

컨볼루션 신경망(CNN)을 이용한 폭발물 성분 용량별 분류 성능 평가에 관한 연구[☆]

A Study on the Evaluation of Classification Performance by Capacity of Explosive Components using Convolution Neural Network (CNN)

이 창 현¹ 조 성 윤² 권 기 원² 임 태 호^{1*}
Chang-Hyeon Lee Sung-Yoon Cho Ki-Won Kwon Tae-Ho Im

요 약

본 논문은 컨볼루션 신경망(CNN)을 이용하여 폭발물 성분의 용량별로 분류할 때의 성능을 평가하는 연구이다. 기존의 폭발물 분류 방식 중에 IMS 증기 탐지기 방식은 폭발물의 농도가 사용자가 장비에서 설정한 임계치를 넘어야만 폭발물의 존재 여부를 판단한다. IMS 증기 탐지기는 폭발물이 존재하더라도 임계치를 넘지 않는 양이면 폭발물이 존재하지 않는다고 판단하는 문제가 있다. 따라서 폭발물 성분의 농도가 임계치를 넘지 않는 양일 때에도 폭발물 성분을 검출하는 방안이 필요하다. 이에 따라 본 논문에서는 폭발물 시계열 데이터를 Gramian Angular Field(GAF) 알고리즘으로 이미지를 변환한 후 이미지와 영상처리뿐만 아니라 시계열 데이터 처리에도 뛰어난 성능을 보이는 딥러닝 모델인 컨볼루션 신경망(CNN)으로 직접 label을 설정해서 지도학습을 진행한 결과 폭발물 성분의 농도가 임계치를 넘지 않는 양일 때에도 폭발물 성분이 존재한다고 판단함과 동시에 폭발물 성분의 종류와 폭발물 성분의 농도의 양을 같이 판단할 수 있는지 성능평가를 진행했다.

☞ 주제어 : 딥러닝, 폭발물 분류, 이상 감지 시스템

ABSTRACT

This paper is a study to evaluate the performance when classifying explosive components by capacity using a convolutional neural network (CNN). Among the existing explosive classification methods, the IMS steam detector method determines the presence or absence of an explosive only when the explosive concentration exceeds the threshold set by the user. The IMS steam detector has a problem of determining that even if an explosive exists, the explosive does not exist in an amount that does not exceed the threshold. Therefore, it is necessary to detect the explosive component even when the concentration of the explosive component does not exceed the threshold. Accordingly, in this paper, after imaging explosive time series data with the Gramian Angular Field (GAF) algorithm, it is possible to determine whether there are explosive components and the amount of explosive components even when the concentration of explosive components does not exceed a threshold.

☞ keyword : Deep learning, explosive classification, abnormality detection system

1. 서 론

군사안보지원사령부 자료에 따르면 2016년 6월에 발생한 ISIS 소속 외국인테러전투원(FTF) 3명이 일으킨 터키 이스탄불 아타튀르크 공항 자폭테러 사건처럼 다양한 이유로 중요시설을 겨냥한 테러가 일어나고 있고 공항, 지하철역 같은 인구 밀집 중요시설은 항상 위험에 노출되어 있다 [1]. 따라서 안전이 곧 수많은 생명이란 직결된 공항과 같은 중요시설에서는 폭발물 검사를 의무적으로 진행하고 있다.

¹ Division of Information and Telecommunication Engineering, Hoseo University, Asan-si, 31499, Korea

² Smart Network Research Center, Korea Electronics Technology Institute, Seoul, 03924, Korea

* Corresponding author (taehoim@hoseo.edu)

[Received 28 February 2022, Reviewed 15 April 2022(R2 25 May 2022), Accepted 02 June 2022]

☆ 이 논문은 2022년 해양수산부 재원으로 해양수산과학기술진흥원의 지원을 받아 수행된 연구임 (No.20200611, 컨테이너 위험화물 자동검색 및 복합탐지 시스템 개발)

☆ 본 연구는 과학기술정보통신부 및 정보통신기획평가원의 대학ICT연구센터지원사업의 연구결과로 수행되었음(IITP-2022-2018-0-01417)



(그림 1) X-ray 위험물 검사방식

(Figure 1) X-ray dangerous object inspection method



(그림 2) 금속탐지기 위험물 검사방식

(Figure 2) Metal detector test method for dangerous substances.



(그림 3) IMS 증기 탐지기

(Figure 3) IMS steam detector

기존의 진행되고 있는 폭발물 검사방식은 그림 1과 같은 X-ray를 이용한 이미지를 이용한 객체 인식 분류 방식과 그림 2와 같은 금속탐지기를 이용해서 사람이 사람을 직접 확인하는 방식과 그림 3과 같은 증기 탐지기 방식으로 위험물을 가열하여 특정 원소가 있을 때 위험물이 있는 것으로 판단하는 방식이 진행되고 있다 [2]. 여기서 IMS 증기 탐지기의 경우 수집된 물질을 이온화 및 전기장 내에서의 특성을 분석하여 마약이나 폭발물 성분을 탐지하는데 사용되는 방식이다 [3]. 기존 IMS 증기 탐지기의 경우 폭발물을 검출할 때 폭발물 농도가 임계치(Threshold)가 넘어야만 폭발물이 존재한다고 판단하기 때문에 이 방식은 폭발물이 존재하더라도 폭발물의 농도가 설정된 임계치가 넘지 않으면 폭발물이 있다고 판단하지 못하는 문제점이 있고, 정전기와 같은 외부적인 요인에 의해 생기는 잡음이 폭발물이 아님에도 설정한 임계치가 넘어가서 폭발물이 존재한다고 판단하는 문제점

과 데이터 주기가 일정하지 않다는 문제가 존재한다. 외부적인 요인에 의해 생기는 잡음과 주기가 일정하지 않은 문제점은 전처리 알고리즘을 설계해 잡음을 제거하고 일정하게 주기를 갖게 한 후 최근 4차 산업 혁명에서 다양한 분야에 사용되고 있는 지도학습 기반의 딥러닝 모델인 컨볼루션 신경망(CNN)을 이용하여 기존의 임계치보다 폭발물의 양이 낮게 있을 때 검출이 되지 않던 상황에서 검출이 될 수 있는지 확인할 필요가 있다. 따라서 본 논문에서는 전처리 과정을 통해 기존의 잡음이 들어간 데이터를 잡음이 없게 만들고 잡음이 없어진 데이터를 기반으로 딥러닝 모델 중 컨볼루션 신경망을 학습 후 나온 가중치 파일을 이용하여 폭발물의 종류를 분류함과 동시에 폭발물의 용량별로 분류할 수 있는지 평가를 해 보고자 한다.

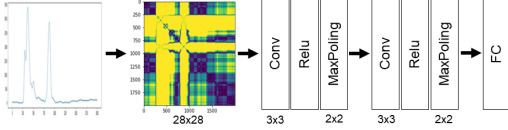
2. 관련 연구

본 논문에서는 폭발물 시계열 데이터에 이미지화 알고리즘을 통해 시계열 데이터 세트를 이미지화를 진행한 후 컨볼루션 신경망(CNN)으로 이미지화가 완료된 데이터 세트를 학습하였을 때 폭발물 분류가 정상적으로 이루어지고 용량별로 정확히 판단할 수 있는지 평가하는 방법을 제안하고자 한다. 따라서 딥러닝 방식인 컨볼루션 신경망(CNN)으로 시계열 데이터 분류 진행을 할 때 CNN이 다른 모델에 비해 좋은 성능이 나오는지 확인이 필요하며, CNN을 적용하기 위해 필요한 이미지화 알고리즘 종류와 이미지화 알고리즘 중 최고의 성능이 나오는 알고리즘이 필요하다. 본 장에서는 선행 연구를 통해 위의 필요한 내용에 대해서 다루었다.

2.1 시계열 데이터 분류 CNN과 다른 모델 비교

시계열 데이터를 Recurrence Plots(RP)라는 m차원 궤적을 이용해서 거리 행렬을 구해서 사용자가 설정한 엡실론 값보다 작은 경우 점을 찍어서 이미지화하는 알고리즘이며, RP 알고리즘은 식 (1)과 같이 표현할 수 있다. RP 알고리즘을 통해서 시계열 데이터를 이미지화를 진행 [4, 5] 후 간단한 그림 4와 같은 컨볼루션 신경망(CNN)으로 UCR 시계열 데이터 중 20개의 데이터 세트로 실험을 진행한 결과 오류율(Error rate)이 1-NN DTW, Shapelet, BoP, SAX-VSM과 같은 다른 시계열 분류 알고리즘보다 낮게 나오는 좋은 결과를 얻었다 [6].

$$R(i, j) = \begin{cases} 1 & \text{if } \|\vec{x}(i) - \vec{x}(j)\| \leq \epsilon \\ 0 & \text{otherwise} \end{cases} \quad (1)$$



(그림 4) 실험에 사용된 CNN 모델 구조

(Figure 4) CNN model structure used in the experiment

2.2 Gramian Angular Field(GAF)

Gramian Angular Field(GAF) 알고리즘은 시간적 상관 관계를 극좌표를 기반으로 표현하는 알고리즘이다. 극좌표 기반의 행렬의 경우 시계열 데이터를 이미지로 변경할 때 왼쪽 상단에서 오른쪽 하단으로 이동함에 따라 시간이 증가하기 때문에 시간 상관관계를 보존할 수 있다는 장점이 있다. GAF 알고리즘의 경우 사용되는 시계열 데이터 값이 매우 크거나 값이 매우 작을 수 있으므로 $[-1, 1]$, $[0, 1]$ 구간으로 정규화시킨 후 나온 신호 \tilde{x}_i 의 시간 인덱스인 t_i 를 반지름 r_i 로 표현하고, 각도 ϕ 와 같이 극좌표로 아래의 식 (2)처럼 변환된다. 여기서 N의 경우 극좌표계의 범위에 대한 정규화 상수이다.

$$\begin{cases} \Phi_i = \text{arccos}(\tilde{x}_i), -1 \leq \tilde{x}_i \leq 1, \text{ and } \tilde{x}_i \in \tilde{X} \\ r_i = \frac{t_i}{N}, t_i \in N \end{cases} \quad (2)$$

GAF의 경우 GASF(Gramian Angular Summation Field) 방식과 GADF(Gramian Angular Difference field) 방식으로 이루어져 있다. 먼저 GASF의 경우 i 와 j 의 시간으로 이루어진 극좌표계 시계열 데이터의 각도의 합으로 아래의 식 (3)처럼 나타낸다.

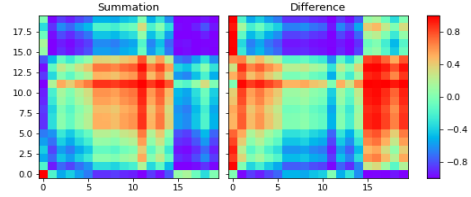
$$\begin{aligned} GASF &= [\cos(\Phi_i + \Phi_j)] \\ &= \tilde{x} \cdot \tilde{x} - \sqrt{I - \tilde{x}^2} \cdot \sqrt{I - \tilde{x}^2} \end{aligned} \quad (3)$$

GADF 방식은 GASF 방식의 반대로 극좌표계의 각도의 차로 아래의 식 (4)처럼 나타낸다.

$$\begin{aligned} GADF &= [\sin(\Phi_i - \Phi_j)] \\ &= \sqrt{I - \tilde{x}^2} \cdot \tilde{x} - \tilde{x} \cdot \sqrt{I - \tilde{x}^2} \end{aligned}$$

(4)

GASF와 GADF 방식을 시계열 데이터의 적용하면 아래의 그림 5와 같이 변화된다 [7].



(그림 5) (왼) GASF, (오) GADF

(Figure 5) (left) GASF, (right) GADF.

2.3 이미지화 알고리즘 종류별 CNN 적용

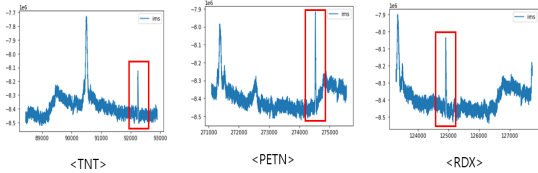
이미지화 알고리즘인 Gramian Angular Field(GAF)와 Markov Transition Field(MTF)를 Intel i7-3630QM CPU와 8 GB 메모리 환경에서 12개의 각각 다른 데이터 세트로 CNN 학습을 진행 후 나온 가중치로 평가를 진행했을 때 GAF가 MTF보다 단 2개의 테스트 데이터 세트를 제외하고는 10개의 테스트 데이터 세트에서 오류율(Error rate)이 낮게 나왔고, 평균 오류율도 GAF는 0.238, MTF는 0.313으로 GAF가 MTF보다 낮은 것을 확인할 수 있다. 그리고 CNN에 GAF와 MTF 방식을 적용했을 때 기존의 시계열 분류방법인 INN-EUCLIDEAN, INN-DTW, FAST SHAPELET, BOP, SAX-VSM 방식보다 오류율(Error rate) 낮게 나온 것을 확인할 수 있다 [8, 9].

3. 폭발물 시계열 데이터의 문제점

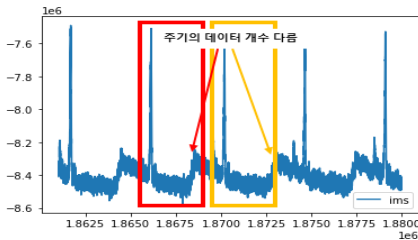
IMS 증기 탐지기에서 뽑아낸 폭발물의 시계열 데이터를 컨볼루션 신경망(CNN)을 적용하기 위해서는 폭발물 시계열 데이터에 존재하는 두 가지의 문제점을 해결해야 한다. 폭발물 시계열 데이터의 첫 번째 문제점은 아래의 그림 6과 같이 빨간색 사각형이 있는 부분이 정전기와 같은 외부 요인에 의해 생기는 잡음이며, 이 잡음의 경우 사용된 폭발물 시료인 TNT 10ng, 50ng, 150ng, 200ng, PETN 10ng, 50ng, 150ng, 200ng, RDX 10ng, 50ng, 150ng, 200ng에서 모두 발생했다.

두 번째 문제점은 폭발물 시계열 데이터의 주기가 아래의 그림 7처럼 일정하지 않다는 점이다. 폭발물 시계열 데이터를 이미지화를 하려면 주기 단위로 나누어서 진행해야 하는데 주기별로 데이터의 개수가 차이가 있다. 따

라서 위의 두 문제점을 해결하기 위해 폭발물 시계열 데이터의 잡음을 제거하는 전처리 과정과 정확하게 이미지화를 진행할 수 있도록 주기를 정확히 나눌 수 있는 전처리 과정이 필요하다.



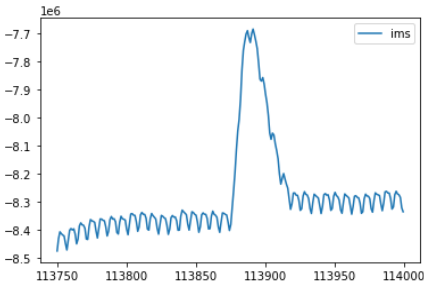
(그림 6) 폭발물 시계열 데이터의 잡음
(Figure 6) Noise in explosive time series data.



(그림 7) 폭발물 시계열 데이터의 불규칙한 주기
(Figure 7) Irregular cycle of explosive time series data.

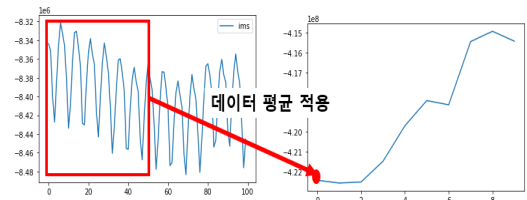
3.1 전처리 알고리즘 : 폭발물 데이터 잡음 제거

폭발물 시계열 데이터 분류 실험에서 사용된 폭발물인 TNT, PETN, RDX에서 발생된 잡음 데이터는 아래의 그림 8처럼 여러 개의 데이터로 이루어져 있고, 일정한 위치에서 생성되는 것이 아닌 임의의 위치에서 발생되기 때문에 제거가 쉽지 않다.

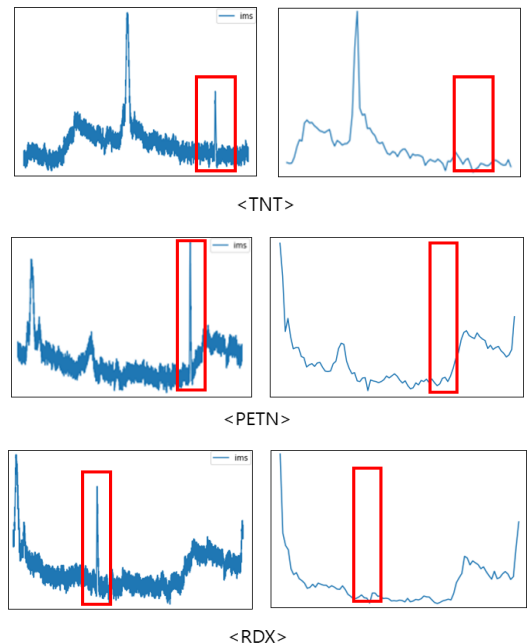


(그림 8) 폭발물 시계열 데이터의 잡음 확대
(Figure 8) Noise enlargement of explosive time series data.

폭발물 데이터의 기존 주기를 변화해서 잡음을 제거하는 방식을 적용했다. 기존 주기의 경우 평균 4455개의 데이터로 이루어져 있으며, 최소 1000000개 이상, 최대 5000000개 이상의 데이터로 이루어져 있다. 이는 폭발물의 종류와는 무관하게 평균 주기는 TNT, PETN, RDX 모두 같다. 그림 9와 같이 기존의 데이터를 100개, 50개씩, 10개씩 등등 데이터씩 묶어서 평균을 내는 여러 번의 실험을 진행한 결과 기존의 데이터를 50개씩 묶어서 평균을 낸 값을 적용했을 때 정상적으로 잡음이 그림 10처럼 제거가 되었다.



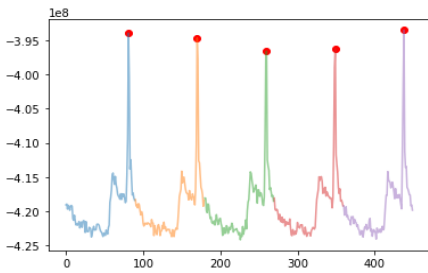
(그림 9) 폭발물 시계열 데이터의 잡음 제거
(Figure 9) Noise removal from explosive time series data.



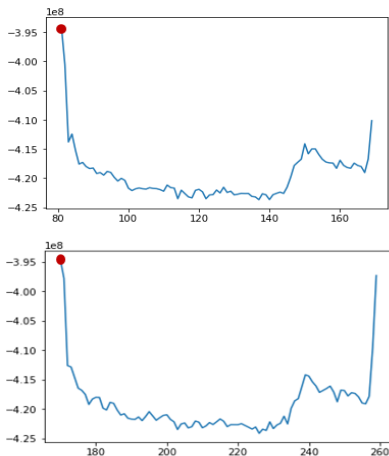
(그림 10) 잡음 제거 전처리 후 폭발물 시계열 데이터
(Figure 10) Explosive time series data after preprocessing to remove noise.

3.2 전처리 알고리즘 : 정확한 주기 산출

폭발물 시계열 데이터를 첫 번째 전처리 과정인 데이터의 잡음을 제거하는 방식이 끝난 후 이미지화를 위해서 데이터를 정확한 간격으로 자른 후 진행해야 한다. 하지만 폭발물의 데이터의 특성상 주기가 일정하지 않고, 잡음 제거 전처리 후 평균 주기가 90개의 데이터로 이루어져 있다는 것을 파악했지만 시계열 데이터 이미지화 알고리즘의 특성상 데이터의 개수의 영향을 많이 받아 원하지 않는 방향으로 결과를 얻을 수 있기에 평균적인 데이터 주기보다 더 정확하게 주기를 나누어야 한다. SciPy 라이브러리의 `find_peaks` 알고리즘 [10]을 사용해서 폭발물 시계열 데이터의 특징인 RIP라는 값을 그림 11처럼 빨간 점으로 표시했다. RIP 값은 IMS 검출을 진행할 때 일정하게 주기마다 나오는 의미가 없는 값이다.



(그림 11) 폭발물 시계열 데이터의 RIP 값
(Figure 11) RIP value of explosive time series data.



(그림 12) 전처리 과정으로 정확한 주기 나누기
(Figure 12) Divide the cycle accurately through the preprocessing process.

RIP라는 기준데이터를 이용해서 다음 RIP가 나올 때 다음 주기로 설정하는 전처리 과정을 통해 정확한 주기로 그림 12처럼 나눌 수 있었다.

4. 실험 환경 및 실험 결과

본 장에서는 직접 폭발물 시료를 용량별로 휘발시켜 IMS 증기탐지기를 통해 취득한 시계열 데이터를 사용해서 실험을 진행했으며, 실험에서 사용된 데이터 세트의 구성과 실험 환경을 소개한다. 또한 폭발물 시계열 데이터를 이미지화를 통해 CNN으로 훈련해서 나온 가중치를 적용했을 때 나오는 실험 결과에 대해서 제시하며, CNN을 `simpleRNN`, `LSTM` 등 다른 시계열 분류 모델과 비교하여 CNN이 폭발물 분류에 뛰어난지 결과를 비교했다.

4.1 실험 환경 및 데이터 세트 구성

본 논문에서 실험 환경은 CPU i7-10700 2.90 GHz, RAM 16GB, GPU RTX 3070 8GB에서 실험을 진행했다. 데이터 세트는 훈련 데이터 세트와 테스트 데이터 세트로 구분했으며, 모든 데이터 세트는 실제 3가지의 폭발물 시료를 농도별로 IMS 증기 탐지기 Smear matrix에 용매를 휘발시킨 후 0초에서 10초까지 스캔을 하였을 때 나오는 데이터로 사용했다. 훈련 데이터 세트는 아래의 표 1과 같이 구성되어있고, 테스트 데이터 세트의 경우는 아래의 표 2와 같이 구성해서 실험을 진행했다. 아래의 표 2에 나온 데이터는 1개당 1개의 주기를 의미하며, 이미지화를 위해 원본 데이터 파일을 전처리해서 따로 주기별로 분리된 상태로 사용했다. 훈련 데이터 세트와 테스트 데이터 세트는 모두 다른 데이터이며 정확한 성능 검증을 위해 중복이 되지 않도록 했다.

(표 1) 훈련 데이터 세트 구성

(Table 1) Training dataset configuration.

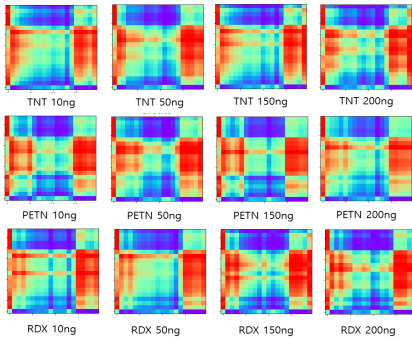
	10ng	50ng	150ng	200ng
TNT	150	199	381	343
PETN	185	230	393	343
RDX	355	373	498	658

(표 2) 테스트 데이터 세트 구성

(Table 2) Configuration of test data sets.

	10ng	50ng	150ng	200ng
TNT	30	30	30	30
PETN	30	30	30	30
RDX	30	30	30	30

CNN을 적용하기 위해 모든 데이터 세트는 GAF 알고리즘의 GADF 방식으로 그림 13처럼 이미지화했고, 이미지화된 데이터 세트는 모델 적으로 객체 인식속도가 빠른 One Stage Detector 방식의 CNN 기반 모델인 YOLOV4 [11]로 훈련 데이터 세트를 labeling 해준 뒤 훈련하여 나온 가중치를 이용해서 테스트 데이터 세트로 검증하는 폭발물 시계열 데이터 분류 실험을 진행했다.



(그림 13) 폭발물 시료 농도별 데이터 이미지화

(Figure 13) Imaging data by explosive sample concentration.

4.2 실험 과정 및 실험 결과

폭발물 같은 위험물은 정확히 판단될 수 있는지 실험을 통해 확인해야 하므로 YOLOV4의 Threshold 값을 0.8 이상으로 조정해서 실험을 진행한 결과 아래의 표 3과 같은 결과를 얻을 수 있었다.

(표 3) 폭발물 시계열 데이터 CNN 적용 실험 결과
(Table 3) Explosive Time Series Data CNN Application Experiment Results

	TNT 10ng	TNT 50ng	TNT 150ng	TNT 200ng
Accuracy	96.7%	100%	100%	93.4%
Error rate	0.033	0	0	0.066
F1-score	0.983	1.0	1.0	0.966
	PETN 10ng	PETN 50ng	PETN 150ng	PETN 200ng
Accuracy	76.7%	86.7%	83.4%	96.7%
Error rate	0.233	0.133	0.166	0.033
F1-score	0.868	0.929	0.909	0.983
	RDX 10ng	RDX 50ng	RDX 150ng	RDX 200ng
Accuracy	96.7%	96.7%	96.7%	100%
Error rate	0.033	0.033	0.033	0
F1-score	0.983	0.983	0.983	1.0

CNN이 폭발물 시계열 데이터 분류를 하는 것이 다른 모델과 비교를 해봤을 때 뛰어난지 판단하기 위해 시계열 데이터 분류 모델 중에 자주 사용되고 있는 SimpleRNN, LSTM, GRU라는 3개의 모델과 비교를 해본 결과 아래의 표 4, 표 5, 표 6과 같은 결과를 얻을 수 있었다. SimpleRNN, LSTM, GRU 실험에 사용된 데이터 세트는 CNN 실험에 사용했던 전처리 과정이 끝난 훈련 데이터의 70%를 Train data, 30%를 Validation data로 사용하고 CNN 실험에 사용된 Test data로 평가에 사용했다.

(표 4) CNN, SimpleRNN, LSTM, GRU Accuracy
(Table 4) CNN, SimpleRNN, LSTM, GRU Accuracy

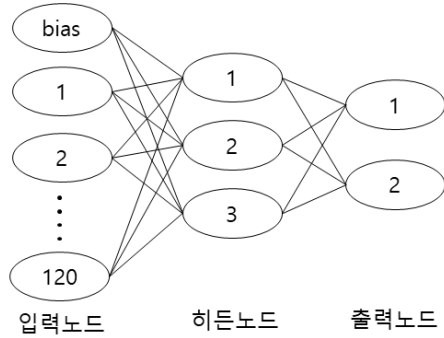
	CNN	Simple RNN	LSTM	GRU
TNT_10ng	96.7%	73.3%	83.3%	86.7%
TNT_50ng	100%	73.3%	86.7%	93.3%
TNT_150ng	100%	76.7%	86.7%	90.0%
TNT_200ng	93.4%	73.3%	80.0%	86.7%
PETN_10ng	76.7%	63.3%	70.0%	73.3%
PETN_50ng	86.7%	73.3%	73.3%	73.3%
PETN_150ng	83.4%	76.7%	83.3%	83.3%
PETN_200ng	96.7%	70.0%	76.7%	80.0%
RDX_10ng	96.7%	63.3%	80.0%	86.7%
RDX_50ng	96.7%	73.3%	80.0%	90.0%
RDX_150ng	96.7%	73.3%	80.0%	83.3%
RDX_200ng	100%	70.0%	76.7%	83.3%

(표 5) CNN, SimpleRNN, LSTM, GRU Error rate
(Table 5) CNN, SimpleRNN, LSTM, GRU Error rate

	CNN	Simple RNN	LSTM	GRU
TNT_10ng	0.033	0.267	0.167	0.133
TNT_50ng	0	0.267	0.133	0.067
TNT_150ng	0	0.233	0.133	0.100
TNT_200ng	0.066	0.267	0.200	0.133
PETN_10ng	0.233	0.367	0.300	0.267
PETN_50ng	0.133	0.267	0.267	0.267
PETN_150ng	0.166	0.233	0.167	0.167
PETN_200ng	0.033	0.300	0.233	0.200
RDX_10ng	0.033	0.367	0.200	0.133
RDX_50ng	0.033	0.267	0.200	0.100
RDX_150ng	0.033	0.267	0.200	0.167
RDX_200ng	0	0.300	0.233	0.167

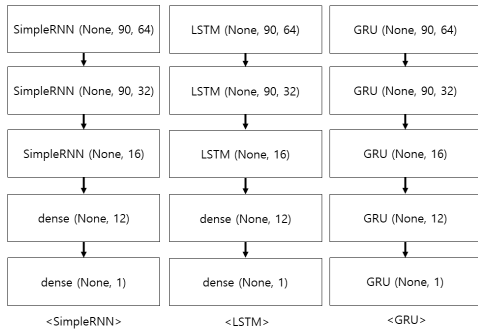
(표 6) CNN, SimpleRNN, LSTM, GRU F1-score
(Table 6) CNN, SimpleRNN, LSTM, GRU F1-score

	CNN	Simple RNN	LSTM	GRU
TNT_10ng	0.983	0.846	0.909	0.929
TNT_50ng	1.0	0.846	0.929	0.966
TNT_150ng	1.0	0.868	0.929	0.947
TNT_200ng	0.966	0.846	0.889	0.929
PETN_10ng	0.868	0.776	0.824	0.846
PETN_50ng	0.929	0.846	0.846	0.846
PETN_150ng	0.909	0.868	0.909	0.909
PETN_200ng	0.983	0.824	0.868	0.889
RDX_10ng	0.983	0.776	0.889	0.929
RDX_50ng	0.983	0.846	0.889	0.947
RDX_150ng	0.983	0.846	0.889	0.909
RDX_200ng	1.0	0.824	0.868	0.909



(그림 15) 선행 연구 신경망 구성
(Figure 15) Configuring a Proactive Neural Network

아래의 그림 14는 실험에 사용된 SimpleRNN, LSTM, GRU의 신경망 layer 구성이다. 실험 과정에서 loss function은 Categorical_crossentropy, Optimizer는 RMSprop을 사용했다.



(그림 14) SimpleRNN, LSTM, GRU 신경망 layer 구성
(Figure 14) Configure SimpleRNN, LSTM, GRU Neural Network Layer

본 논문에서 실험을 통해 나온 결과를 비교하기 위해 신경망을 이용한 중성자 보안검색 장치를 위해 연구한 폭발물 분류에 관한 선행 연구에서는 탄소(C), 질소(N), 산소(O) 3가지 성분을 이용해서 아래의 그림 15와 같이 Bias를 포함한 121개의 입력 노드와 4개의 은닉 노드, 2개의 출력 노드를 이용한 간단한 신경망으로 분류를 진행했는데 폭발물 16개의 테스트 데이터 세트로 81.25%라는 정확도를 얻었고[12], 본 논문에서 실험한 결과와 선행 연구와 비교하면 CNN을 사용해서 폭발물 분류를 진행했을 때 더 좋은 결과를 얻을 수 있었다.

5. 결 론

폭발물 데이터를 최근 화학분석 기반 폭발물 탐지 기술인 질량 분석법, 테라헤르츠 분광법, 근적외선 분광법, 화학 및 전기화학 감지기 등 여러 방법으로 분류하고자 하는 연구가 진행되고 있으나 분광법 같은 경우 시간이 많이 소요되는 문제점이 있고, 화학 및 전기화학 감지기의 경우 일정 용량이 넘어야 폭발물을 검출할 수 있는 단점이 존재한다[13]. 따라서 본 논문은 하드웨어적인 분류 방법에서의 문제점을 보완하고자 딥러닝 모델인 CNN으로 폭발물 시계열 데이터를 분류할 수 있는지 평가하는 것을 목적으로 다룬다. 하지만 폭발물 시계열 데이터의 센싱 데이터는 폭발물 시료를 구하기도 힘들고, 보유한 소량의 시료를 휘발시켜 IMS 증기 탐지기로 데이터를 얻고 실험을 진행하다 보니 제한된 데이터로 실험을 진행할 수밖에 없었다. 직접 시료를 휘발시켜 얻은 데이터 세트에 실험을 진행한 결과 TNT와 RDX는 적은 데이터 세트임에도 불구하고 이미지화를 진행했을 때 특징이 폭발물 농도별로 명확했기에 평균 인식을 90% 이상이라는 좋은 결과를 얻었다. PETN의 경우 시료의 농도별로 유사한 경우가 많아서 다른 폭발물 시료보다 좋은 수치가 나오지 않은 것을 실험을 통해 결과를 얻었다. 폭발물 시계열 데이터 분류 과정에서 CNN이 SimpleRNN, LSTM, GRU와 비교했을 때 3가지 폭발물인 TNT, PETN, RDX에서 더 높은 성능을 보였다. 적은 데이터로도 충분히 좋은 결과를 얻었지만 향후 더 많은 데이터를 확보하고, 3가지의 폭발물 TNT, PETN, RDX 말고도 더 많은 종류의 폭발물 시료 데이터를 확보해서 추가로 실험을 진행한다면 다양한 상황에 대해 좋은 결과를 얻을 수 있을 것으로 생각된다.

참고문헌(Reference)

- [1] Defense Security Support Command, “Major terrorist incidents.”, 2020.
<https://www.dssc.mil.kr/dsscckr/278/subview.do?enc>
- [2] S. Singh, M. Singh, “Explosives detection systems (EDS) for aviation security”, *Signal Process*, pp. 31-55, 2003.
[https://doi.org/10.1016/S0165-1684\(02\)00391-2](https://doi.org/10.1016/S0165-1684(02)00391-2)
- [3] Sin Jin KIM, Hark Sang KIM, “Analysis of IMS Signal”, *The Institute of Electronics and Information Engineers Academic conference*, 701-705(5 pages), 1992.
<https://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE06326427>
- [4] Marwan, N., Romano, M.C., Thiel, M., Kurths, J., “Recurrence plots for the analysis of complex systems”, *Physics Reports* 438(5-6), 237-329, 2007.
<https://doi.org/10.1016/j.physrep.2006.11.001>
- [5] Souza, V. Silva, D. Batista, G. “Time Series Classification Using Compression Distance of Recurrence Plots”, 687-696, *IEEE International Conference on Data Mining (ICDM)*, 2013.
<https://ieeexplore.ieee.org/abstract/document/6729553>
- [6] Hatami N, Gavet Y, Debayle J. “Classification of time-series images using deep convolutional neural networks” In: *International conference on machine vision*, 2017.
<https://doi.org/10.1117/12.2309486>
- [7] Wang, Z., Oates, T., “Imaging time-series to improve classification and imputation”. *Proceedings of International Joint Conference on Artificial Intelligence*, p. 3939, 2015.
<https://www.aaai.org/ocs/index.php/IJCAI/IJCAI15/paper/viewPaper/11082>
- [8] Z. Wang, T. Oates, “Encoding time series as images for visual inspection and classification using tiled convolutional neural networks,” in *Workshops at the 29th AAAI Conference on Artificial Intelligence*, vol. 1, 2015.
https://www.researchgate.net/publication/275970614_Encoding_Time_Series_as_Images_for_Visual_Inspection_and_Classification_Using_Tiled_Convolutional_Neural_Networks
- [9] LeCun, Y.; Kavukcuoglu, K.; and Farabet, “Convolutional networks and applications in vision.” In *Circuits and Systems(ISCAS), Proceedings of 2010 IEEE International Symposium on*, 253 - 256. IEEE, 2010.
<https://ieeexplore.ieee.org/abstract/document/5537907>
- [10] SciPy documentation “signal processing Peak finding”, 2021.
https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.find_peaks.html#scipy.signal.find_peaks
- [11] Alexey Bochkovskiy, Chien-Yao Wang, and HongYuan Mark Liao. “Yolov4: Optimal speed and accuracy of object detection”, *arXiv preprint arXiv:2004.10934*, 2020.
<https://arxiv.org/abs/2004.10934>
- [12] Chang Rak CHOI, ji soo KIM, Soo Hyung KIM, Cheul Muu SIM, “A Comparative Study on Neural Network Classifiers for Neuron-Type Security Device”, *Korea Information Processing Society Conference*, 3-6(4 pages), 2007
<https://www.koreascience.or.kr/article/CFKO200723367097770.page>
- [13] Sang Hyeon MOON, Won Joo LEE, Ki Young LEE, “Research Trends in Chemical Analysis Based Explosive Detection Techniques”, *The Korean Society of Industrial and Engineering Chemistry (KSIEC)*, 1-10(10 pages), 2022
<https://doi.org/10.14478/ace.2021.1073>

● 저 자 소 개 ●



이 창 현(Chang Hyun Lee)

2020년 호서대학교 정보통신공학과(공학사)
2021년~현재 호서대학교 대학원 정보통신공학과(공학석사)
관심분야 : 시계열 데이터 처리, 딥러닝
E-mail : lygy6322@naver.com



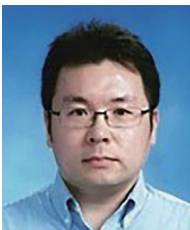
조 성 윤(Sung Yoon Cho)

2006년 연세대학교 전기전자공학과 학사
2008년 연세대학교 전기전자공학과 석사
2013년 연세대학교 전기전자공학과 박사
2013년~2020년 삼성전자 시스템 LSI사업부 책임연구원
2020년~현재 한국전자기술연구원 스마트네트워크연구센터 책임연구원
관심분야 : 5G/6G 통신시스템, 적응적 신호처리, 딥러닝
E-mail : sycho@keti.re.kr



권 기 원(Ki Won Kwon)

1997년 광운대학교 컴퓨터공학과 학사
1999년 광운대학교 대학원 컴퓨터공학과 석사
2011년 중앙대학교 대학원 전자전기공학과 박사
1999년~현재 한국전자기술연구원 스마트네트워크연구센터 센터장
관심분야 : 디지털트윈, 유무선디지털통신시스템, 해양수산ICT융합
E-mail : kwonkw@keti.re.kr



임 태 호(Tae Ho Im)

2012년 중앙대학교 대학원 전자전기공학과(공학박사)
2012년~2015년 삼성전자 DMC연구소 책임연구원
2015년~2018년 호서대학교 해양IT융합기술연구소 조교수
2019년~2020년 호서대학교 정보통신공학부 조교수
2021년~현재 호서대학교 정보통신공학부 부교수
관심분야 : 5G 이동통신, LPWAN, 수중통신, 딥러닝
E-mail : taehoi@hoseo.edu