, H.264(MP4), WAV로 총 4개를 선정하였다.

	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%								7%
LOG									97%							
PDF										38%						
PNG					16%						21%					
PPT					16%											
RTF		15%				13%										
TEXT					18%									67%		3%
XLS																92%
XML		8%														92%

Fig. 3. Qian Chen, et al. Classification Results(4)

그중 비디오 파일의 특징은 코덱에서 나타나기 때문에 파일 조각으로 비디오 파일 타입을 식별해낼 수 없다. 그러므로 비디오의 경우 파일 타입이 아닌 코덱으로 선정하였다.

### 4.2 파일 조각 크기 선정

Signature와 파일 메타정보가 없다면 데이터 조각의 경계들을 구분할 수 없다. 그러므로 조각의 최소단위로 파일 타입을 식별해낼 수 있다면, 저장 매체에 존재하는 모든 파일 조각들의 파일 타입을 식별해낼 수 있다. 따라서 저장 매체에 존재할 수 있는 파일 조각의 최소 크기는 시스템의 데이터 입출력 단위이고 파일시스템, 볼륨 크기, 운영체제에 의존적이므로 본 연구에서는 현재 가장 많이 사용되는 'NTFS(New Technology File System), 256GB ~ 2TB 볼륨 크기, Windows 10 운영체제' 환경의 입출력 단위인 4096 bytes로 파일 조각들을 생성해 데이터 셋으로 사용한다.

### 4.3 파일 조각 생성 방법

파일 포맷에는 Header/Footer뿐만 아니라 구조체 내부에도 특정 Signature가 존재한다. 파일 포맷들은 서로 다른 구조체를 가지고 있어, 구조체의 Signature를 통해서도 파일 조각의 파일 타입을 식별해낼 수 있다.

따라서 파일의 큰 부분을 차지하는 데이터 영역에서 파편화가 주로 발생하므로 Fig.4.와 같이 PNG의 'IDAT', Fig.5.와 같이 JPG의 'scanData', Fig. 6.과 같이 MP4의 'mdat', Fig.7.과 같이 WAV의 'DATACHUNK'를 추출하여, 구조체의 Signature들을 모두 제거하였다. 그리고 구조체에는 Signature뿐만 아니라 구조체의 크기, 압축 Signature 등 파일 조각의 파일 타입을 식별할 때 도움을 줄 수 있는 다양한 파일 메타정보들도 포함되어 있어 모두 제거하였다. 위와 같은 방식으로 정제된 데이터를 409

Name	Value
> struct PNG_SIGNATURE sig	
> struct PNG_CHUNK chunk[0]	IHDR (Critical, Public, Unsafe to Copy)
> struct PNG_CHUNK chunk[1]	sRGB (Ancillary, Public, Unsafe to Copy)
> struct PNG_CHUNK chunk[2]	gAMA (Ancillary, Public, Unsafe to Copy)
> struct PNG_CHUNK chunk[3]	pHYs (Ancillary, Public, Safe to Copy)
> struct PNG_CHUNK chunk[4]	IDAT (Critical, Public, Unsafe to Copy)
> struct PNG_CHUNK chunk[5]	IDAT (Critical, Public, Unsafe to Copy)
> struct PNG_CHUNK chunk[6]	IEND (Critical, Public, Unsafe to Copy)

Fig. 4. PNG File Structure

Name	Value
▼ struct JPGFILE jpgfile	
enum M_ID SOIMarker	M_SOI (FFD8h)
> struct APP0 app0	
> struct APP2 app2	
> struct DQT dqt[0]	
> struct DQT dqt[1]	
> struct SOF0 sof0	
> struct DHT dht[0]	
> struct DHT dht[1]	
> struct DHT dht[2]	
> struct DHT dht[3]	
> struct SOS scanStart	
> char scanData[85634]	
enum M_ID EOIMarker	M_EOI (FFD9h)

Fig. 5. JPG File Structure

Name	Value
> Box[0]	ftyp
> Box[1]	moov
> Box[2]	mdat

Fig. 6. MP4 File Structure

Name	Value
> struct WAVRIFFHEADER header	
> struct FORMAT_CHUNK format	
> struct DATACHUNK data	

Fig. 7. WAV File Structure

6 bytes로 나누어 데이터 셋으로 사용하는 파일 조각들을 생성하였다.

### 4.4 특징 추출

모든 Signature가 제거되고 4096 bytes로 나누어진 데이터들을 파일 타입별로 각 10만 개씩 바이트 빈도 평균을 내었다. 그 후 평균을 바탕으로 파일 타입의 특징값을 추출하였다. 0x00 ~ 0xFF까지 모든 HEX(HEX) 값에 대한 빈도수를 확인하였으며 4096 bytes 기준으로 각 파일의 HEX 값은 평균적으로 15.9 ~ 16.1의 빈도수를 가진다.

따라서 본 연구는 HEX 값의 빈도수가 15.9보다 작거나 16.1보다 큰 HEX 값들에 집중하여 특징을 추출하였으며 각 파일 타입의 대표적인 5개 특징값과 빈도수는 Table 1.과 같으며 이외에도 파일 타입의 특징을 나타내는 공통적인 특징값을 발견하여 총 64개의 특징값을 추출하였다.

### 4.5 Gray-Scale 테이블 설계

본 논문에서는 불필요한 학습 데이터를 줄이고 특징값을 더 쉽게 학습하는 모델을 만들기 위해 특징값을 Gray-Scale 이미지화하였다. 또한, 이미지 분류의 대표적인 알고리즘인 CNN을 사용하였다.

Table 1. Features and Frequency of each file type

File Type	Feature 1 (Frequency)	Feature 2 (Frequency)	Feature 3 (Frequency)	Feature 4 (Frequency)	Feature 5 (Frequency)
JPG	0x00(33.26)	0x20(24.72)	0x92(21.94)	0x24(21.62)	0x4A(20.52)
PNG	0x00(31.08)	0x10(25.30)	0x08(24.49)	0x18(22.69)	0xEF(21.00)
H.264	0x00(19.53)	0xFF(16.89)	0x01(16.48)	0xFE(16.37)	0xFD(16.23)
WAV	0x00(177.16)	0xFF(143.09)	0xFE(93.43)	0x01(90.23)	0x02(77.85)

02	03	FF	EF	F7	00	10	49
92	20	01	A4	55	95	6D	48
D4	A9	E8	A5	AA	F0	2F	A7
52	4A	54	A8	8E	29	17	2A
DB	F3	D3	15	D0	13	EB	B5
B0	1D	BD	08	18	04	0C	2D
AD	0D	E7	D7	D8	26	42	21
24	84	FC	BF	7F	FD	FE	FB

Fig. 8. Gray-Scale Feature Table

바이트 빈도 평균을 이용하여 추출된 총 64개의 HEX 값을 Gray-Scale과 CNN 모델 적용에 최적화하여 설계하였으며 고안된 테이블은 Fig.8.과 같다.

Gray-Scale 테이블은 11개의 공통으로 가지는 특징값을 시작으로 22개의 JPG 특징값, 20개의 PNG 특징값, 11개의 H.264 특징값을 배치하였다. Fig.8.을 이용하여 바이트 빈도를 담은 파일을 생성하였고, 0 ~ 255까지 표현이 가능한 Gray-Scale 방식을 고려하여 255 이상의 빈도수를 가질 때 최댓값인 255에 수렴하도록 설계하였다. 또한, 뚜렷한 Edge-Detection을 위해 각각의 특징의 바이트 빈도를 64번씩 반복하여 파일을 생성하였다.

Table 2. Gray-Scale image









JPG		H.264	
PNG		WAV	

Table 3. Gray-Scale image (35% increase in brightness)

JPG		H.264	
PNG		WAV	

최종적으로 4096 bytes의 파일이 생성되며 Gray-Scale 변환 시 파일 타입에 따라 가로줄의 색상이 차이 나는 것을 Table 2.와 Table 3.을 통해 확인할 수 있다.

### V. 멀티미디어 데이터 파일 조각 파일 타입 분류

본 장에서는 시험 환경부터 분류 결과까지 본 연구에서 진행된 연구결과에 대해서 상세히 설명한다.

Table 4. Results of each file type classification

	Accuracy	Precision	Detail False Positive
JPG	86.31%	95.97%	H.264(6.85%), PNG(6.0%), WAV(0.84%)
PNG	87.04%	91.33%	H.264(12%), JPG(0.96%), WAV(0.0%)
H.264	97.94%	83.36%	JPG(1.05%), PNG(1.01%), WAV(0.0%)
WAV	96.45%	99.17%	JPG(1.61%), PNG(1.25%), H.264(0.69%)
평균	91.93%	92.45%	

### 5.1 시험 환경

딥러닝 모델은 이미지 분류 부문에서 뛰어난 성능을 보이는 CNN 모델로 선정하였으며, 시험 환경은 아마존 AWS의 p3.2xlarge를 사용하였으며 세부 사양은 vCPU(8), ECU(26), 메모리(61GiB)로 진행하였다

### 5.2 학습 방법

학습 데이터와 시험 데이터를 서로 독립된 파일에서 수집하였고 데이터의 모든 Signature를 제거한 후 4096 bytes로 분할하였다. 또한, 파일의 선형성이 시험에 반영되지 않도록 모든 파일을 완벽하게 섞었다. 분할된 데이터를 바이트 빈도 특징값을 포함한 파일로 변환한 후 64x64 크기로 Gray-Scale 이미지를 실시하여 최종적으로 학습 데이터 12만 개, 시험 데이터 4만 개를 생성하였다.

학습 데이터 12만 개의 이미지를 폴더로 라벨링을 하고 CNN 학습을 하였으며 3번의 Convolution 레이어와 Maxpooling 레이어를 거치고 Flatten 함수를 통해 리스트를 1차원으로 만든다. 상기 과정에서 Dropout 함수와 데이터 셋을 학습(7), 시험(3)의 비율로 나눠 Overfitting을 방지했으며 이후 Softmax 활성화 함수를 통해 출력값을 정규화하여 출력한다. Optimizer를 Adam으로 채택하였고 Epoch는 30, Batch-size는 256으로 학습을 진행하였다. 해당 과정을 통해 만들어진 모델에 총 4만 개의 데이터로 시험을 진행하였다.

### 5.3 시험 결과

서로 독립적인 학습 데이터 12만 개와 시험 데이터 4만 개를 이용하여 파일 분류를 진행하였다. 본

논문에서 제시한 파일 타입 분류 방법론을 통해 Accuracy 평균 91.93%, Precision 평균 92.45%의 결과를 확인할 수 있었다.

파일 타입 분류 결과는 Table 4.와 같으며 Recall의 경우 91.93%로 Accuracy와 같아 Table 4.에서 생략하였다.

## VI. 결 론

본 연구는 기존의 연구들과는 다르게 파일에 존재하는 모든 Signature를 제거하였으며, 순수 데이터를 바탕으로 바이트 빈도 통계를 내어 특징적인 HEX 값을 추출하였다. 또한, 최적의 Gray-Scale 테이블을 설계하여 변환된 이미지를 CNN 모델을 통해 평균 91.93%의 정확도로 분류할 수 있었다.

기존 연구에서 본 논문과 연구의 대상이 되는 파일 타입이 온전히 같지 않으므로 직접적인 비교는 무의미하다. 하지만 본 논문과 가장 유사한 Qian Chen, et al. 이 진행한 연구[4]에 비해 JPG는 86.31%로 약 71%, PNG는 87.04%로 약 66% 개선되는 모습을 보였다.

본 연구진들은 해당 연구보다 더 많은 파일 타입에 적용하기 위해 다양한 시도를 하고 있으며, 본 연구에서 제안한 파일 타입 분류 방법을 적용한 솔루션을 개발 중이다. 위와 같은 솔루션이 완성된다면 멀티미디어의 파일 타입을 효과적으로 분류할 것이며 향후 파일 복구 연구에 많은 도움이 될 것으로 기대한다. 본 연구를 통해 이러한 연구가 활발히 진행되어 더 많은 확장자를 분류할 수 있다면 훗날 디지털 포렌식 분야에서 Signature, 파일 메타정보가 없는 파편화된 파일에 대한 복구가 손쉽게 이루어져 포렌식 수사에 많은 기여를 할 것으로 예상된다.

## References

- [1] Jin-kook Kim, Seung-bong Lee, Jung-heum Park, Je-wan Bang, Sang-jin Lee, "A Study of File Carving Techniques," *Journal of Digital Forensics*, (2), pp. 19-31, Jun, 2008.
- [2] Mason McDaniel and M.Hossain Heydari, "Content Based File Type Detection Algorithms," *Hawaii International Conference on System Sciences*, pp. 10-19, Jan, 2003.
- [3] Qiming Li, Alvin Y.Ong, Ponnuthurai N.Suganthan and Vrizlynn L.L.Thing, "A novel support vector machine approach to high entropy data fragment classification," *Proceedings of the South African information security multi-conference*, pp. 236-247, Jan, 2010.
- [4] Qian Chen, Qing Liao, Zoe L.Jiang, Junbin Fang, Siuming Yiu, Guikai Xi, Rong Li, Zhengzhong Yi, Xuan Wang, Lucas C.K.Hui, Dong Liu and En Zhang, "File Fragment Classification Using Grayscale Image Conversion and Deep Learning in Digital Forensics," *IEEE Symposium on Security and Privacy Workshops*, pp. 140-147, May, 2018.

〈저자소개〉



윤 현 호 (Hyun-ho Yoon) 학생회원  
2020년 2월: 수원대학교 정보보호학과 졸업  
〈관심분야〉 정보보호, 악성코드 분석, 리버싱



김 재 현 (Jae-heon Kim) 학생회원  
2014년 3월~현재: 숭실대학교 컴퓨터학부 학사과정  
〈관심분야〉 정보보호, 디지털포렌식



조 현 수 (Hyun-soo Cho) 정회원  
2019년 8월: 전주대학교 정보통신공학과 졸업  
〈관심분야〉 정보보호, 취약점 분석



원 중 은 (Jong-Eun Won) 정회원  
2017년 2월: 영진전문대 컴퓨터정보계열 졸업  
〈관심분야〉 정보보호, 통신공학



김 건 우 (Gyeon-woo Kim) 학생회원  
2017년 3월~현재: 연세대학교 미래캠퍼스 컴퓨터공학과 학사과정  
〈관심분야〉 정보보호, 소프트웨어공학



조 재 현 (Jae-hyeon Cho) 학생회원  
2016년 3월~현재: 중부대학교 정보보호학과 학사과정  
〈관심분야〉 정보보호, 소프트웨어공학