CNN과 GRU를 활용한 파일 유형 식별 및 분류

[1]성민규, [2*]손태식

File Type Identification Using CNN and GRU

[1]Mingyu Seong, [2*]Taeshik Shon

요 약

현대 사회에서의 디지털 데이터의 빠른 증가로 디지털 포렌식이 핵심적인 역할을 하고 있으며, 파일 유형 식별은 그 중에서 중요한 부분 중 하나이다. 파일 유형을 빠르고 정확하게 식별하기 위해서 인공지능을 사용한 파일 유형 식별 모델 개발 연구가 진행되고 있다. 그러나 기존 연구들은 일부 국내 점유율이 높은 파일을 식별할 수 없어, 국내에서 사용하기에 부족함이 있다. 따라서 본 논문에서는 CNN과 GRU를 활용한 더욱 정확하고 강력한 파일 유형 식별 모델을 제안한다. 기존 방법의 한계를 극복하기 위해 제안한 모델은 FFT-75 데이터셋에서 가장 우수한 성능을 보이며, 국내에서 높은 점유율을 가지는 HWP, ALZ, EGG와 같은 파일 유형도 효과적으로 식별할 수 있다. 제안한 모델과 세 개의 기존 연구 모델(CNN-CO, FiFTy, CNN-LSTM)을 서로 비교하여 모델 성능을 검증하였다. 최종적으로 CNN과 GRU 기반의 파일 유형 식별 및 분류 모델은 512바이트 파일 조각에서 68.2%의 정확도를, 4096바이트 파일 조각에서는 81.4%의 정확도를 달성하였다.

Abstract

With the rapid increase in digital data in modern society, digital forensics plays a crucial role, and file type identification is one of its integral components. Research on the development of identification models utilizing artificial intelligence is underway to identify file types swiftly and accurately. However, existing studies do not support the identification of file types with high domestic usage rates, making them unsuitable for use within the country. Therefore, this paper proposes a more accurate file type identification model using Convolutional Neural Networks (CNN) and Gated Recurrent Units (GRU). To overcome limitations of existing methods, the proposed model demonstrates superior performance on the FFT-75 dataset, effectively identifying file types with high domestic usage rates such as HWP, ALZ, and EGG. The model's performance is validated by comparing it with three existing research models (CNN-CO, FiFTy, CNN-LSTM). Ultimately, the CNN and GRU based file type identification and classification model achieved 68.2% accuracy on 512-byte file fragments and 81.4% accuracy on 4096-byte file fragments.

Keywords: *Digital Forensics, File fragment type identification, File carving, CNN, GRU*

[1] *아주대학교 사이버보안 전공 학사 (mandu9280@ajou.ac.kr)*
[2*] *교신저자 아주대학교 사이버보안학과 교수 (tsshon@ajou.ac.kr)*

Received: Jan. 10, 2024, Revised: Feb. 22, 2024, Accepted: Mar. 02, 2024

I. 서론

현대 사회에서는 디지털 기술의 발전으로 인해 다양한 형태의 디지털 데이터가 급속히 증가하고 있다. 이러한 데이터의 증가로 파일 유형 분류 기술이 중요해지고 있다. 파일 유형 식별 기술은 운영체제의 기능, 방화벽, 침입 탐지 시스템, 안티바이러스, 파일 필터링, 스테그분석, 디지털 포렌식 등에서 널리 사용된다[1][2].

디지털 포렌식은 범죄 수사 및 사법 과정에서 핵심적인 역할을 하고 있다. 특히, 디지털 증거물의 신속하고 정확한 분석은 디지털 범죄 사건의 해결과 피해자의 보호에 중대한 영향을 미치고 있다. 그러나 디지털 포렌식 분야에서 가장 기본적인 단계인 파일 유형 식별은 여전히 주요한 과제 중 하나다. 다양한 형식과 확장자를 가진 파일들을 신속하게 식별하는 것은 디지털 포렌식 전문가들에게 요구되는 핵심 기술이다.

메모리 포렌식과 단편화된 데이터 분석에서는 파일 유형을 식별하는 데에 메모리의 작은 데이터 조각에 의존한다. 이러한 조각들은 메모리 전체에 흩어져 있어, 데이터의 연결과 복구를 어렵게 만든다. 그러나 전체 파일을 복구할 수 없더라도 개별 조각에 대한 파일 유형 식별은 수사에 유용한 정보를 제공할 수 있다[3]. 파일의 유형을 결정하기 위해서 파일시스템의 메타데이터에 확장자를 기록하거나, 파일의 헤더 부분에 타입 구분자를 기록하는 방식이 주로 사용된다[1][2]. 대부분의 기존 기술은 이러한 데이터를 검사하는 방법을 기반으로 한다. 매직 넘버 기반 방법은 파일 시그니처라고 하는 고유한 바이트열을 찾아 그 값에 해당하는 파일 유형을 식별한다. 그러나 *.txt 와 같이 파일 시그니처가 없는 일부 유형의 파일에서는 매직 넘버 기반의 방법을 사용할 수 없다. 이러한 접근 방식은 제한된 식별 가능한 파일 유형의 수, 분류 정확도의 저하, 낮은 처리 속도와 같은 여러 문제가 있다. 이러한 문제를 해결하기 위해 최근에는 신경망(Neural Network)을 사용한 파일 유형 분류기 개발 연구들이 제안되었다.

그러나 이러한 연구들에서 개발된 파일 유형 분류기는 국내에서 많이 사용되는 파일 형식 중에서 한글 워드 프로세서인 HWP 와 같은 형식을 지원하지 않고 있다. 국내에서는 HWP 파일이 흔하게 사용되고 있으며, 따라서 이러한 국내 특화 파일 형식에 대한 지원이 필요하다. 현존하는 연구 및 모델에서는 국내 점유율이 높은 파일 유형을 지원하지 않는다는 한계를 극복하고 국내 사용자들에게 높은 활용성을 제공할 수 있도록 한글 워드 프로세서 등의 파일 형식에 대한 추가적인 연구와 기능 개발이 필요하다.

본 연구에서는 CNN(합성곱 신경망)과 GRU(게이트 순환 유닛)를 활용하여 파일 유형을 기존보다 더 정확하게 식별하는 방법을 제안한다. FFT-75 라는 가장 큰 파일 조각 데이터셋과 직접 수집한 국내 파일 유형의 조각 데이터셋에서 제안모델은 512 바이트 크기의 조각에서 평균 68.2%, 4096 바이트 크기의 조각에서 평균 81.4%의 우수한 정확도를 보였다. 그 뿐만 아니라, 국내에서 자주 사용되는 파일 유형을 식별하는 기능을 통해 국내 디지털 포렌식 분야에 대한 적용 가능성을 강조하고자 한다. 이를 통해 범죄 수사 및 사법 과정에서의 증거물 분석 과정을 향상하는 데 이바지할 것으로 기대된다.

본 논문은 제 2 장에서 현재까지의 연구 동향을 살펴보고, 제 3 장에서 개발한 분류기 모델의 알고리즘에 대해 상세히 설명한다. 마지막으로 실험 결과를 통해 제안된 시스템의 성능을 검증하고, 오분류된 이유에 대해 논의한다.

II. 관련 연구

Fitzgerald et al.은 n-grams 와 bigrams, entropy 및 복잡도 측정을 사용한 SVM 기반 분류기를 제안했다. govdocs1 데이터셋에서 파생된 24 가지 파일 유형을 학습했다. Bag-of-words 모델을 사용하여 텍스트 문서를 분류했다. 텍스트 형식의 파일에 대해서는 높은 정확도를 보였지만, 오피스 문서와 같은 복합 파일 형식에서는 매우 낮은 정확도를 보였다[4].

N. L Beebe et al.은 파일 및 데이터 유형의 정확한 분류를 위해 연결된 N-gram 벡터를 활용하는 방법을 개발했다. 파일 내의 텍스트 데이터의 N-gram 통계적 특성을 분석하여 파일을 식별하는 Sceadan 을 개발하여 38 가지 다양한 파일 형식을 다루고 평균 73.8%의 정확도를 보였다[5].

Xu et al.은 512 바이트 블록을 32x32 픽셀의 흑백 이미지로 변환하고 Gabor 필터를 기반으로 512-D GIST 이미지 특징을 추출했다. 다양한 분류 알고리즘을 실험하였으며 29 가지의 파일 유형을 분류할 수 있다. 파일 비편향 모델에서 39.7%, 유형 비편향 모델에서 54.7%의 평균 분류 정확도를 보였다 [6].

Zheng et al.은 SVM 기반의 파일 유형 분류기를 제안했다. 이 연구에서는 복합 파일 유형에 다른 파일 유형이 포함되는 경우를 고려하여 12 가지 파일 유형을 12 가지 데이터 유형으로 맵핑하여 데이터 특징을 추출해 학습했다. 512 바이트 크기의 파일 조각에서 평균 정확도 88.58%를 보였다[7].

Beebe et al.은 52 가지 파일 형식을 클러스터링하여 두 단계의 계층적 구조를 제안했다. Unigrams, bigrams 및 전역 통계적 특징 (평균, 엔트로피, 복잡성 등)을 사용했으며 K-means 클러스터링을 기반으로 한 탐색적 분석을 중점적으로 다뤘다. 해당 모델은 74.1%의 정확도를 보였다[8].

Chen et al.은 2 차원 합성곱 신경망 기반 모델을 제안했다. 4,096 바이트의 파일 조각을 64x64 픽셀의 흑백 이미지로 변환한 데이터를 학습했다. 16 개의 파일 유형을 구분할 수 있으며, 70.9%의 정확도를 보였다. 그러나 텍스트 파일 및 오피스 문서 유형에 대해 상당히 낮은 정확도를 보였다는 한계가 있다 [9].

Manish Bhatt et al.은 최적화된 SVM 을 사용하는 계층적 기계 학습 기반 접근법을 제안했다. 14 가지 파일 유형, 512 바이트 크기의 1,000 개 조각 데이터셋을 사용했으며, 평균 정확도 67.78% F-1 점수 65%를 보였다[10].

Anirudh Bhat et al.은 각 파일에 대해 계산된 2 바이트 시퀀스 히스토그램 특징 벡터를 이용하여 Feed-Forward 신경망을 사용한 접근법을 제안했다. 공개된 Govdocs1 데이터셋으로 모델을 학습했고, 13 개의 파일 유형을 구분할 수 있으며, 1,024 개와 512 개의 뉴런을 가진 2 개의 은닉층을 포함한 모델이 90.32%의 정확도를 보였다[11].

Govind Mittal et al.은 1 차원 합성곱 신경망(CNNs)를 기반으로 75 개의 파일 유형을 식별하는 도구인 FiFTy 를 개발하였다. 가장 많은 75 가지 파일 유형을 가진 FFT-75 데이터셋을 공개하여 이후 관련 연구에 많은 영향을 주었다. 512 바이트와 4,096 바이트 조각 크기에 대해 각각 65.6%, 77.5%의 정확도를 달성하였다[12].

Haque, Md. Enamul 과 Mehmet Engin Tozal 은 바이트 임베딩 (Byte2Vec)을 사용하여 조각을 밀집된 벡터 표현으로 매핑하는 특징 생성 모델을 제안했다. Byte2Vec 은 특징 추출에 사용되었으며, 최근접 이웃 (kNN)은 분류에 사용됐다. 61 개의 파일 유형을 분류할 수 있으며 72%의 정확도, 74%의 정밀도, 72%의 재현율을 보였다[13].

Saaim et al.은 depthwise separable convolutions(DSCNN)을 사용한 경량형 파일 조각 분류 모델을 제안했다. FFT-75 데이터셋 시나리오 #1 에서 각각 512 바이트와 4,096 바이트의 조각 크기에 대해 65.89% 및 78.45%의 평균 정확도를 보였다 [14].

Ghaleb et al.은 Squeeze-and-Excitation 를 사용한 DSCNN 기반 경량형 모델(DSCSE)을 제안했다. 기존의 DSCNN 기반 모델[14]보다 4,096 바이트 조각에서 6.3 배 적은 부동 소수점 연산(FLOPs)으로 유사한 정확도를 달성하고, 512 바이트 조각에서는 87 배 적은 연산을 보였다. FFT-75 데이터셋 시나리오 #1 에서 각각 512 바이트와 4,096 바이트 파일 조각에 대해 66.33% 및 79.27%의 평균 정확도를 보였다 [15].

Nan Zhu et al.은 CNN 과 LSTM 을 기반으로 한 파일 조각 유형 식별 네트워크 구조를 제안했다. 이 구조는 먼저 훈련 가능한 임베딩 레이어를 사용하여 희소 이진 파일 조각을 조밀한 실숫값 표현으로 변환한다. 이후 연속적인 합성곱 모듈을 사용하여 더 높은 수준의 파일 조각의 표현을 학습하고, 이러한 특징들은 LSTM 에 입력되어 분류된다. 심층 신경망 구조는 FFT-75 을 사용해 학습했다. 512 바이트 및 4,096 바이트 파일 조각에 대해 평균 정확도가 각각 66.5% 및 78.6%로, 기존 연구보다 높은 성과를 보였다[16].

III. CNN 및 GRU 기반의 파일 유형 분류 모델

이 장에서는 최신 CNN 모델을 활용하여 한국에서의 사용 빈도가 높은 파일 유형을 분류하는 기능을 추가하고 성능을 개선하여 국내 디지털 포렌식 수사에 도움을 줄 모델을 제안한다. 그림 1 과 그림 2 는 각각 512 바이트, 4,096 바이트 조각에 대한 학습 및 분류 네트워크 아키텍처를 그림으로 나타낸 것이며, 표 1 은 아키텍처에 대한 하이퍼 파라미터의 값, 매개변수, 배치 크기 등을 정리한 표이다. 임베딩 레이어 계층과 연속 합성곱 모듈 구성은 Govind Mittal et al.[12]과 Nan Zhu et al.[16]의 모델을 참고했다.

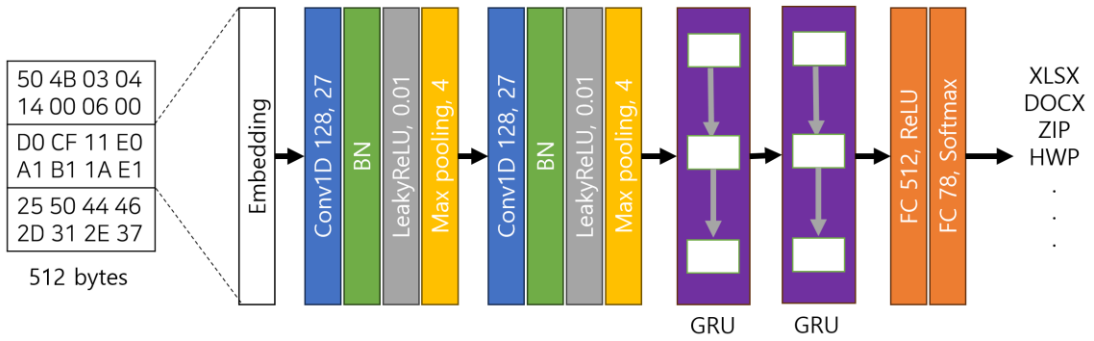

Figure 1. Illustration of the proposed network architecture for 512 bytes
그림 1. 512 바이트 조각에 대한 네트워크 아키텍처

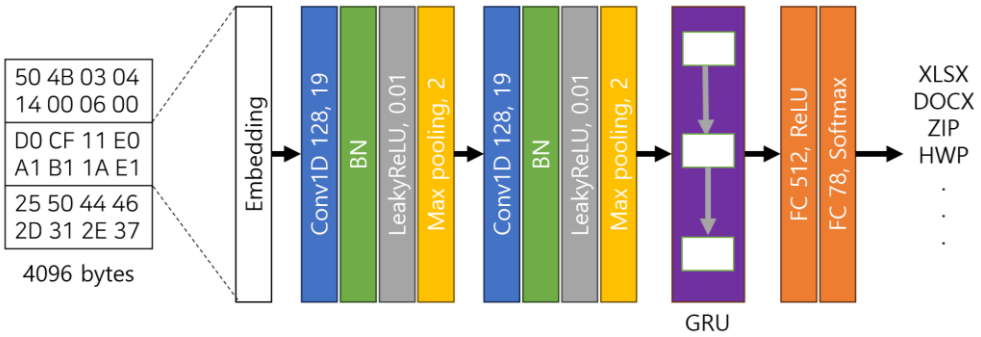

Figure 2. Illustration of the proposed network architecture for 4,096 bytes
그림 2. 4,096 바이트 조각에 대한 네트워크 아키텍처

Table 1. Detailed Model Architecture
표 1. 전체 모델 아키텍처 상세

Block size	Model	Params	batch size	epochs
512	E(64)-C1D(128,27)-BN-LR(0.01)-MP(4)-C1D(128,27)-BN-LR(0.01)-MP(4)-GRU(512)-GRU(512)-FC(512)-FC(78)	3,545,422	128	5
4096	E(32)-C1D(128,19)-BN-LR(0.01)-MP(2)-C1D(128,19)-BN-LR(0.01)-MP(2)-GRU(512)-FC(512)-FC(78)	1,687,374	256	3

Layers: [E] embedding (embedding vector length); [C1D] convolution (filter size, stride); [BN] batch normalization; [LR] leakyrelu (alpha); [MP] 1-D max-pooling; [GRU] gated recurrent unit (units); [FC] fully connected (units);

3.1 학습 가능한 임베딩 레이어

임베딩 레이어는 파일 조각 데이터의 각 바이트를 숫자로 표현된 고차원의 실숫값 벡터로 바꾸는 역할을 한다. 자연어 처리에서 단어를 벡터로 표현하는 임베딩과 유사한 원리를 이용했다. 각 바이트는 고유한 번호로 대응되고, 그 번호에 해당하는 임베딩 벡터로 변환된다. 이를 통해 단일 바이트 배열을 64 개 또는 32 개의 의미 있는 모델 학습용 실숫값으로 줄일 수

있다. 임베딩 레이어는 각 바이트를 실수 표현으로 변환하는 데 가장 효율적이며, 학습 모델이 파일 조각을 더 잘 이해할 수 있도록 해준다. 이어지는 합성곱 신경망 모듈에서 pooling 연산으로 이 차원을 감소해 나간다[12].

3.2 연속 합성곱 모듈

Convolutional Neural Network(CNN)는 주로 이미지 인식 및 패턴 인식 작업에 사용되는 딥러닝 아키텍처이다. CNN 은 데이터에서 특징을 추출하고 계층적으로 패턴을 학습하여 복잡한 패턴과 의미 있는 정보를 인식할 수 있는 모델이다.

임베딩 레이어에서 얻은 특징 맵은 연결되어 있는 2 개의 합성곱 모듈에 전달된다. 그림 1 과 그림 2 에 나온 것처럼 각 합성곱 모듈에는 1 차원 합성곱 레이어, 배치 정규화 레이어, Leaky Rectified Linear Unit (LeakyReLU) 및 max-pooling 계층으로 구성되어 있다. 두 합성곱 레이어의 출력 채널 수는 128 이고, 512 바이트에 사용된 모델에서 출력 특징 맵 크기는 각각 (243, 128), (108, 128)이다. 4,096 바이트에 사용된 모델에서는 각각 (2008, 128), (1009, 128)이다. 이 합성곱 모듈을 통해 파일 조각 데이터에서 특징을 추출할 수 있다.

3.3 결정 모듈

Gated Recurrent Unit (GRU) 레이어는 순환 신경망(RNN)의 한 형태로, 주로 연속 데이터에서 장기적인 의존성을 학습하고 기억하기 위해 사용된다. GRU 는 현재 시점의 은닉 상태를 중심으로 동작하며, 초기화 게이트와 업데이트 게이트를 활용하여 정보의 유지와 갱신을 조절한다. 초기화 게이트는 이전 정보를 얼마나 유지할지를 결정하며, 업데이트 게이트는 현재 입력을 얼마나 반영할지를 제어한다.

GRU 는 LSTM 과 비슷한 성능을 제공하면서도 더 간결한 구조로 되어 있어 학습 파라미터의 수가 적다. 이로써 모델을 효율적으로 학습시키고 결과는 유사한 성능이 나타난다. GRU 는 자연어와 같은 다양한 연속적 데이터 처리 작업에서 효과적으로 사용되며, 특히 LSTM 과 유사한 성능을 제공하는 동시에 계산 효율성을 높이는 측면에서 주목받고 있다.

본 제안모델에서 GRU 네트워크는 파일 조각에서 중요한 바이트열을 식별하기 위해 사용된다. GRU 유닛은 합성곱 모듈의 출력을 입력으로 받아 바이트 간의 연속적인 특징을 학습한다. 512 바이트 모델에서는 이중 GRU 레이어를 사용했고 4,096 바이트 모델에서는 단일 GRU 레이어를 사용했다. 이는 학습 시간 및 학습 정확도 등을 고려하여 결정했다. GRU 의 마지막 출력은 FC 레이어로 전달된다.

최종 출력은 두 개의 FC 레이어를 통해 형성한다. 첫 번째 FC 레이어는 ReLU 활성화 함수를 사용하여 출력을 활성화한다. 두 번째 FC 레이어는 SoftMax 활성화 함수를 사용하여 최종 결과를 출력한다.

IV. 실험 결과 및 분석

이 장에서는 제안한 방법을 세 가지 기준과 비교한 결과를 제시한다. 먼저, 사용된 데이터셋인 FFT-75 와 추가로 수집한 데이터를 소개하고 실험 환경을 설명한다. 그런 다음 CNN-CO, FiFTy 및 CNN-LSTM 을 포함한 세 가지 비교 기준 모델을 설명한다. 마지막으로 정확도 비교 결과를 제시하고 분류 오류의 원인에 대해 설명한다.

4.1 데이터셋

본 논문에서는 가장 많은 파일 유형을 가지고 있고 각 파일의 수가 균등하게 존재하는 파일 조각 데이터셋인 FFT-75 [17]를 학습과 성능 평가에 사용했다. 이 데이터셋에는 11 개의 RAW 사진 형식, 6 개의 비트맵 사진 형식, 3 개의 벡터 그래픽 형식, 7 개의 동영상 형식, 13 개의 압축 형식, 4 개의 실행 프로그램 형식, 7 개의 문서 포맷, 4 개의 게시물 형식, 9 개의 사람이 읽을 수 있는 텍스트 파일 형식, 7 개의 오디오 형식 및 기타 포맷 4 개(Misc 로 표시)를 포함한 75 가지

다른 파일 유형 조각이 포함되어 있다. 모든 형식의 자세한 목록은 표 3 에서 확인할 수 있다. 다양한 파일 시스템의 클러스터 크기를 수용하기 위해 512 바이트 및 4,096 바이트 블록 모두를 대상으로 한다.

또한 국내 컴퓨터 사용 환경에 맞추기 위하여 HWP, ALZ, EGG 파일을 수집하였다. 해당 확장자 파일들은 학습을 위한 대량의 데이터셋이 존재하지 않으므로, 공개된 파일들을 직접 수집하거나 파일 변환작업을 통해 획득하였다. HWP 파일은 공개된 샘플 파일과 공공기관 홈페이지의 보도자료의 첨부파일 등에서 총 1,113 개를 수집하였다. ALZ 과 EGG 는 Govdocs1 데이터셋 [18] 중에서 csv, docx, jpg, log, pptx, rtf, swf, xlsx 파일들을 대상으로 알집 프로그램의 각각 압축하기 기능을 사용하고 압축률 보통 옵션으로 압축하여 파일을 획득하였다. 이후 각 파일의 512 바이트 및 4,096 바이트 조각을 잘라내어 훈련(80%), 검증(10%), 테스트(10%) 세 집합으로 균등하게 나누었다. 각각의 크기별로 추출하는 과정에서 해당 크기에 미달하는 파일은 데이터셋에서 제외했다. 그 후에 FFT-75 데이터셋의 각 세 집합에 병합한 뒤에 무작위로 배열 순서를 섞었다.

4.2 비교 기준 모델

본 제안모델의 성능을 검증하기 위해 세 가지 기준 모델과 비교한다: (1) 바이트 동시 등장 행렬(Co-occurrence Matrix)(CNN-CO 로 표시)에 대해 훈련된 합성곱 신경망(CNN) 모델. 이 CNN 은 입력으로 크기가 128x128 인 다운샘플링된 동시 등장 행렬을 사용하며 크기가 3 이고 필터 수가 48 인 네 개의 2-D 합성곱 레이어와 크기가 64 및 75 인 두 개의 밀집 레이어로 구성된다. 이 모델은 기존 연구 모델의 성능 비교에도 이용되었다. 각 파일 유형별 정확도 결과값은 해당 논문에 표기된 값을 인용하였다. (2) CNN 기반 파일 단편 조각 유형 식별 방법 [8] (FiFTy 로 표시). 이 모델은 1-D 합성곱 레이어를 기반으로 한 구조로 되어 있다. (3) 최신 CNN-LSTM 기반 파일 단편 조각 유형 식별 방법 [10] (CNN-LSTM 으로 표시). 이 모델은 1-D 합성곱에 LSTM 레이어를 이어 붙인 구조로 구성되어 있으며, 가장 높은 정확도를 보여준 모델이다.

4.3 실험 환경

데이터 조각 크기 별로 각각 실험 환경이 서로 다르다. 각 실험 시스템별 사양은 표 2 와 같다.

Table 2. System Specifications
표 2. 시스템 사양

Data Fragment Size	512	4096
Platform	Kaggle	Google Cloud Platform - a2-highgpu-1g
Tensorflow Version	2.13.0	2.11.0
CPU	-	12 vCPU
RAM	29GB	85GB
GPU	NVIDIA Tesla P100	NVIDIA Tesla A100 40GB
VRAM	16GB	HBM2 40GB

모델 학습은 각각 최대 10 epochs 를 수행했으며, 1 epoch 마다 학습된 가중치를 별도의 파일로 저장하여 학습 정확도 및 검증 정확도가 가장 높을 때를 최종 결과로 사용했다. 그리고 편리한 비교를 위해 각 파일 조각 유형의 분류 정확도 표현에 오차 행렬을 사용했다.

4.4 실험 결과

표 3 은 제안모델과 다른 기준 모델 간의 분류 정확도를 정리한 표이다. 정확도가 가장 높은 유형의 개수가 512 바이트 조각의 경우, 제안모델이 36 개로 가장 많고 FiFty 가 10 개로 가장 적다. 이어서 CNN-CO 가 20 개, 최신모델인 CNN-LSTM 이 25 개이다 (둘 이상의 모델이 같은 유형에 대해 가장 높은 정확도를 가진 경우, 모두 가장 높은 것으로 취급한다). 4,096 바이트 조각의 경우 제안모델 36 개, CNN-LSTM 25 개, CNN-CO 20 개, FiFty 10 개 순이다. 제안 모델이

최신 모델인 CNN-LSTM 보다 11 개나 더 많다는 것은 더 많은 유형의 파일을 더욱 정확하게 분류할 수 있다는 것을 의미한다. 512 바이트 조각에서 제안모델의 평균 정확도는 68.2%로, 이는 CNN-CO 의 64.4%, FiFTy 의 65.6% 및 CNN-LSTM 의 66.5%보다도 높다. 4,096 바이트 조각에서 제안모델의 평균 정확도는 81.4%로, CNN-CO 의 75.3%, FiFTy 의 77.5% 및 CNN-LSTM 의 78.4%와 큰 차이가 난다. 이번 제안모델에 3 가지 파일 유형이 추가되었음에도 가장 높은 평균 분류 정확도를 달성하여 제안모델이 최고의 성능을 가지고 있다는 것을 알 수 있다. 추가된 3 개 유형의 정확도도 거의 100%에 가까운 정확도를 보여주고 있다.

Table 3. Result
표 3. 결과

		512-byte				4096-byte			
Filetype	**Tag**	**CNN-CO**	**FiFTy**	**CNN-LSTM**	**CNN-GRU**	**CNN-CO**	**FiFTy**	**CNN-LSTM**	**CNN-GRU**
ARW	Raw	97.8	97.8	96.4	**98.0**	**98.8**	98.9	98.3	98.7
CR2	Raw	82.4	86.7	87.6	**90.2**	95.3	94.8	95.6	**97.2**
DNG	Raw	81.6	**82.7**	80.0	81.9	96.7	96.1	**97.0**	93.7
GPR	Raw	98.7	99.2	**99.4**	**99.4**	99.5	**99.9**	**99.9**	99.7
NEF	Raw	89.3	87.7	86.4	**90.1**	89.0	95.3	**95.6**	95.1
NRW	Raw	96.1	**96.9**	95.0	96.0	96.7	97.7	98.0	**98.7**
ORF	Raw	84	86.3	**87.4**	83.9	97.6	96.3	96.9	**98.2**
PEF	Raw	**96.1**	95.1	94.3	94.9	98.9	99.1	**99.2**	98.8
RAF	Raw	97.9	98.3	98.6	**98.7**	91.3	87.1	89.4	**96.0**
RW2	Raw	**96.6**	96.5	93.8	96.2	97.9	97.9	**98.0**	97.9
3FR	Raw	99.4	99.6	**99.8**	**99.8**	92.6	99.5	**99.6**	**99.6**
JPG	Bitmap	77.5	83.5	87.0	**88.9**	**92.5**	86.3	90.3	91.5
TIFF	Bitmap	90	**96.1**	95.6	94.7	98.6	**99.0**	97.4	97.3
HEIC	Bitmap	2.0	31.7	**36.2**	1.3	4.0	**49.5**	46.5	13.5
BMP	Bitmap	97.5	98.0	**98.1**	97.5	99.4	98.4	99.6	**99.7**
GIF	Bitmap	84.5	93.5	**93.6**	**93.6**	98.5	99.1	99.3	**99.4**
PNG	Bitmap	47.6	67.2	**74.2**	73.0	72.9	88.0	85.4	**92.6**
AI	Vector	25.5	23.3	25.0	**30.0**	50.9	57.7	55.4	**79.9**
EPS	Vector	**98.8**	98.7	**98.8**	98.2	98.0	95.8	**99.0**	98.8
PSD	Vector	93.7	85.3	**95.4**	94.1	95.6	95.9	96.0	**96.2**
MOV	Video	8.7	6.1	**15.2**	9.2	0.0	18.5	16.5	**50.4**
MP4	Video	**14.8**	1.6	12.3	3.5	62.2	71.8	68.5	**74.1**
3GP	Video	**91.6**	85.6	86.0	86.5	98.8	98.1	98.9	**99.1**
AVI	Video	13.8	10.9	**14.8**	11.8	64.9	67.1	**70.1**	67.8
MKV	Video	**90.5**	88.0	88.0	88.2	99.1	98.4	**99.3**	98.8
OGV	Video	67.4	57.3	65.0	**73.3**	94.5	94.9	94.9	**96.8**
WEBM	Video	39.6	39.8	**42.7**	40.9	71.5	78.1	**79.5**	78.9
APK	Archive	32.5	29.9	31.2	**35.2**	48.3	49.7	47.4	**63.5**
JAR	Archive	36.3	41.2	44.3	**48.1**	74.8	73.8	74.0	**76.3**
MSI	Archive	5.1	10.1	9.8	**11.8**	26.6	60.1	57.5	**64.7**
DMG	Archive	7	18.2	16.4	**22.6**	17.4	19.4	24.8	**35.8**
7Z	Archive	0.9	0.0	7.6	**32.5**	**53.2**	0.1	23.2	12.1
BZ2	Archive	8.6	13.9	11.6	**14.4**	75.8	80.6	**82.6**	76.8
DEB	Archive	9.9	13.8	12.7	**14.0**	1.5	0.8	2.9	**4.2**
GZ	Archive	11.3	13.2	15.0	**17.6**	**53.0**	42.1	55.1	51.5

PKG	Archive	5.6	4.5	7.3	**9.1**	41.0	62.7	57.5	**59.1**
RAR	Archive	13	24.0	23.6	**26.8**	18.0	46.6	39.4	**39.9**
RPM	Archive	7.9	**13.6**	12.5	13.4	10.7	21.9	22.4	**35.4**
XZ	Archive	0.0	0.0	6.1	**8.8**	0.0	12.1	9.5	**13.3**
ZIP	Archive	**16.2**	13.8	15.1	14.6	36.6	34.2	38.5	**58.5**
EXE	Executable	**9.2**	0.3	6.8	1.8	4.3	2.7	4.9	**6.2**
MACH-O	Executable	93.6	92.6	**94.2**	91.6	95.7	95.0	**96.0**	95.8
ELF	Executable	**87.6**	85.9	**87.6**	83.8	92.2	86.4	**92.8**	91.7
DLL	Executable	91.7	91.1	92.2	**93.1**	94.4	91.7	94.5	**94.8**
DOC	Office	**91.4**	87.2	89.0	90.0	91.6	88.9	90.0	**93.6**
DOCX	Office	14.7	**19.4**	17.7	17.1	49.9	60.6	63.4	**85.2**
KEY	Office	35.8	44.1	44.3	**46.4**	**56.7**	43.1	49.8	52.2
PPT	Office	44.2	41.3	38.2	**47.8**	58.4	53.5	55.6	**67.8**
PPTX	Office	38.3	44.4	41.3	**48.4**	32.7	36.0	37.5	**59.1**
XLS	Office	99.4	99.3	**99.5**	**99.5**	99.2	99.3	99.5	**99.8**
XLSX	Office	96.8	95.2	**96.9**	96.5	97.4	97.2	98.0	**98.3**
DJVU	Published	18	**24.6**	22.8	15.1	34.8	30.4	**40.8**	39.2
EPUB	Published	29.3	20.7	24.8	**31.9**	74.3	79.7	**80.0**	56.6
MOBI	Published	72.5	72.4	**73.0**	72.4	74.4	73.7	**74.7**	74.5
PDF	Published	21.9	23.1	22.2	**25.7**	48.3	45.8	46.7	**60.4**
MD	Human-readable	**98.0**	97.1	97.4	97.8	**99.9**	97.4	**99.9**	99.7
RTF	Human-readable	99.5	**99.7**	**99.7**	99.4	**100**	99.8	**100**	99.9
TXT	Human-readable	**95.0**	93.7	93.4	92.6	92.9	93.0	93.2	**93.6**
TEX	Human-readable	97.8	97.6	**97.9**	96.4	98.8	98.3	**98.9**	98.1
JSON	Human-readable	**99.8**	99.5	99.6	99.4	**100**	99.8	99.9	99.8
HTML	Human-readable	**98.4**	97.5	97.6	96.6	99.5	99.6	99.5	**99.7**
XML	Human-readable	**100**	**100**	**100**	99.9	**100**	**100**	**100**	**100**
LOG	Human-readable	**100**	99.9	**100**	99.9	**99.9**	**99.9**	**99.9**	**99.9**
CSV	Human-readable	**99.9**	99.6	99.6	99.2	99.0	99.7	99.6	**99.9**
AIFF	Audio	95.7	99.4	**99.7**	99.4	94.5	98.8	**99.0**	**99.0**
FLAC	Audio	66.9	**74.2**	70.4	72.8	87.8	**97.9**	95.6	93.8
M4A	Audio	92.6	94.3	97.3	**98.1**	96.5	98.2	**98.3**	98.2
MP3	Audio	94.0	94.4	95.0	**95.8**	98.9	98.9	99.0	**99.3**
OGG	Audio	84.2	90.4	90.3	**91.4**	95.4	95.5	96.0	**97.6**
WAV	Audio	99.9	**100**	**100**	**100**	74.0	98.7	**98.8**	98.7
WMA	Audio	**99.2**	98.2	98.7	99.0	**99.9**	**99.9**	**99.9**	**99.9**
PCAP	Misc	**55.4**	51.0	49.3	50.8	95.6	95.4	95.5	**96.4**
TIF	Misc	98.1	98.1	**98.7**	97.8	99.2	99.3	99.4	**99.5**
DWG	Misc	**96.5**	96.2	96.3	**96.5**	**98.8**	96.9	**98.8**	98.3
SQLITE	Misc	98.9	98.6	**99.4**	98.2	99.5	99.8	98.5	**99.9**
HWP	Office	-	-	-	**98.2**	-	-	-	**100**
ALZ	Archive	-	-	-	**100**	-	-	-	**99.1**
EGG	Archive	-	-	-	**100**	-	-	-	**100**
Total Wins		20	10	25	36	12	7	26	48
Average		64.4	65.6	66.5	**68.2**	75.3	77.5	78.6	**81.4**

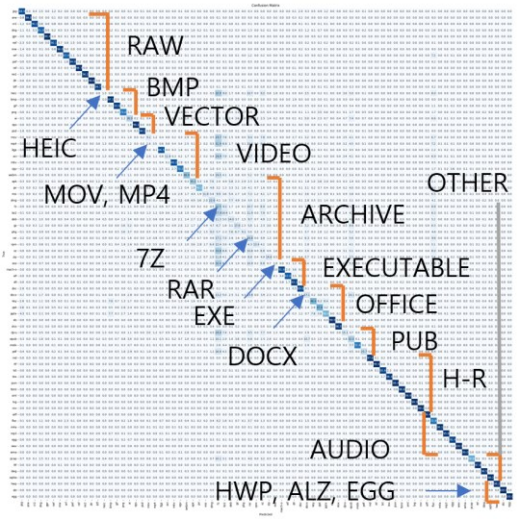

Figure 3. 512 Confusion Matrix
그림 3. 512 오차 행렬

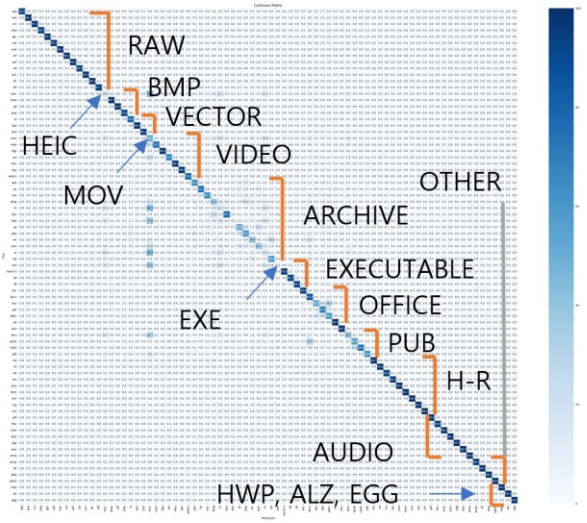

Figure 4. 4,096 Confusion Matrix
그림 4. 4,096 오차 행렬

4.5 오분류 분석

512 바이트와 4,096 바이트 크기의 조각에 대한 78 개 파일 유형에 대한 오차 행렬을 계산하여 분류 오류의 원인을 분석했다(그림 3 및 그림 4 참조). 압축 파일(BZ2, XZ, ZIP 등)에 속한 유형들의 정확도가 낮은 이유는 압축 파일이 작은 공간에 메타데이터를 포함하기 때문이다. 또한, 이 그림에서는 Bitmap, Video 및 Published 에 속한 파일 조각이 Archive 에 속한 7Z 및 RAR 로 잘못 분류되는 경우가 많다. 이는 압축 및 복합 파일 형식이 다른 유형의 파일을 포함할 수 있기 때문이다. 이미지 및 Video 파일은 특정한 압축의 형태를 가지고 있으므로 해당 파일을 BZ2, ZIP 등으로 압축하더라도 원본 파일에 큰 변화를 주지 못한다. 하지만 ALZ 와 EGG 에서는 높은 정확도를 보였는데, 이는 수집한 파일의 크기가 작아서 파일 시그니처 값 등이 포함된 것이 원인일 수 있다. 이는 충분한 크기의 파일들을 확보하고 다양한 위치에서 데이터 조각을 수집하여 추가적인 학습을 통해 개선해야 할 점이다.

4,096 바이트 크기 조각에서는 Archive 의 정확도가 향상되었다. 이는 더 큰 크기의 조각에서 더 많은 바이트 패턴들을 찾아내었기 때문인 것으로 보인다. 그러나 여전히 Archive 및 Published 에 속한 파일들이 VIDEO 의 MOV 형식으로 잘못 분류된 경우가 많았다. 서로를 포함하거나 혼동될 수 있는 유형들은 그 특성을 고려하여 별도로 분리하여 식별하거나 전용 도구를 사용하여 특징을 추출하는 등의 특별한 방법이 필요해 보인다.

HWP 파일은 대부분 한글로 작성되므로 HWP 가 높은 정확도를 보이는 것은 영어로 작성된 다른 Office 파일과 비교하여 파일 고유의 특징이 아니라 인코딩 유형에 따라 분류되었기 때문일 수 있다. 이러한 요소를 해소하기 위하여 다른 파일 유형에도 한글로 작성된 파일들을 추가 학습할 필요가 있다.

V. 결론

본 연구에서는 파일 단편 조각의 유형을 식별하는 방법을 분석하고 개발했다. 512 바이트와 4,096 바이트 크기의 파일 조각 데이터만을 사용하여 해당 파일의 유형을 식별할 수 있는 인공지능 모델을 제안했다. 기존 연구 중에서는 CNN 및 LSTM 을 사용한 모델이 가장 높은 정확도를 보였으나, 본 연구에서는 LSTM 대신 GRU 를 활용하여 모델의 정확도를 향상시켰다. 개발한 모델은 널리 사용되는 FFT-75 데이터셋에서 기존 연구 모델보다 높은 정확도를 보여주었다. FFT-75 는 현재까지 공개된 최대 규모의 파일 조각 데이터셋으로서, 이를 기반으로

한 모델의 높은 성능은 큰 의미가 있다.

더불어, 국내 시장에서 높은 점유율을 가지고 있는 HWP, ALZ 및 EGG 와 같은 유형을 분류하기 위한 기능을 모델에 추가하여 국내 환경에서의 활용성을 높였다. 이로써 해당 모델은 국내에서도 효과적으로 활용할 수 있는 기반을 제공할 수 있다.

파일 유형 식별 기술은 다양한 분야에서 활용되는데, 특히 디지털 포렌식에서 정확한 파일 유형 식별이 수사 및 증거 수집 과정에서 핵심 역할을 한다. 이는 디지털 포렌식 조사의 효율성을 높이는 데 기여할 것으로 기대한다.

그러나 아직 발전의 여지가 있다. 압축 파일이나 동영상 파일과 같은 특정 유형에서는 정확도가 낮거나 잘못된 식별이 발생하는 경우가 많았다. 한국 특화 유형 데이터셋에서는 정확도가 높으나 데이터가 편향되었을 수 있는 요소들이 있었다. 따라서 향후 연구에서는 균형 있는 충분한 크기의 데이터셋을 수집하고, 이러한 유형의 파일 조각을 더 효과적으로 식별하는 것에 특화된 신경망을 설계하여 분류 정확도를 더욱 향상시킬 계획이다.

VI. 참고문헌

[1] M. C. Amirani, M. Toorani, and A. Beheshti Shirazi, “A New approach to Content-based File Type Detection,” IEEE Symposium on Computers and Communications, 2008, pp. 1103–1108.

[2] Jonghoon Won, Minji Kang, Jisung Park, Jihong Kim, “File-Fragment Type Identification using Selected N-grams by Apriori Algorithm,” in Proc. Korea Software Congress (KSC). Gangwon State, 2018, pp. 1459-1461.

[3] F. Mansouri Hanis and M. Teimouri, “Dataset for file fragment classification of textual file formats,” BMC Res. Notes, vol. 12, no. 1, p. 801, Dec. 2019.

[4] S. Fitzgerald, G. Mathews, C. Morris, and O. Zhulyn, “Using NLP techniques for file fragment classification,” Digit. Invest., vol. 9, pp. S44–S49, 2012.

[5] 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 Trans. Inf. Forensics Security, vol. 8, no. 9, pp. 1519–1530, 2013.

[6] T. Xu, M. Xu, Y. Ren, J. Xu, H. Zhang, and N. Zheng, “A file fragment classification method based on grayscale image,” J. Comput., vol. 9, no. 8, pp. 1863–1870, 2014.

[7] N. Zheng, J. Wang, T. Wu, and M. Xu, “A fragment classification method depending on data type,” in Proc. IEEE Int. Conf. Comput. Inf. Technol.; Ubiquitous Comput. Commun.; Dependable, Autonomic Secure Comput.; Pervasive Intell. Comput., pp. 1948–1953, 2015.

[8] N. Beebe, L. Liu, and M. Sun, “Data type classification: Hierarchical class-to-type modeling,” in Advances in Digital Forensics XII (IFIP Advances in Information and Communication Technology). New Delhi, India: Springer, pp. 325–343, 2016.

[9] Q. Chen et al., “File Fragment Classification Using Grayscale Image Conversion and Deep Learning in Digital Forensics,” 2018 IEEE Security and Privacy Workshops (SPW), pp. 140–147, May. 2018.

[10] Manish Bhatt, Avdesh Mishra, Md. Wasi Ul Kabir, S. E. Blake-Gatto, Rishav Rajendra, Tamjidul Hoque and Irfan Ahmed, “Hierarchy-Based File Fragment Classification,” Mach. Learn. Knowl. Extr. 2, no. 3, pp. 216-232, 2020.

[11] Bhat, Anirudh, Aryan Likhite, Swaraj Chavan and Leena Ragha, “File Fragment Classification using Content Based Analysis,” ITM Web of Conferences, 2021.

[12] G. Mittal, P. Korus and N. Memon, “FiFTy: Large-Scale File Fragment Type Identification Using Convolutional Neural Networks,” IEEE Transactions on Information Forensics and Security, vol. 16, pp. 28-41, 2021.

[13] Haque, Md. Enamul and Mehmet Engin Tozal, “Byte embeddings for file fragment classification,” Future Generation Computer Systems, vol. 127, pp. 448-461, 2022.

[14] K. M. Saaim, M. Felemban, S. Alsaleh, and A. Almulhem, “Light-Weight File Fragments Classification Using Depthwise Separable Convolutions,” IFIP Adv. Inf. Commun. Technol., vol. 648 IFIP, pp. 196–211, 2022.

[15] M. Ghaleb, K. Saaim, M. Felemban, S. Al-Saleh, and A. Al-Mulhem, “File Fragment Classification using Light-Weight Convolutional Neural Networks,” arXiv, May 01, 2023.

[16] Nan Zhu, Yang Liu, Kun Wang and Changyou Ma, “File Fragment Type Identification Based on CNN and LSTM,” Proceedings of the 2023 7th International Conference on Digital Signal Processing, Association for Computing Machinery, New York, NY, USA, pp. 16–22, 2023.
[17] Govind Mittal, PawełKorus, and Nasir Memon, File Fragment Type (FFT)-75 Dataset [Online]. Available: http://dx.doi.org/10.21227/kfxw-8084.
[18] Simson Garfinkel, Paul Farrell, Vassil Roussev, and George Dinolt, “Bringing science to digital forensics with standardized forensic corpora,” Digit. Investig. vol. 6, pp. S2-S11, 2009.

저자소개

성민규(*Mingyu Seong*)

2024 년 2 월 아주대학교 사이버보안 전공 학사

관심분야 : Digital Forensics, Cyber Security

손태식(*Taeshik Shon*)

2005 년~2011 년 삼성전자 통신 · DMC 연구소 책임연구원
2017 년~2018 년 Illinois Institute of Technology 방문교수
2011 년~현재 아주대학교 사이버보안학과 교수

관심분야 : Digital Forensics, ICS/Automotive Secuirty