

# 객체 탐지 모델을 활용한 전기 아크 위험성 예측 시스템 개발

(Development of Prediction of Electric Arc Risk using Object Dection Model)

이규빈\*, 김승연\*, 안동혁\*\*

(Gyu-bin Lee, Seung-yeon Kim, Donghyeok An)

## 요약

전기에너지에 대한 높은 의존도 때문에 국내에서 발생하는 화재 중 전기화재가 상당한 비중을 차지한다. 국내에서 발생하는 전기화재 4건 중 3건이 전선의 단락이나 접촉 불량에 의한 전기 아크에 의해 발생했다. 전기 아크란 절연체 사이에서 발생하는 전기적 전류의 방전 현상으로 순간적으로 상당한 열을 내뿜는다. 아크에 의한 전기 화재를 줄이기 위해서 본 연구에서는 전기 아크 위험성 예측을 목표로 한다. 아크 감지기에서 아크 데이터를 수집하고 시간순대로의 아크 데이터를 기반으로 그래프로 변환하였다. 머신 러닝의 데이터 학습에 서로 다른 시계열 데이터의 수로 변환한 그래프들을 사용하였다. 생성된 학습 모델의 성능을 측정하기 위해서 테스트 데이터를 기반으로 평가를 진행하였다. 결과에서 예측 시 사용하는 시계열 아크 데이터의 수가 20개일 때 예측률이 86%로 우수함을 확인하였다.

■ 중심어 : 머신러닝 ; 전기 아크 ; 딥러닝 ; 예측 시스템

## Abstract

Due to the high dependence on electric energy, electric fires make up a significant portion of fires in Korea. Electric arcs by short circuits or poor contact cause three of four electrical fires. An electric arc is a discharge phenomenon of electrical current between the insulators, which instantaneously produces high temperature. In order to reduce the fire due to electric arc, this study aims to predict the electric arc risk. We collected arc data from the arc detectors and converted into graphs based on temporal arc data. We used machine learning for training converted graph with different number of temporal arc data. To measure the performance of the learning model, we use the test data. In the results, when the number of temporal arc data was 20, the prediction rate was high as 86%.

■ keywords : machine learning ; electric arc ; deep learning ; prediction system

## I. 서론

현대 사회에서 전기에너지는 필수적인 구성 요소로서 목적에 따라 여러 분야에서 다양한 형태의 에너지로 변환되어 사용된다. 이를 통해 전기에너지가 육체노동을 대체함으로써 현대인들은 많은 편리함을 누리고 있으나, 전기화재와 같은 전기에너지의 위험성도 공존한다. 그림 1은 2011년부터 2015년까지의 총 화재 발생 수와 전기화재 발생 수를 나타낸다 [1]. 5년 동안 전기로 인해 발생한 화재가 약 17%에서 21%를 차지하기 때문에 전기화재 예방이 필요하다. 그림 2에서 전기화재의 원인을 보여주고 있듯이 아크에 의한 전기화재가 대부분을 차지한다[1].

아크는 전류가 급격하게 흘러 전기적 방전 현상이 발생하기 때문에 수 천도의 불꽃이나 스파크가 발생하는 것을 의미하며, 아크는 전기 제품의 접촉 불량, 연결 결함, 전선의 노후화 등에 의해서 주로 발생한다. 2019년 4월 강원도 고성, 속초 산불의 사례에서 볼 수 있듯이 아크는 온도가 높고 연소속도가 빠르기 때문에 전기화재를 줄이기 위해서 아크 발생의 위험성을 예측하는 것이 필요하다.

대부분의 전기 배선에 설치하는 누전 차단기는 감전 사고에 대한 감전 예비용 안전장치로 아크에 대한 감지 기능이 없기 때문에 아크로 인한 사고에 대처할 수 없다. 이로 인해 미국에서는 2002년부터 아크 차단기의 법적 설치를 의무화하여 전기화재 발생 비율을 절반 이상으로 감소시켰다. 국내에서도 2016년

\* 학생회원, 창원대학교 컴퓨터공학과 학부생

\*\* 정회원, 창원대학교 컴퓨터공학과 조교수

이 성과는 2018년도 과학기술정보통신부의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2018R1C1B6008187).

접수일자 : 2019년 11월 18일

게재확정일 : 2020년 03월 04일

수정일자 : 1차 2020년 01월 24일, 2차 2020년 02월 24일

교신저자 : 안동혁 e-mail : donghyeokan@changwon.ac.kr

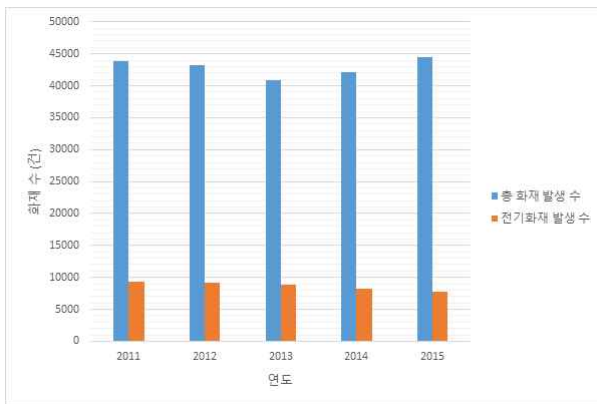


그림 1. 전기화재 통계

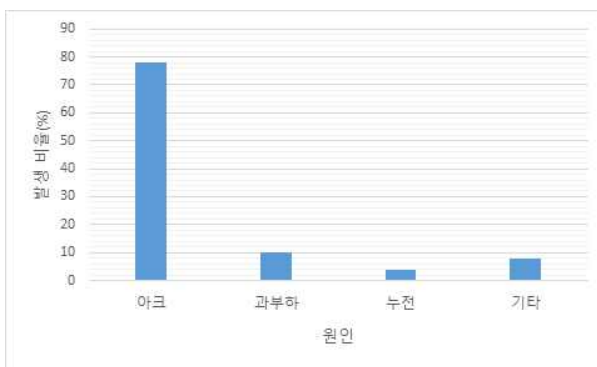


그림 2. 전기화재 원인

6월 30일 제정된 국가 건설기준과 2016년 개정 고시된 한국산업표준(KSC IEC)에 따라 고층건물이나 재래시장과 같이 화재가 급속도로 확대되기 쉬운 장소에 아크 차단기의 설치를 권장하고 있다.

본 논문에서는 아크로 인한 전기화재의 위험을 줄이기 위해서 아크 위험성 예측을 목표로 한다. 이를 위해서 거제 및 제주도에 있는 재래시장에 설치된 아크 감지기로부터 수집되는 데이터를 기반으로 머신 러닝을 수행하여 아크 위험성을 예측하는 시스템을 개발하고자 한다. 주기적으로 수집되는 아크 데이터는 수치에 따라 안전(Safe), 경고(Warning), 위험(Danger)이라는 세 가지 위험도로 분류된다. 일정 기간 수집된 아크 데이터들의 변동을 통해 다음 아크 데이터의 위험도를 예측한다. 이를 위해 수집된 아크 데이터를 시간에 따른 그래프로 변환해 데이터를 시각화한다. 시각화한 데이터는 학습에 알맞은 형태로 변환한 후, 머신 러닝을 활용하여 다음 아크 데이터의 위험도를 예측한다[2]. 예측에 사용하는 시계열 아크 데이터의 수를 10개, 15개, 20개로 구분하여 학습하여 아크 데이터의 위험도 예측 결과를 측정하였으며, 시계열 아크 데이터의 수가 20개일 때 높은 예측이 가능하였다.

본 논문의 구성은 다음과 같다. 2절에서는 머신러닝과 Faster

R-CNN에 대한 배경지식과 아크 탐지에 대한 관련 연구를 간략하게 소개한다. 3절에서 전기 아크 예측 시스템을 설계하고 구현한다. 4절에서는 구현한 전기 아크 예측 시스템의 실험 환경을 소개하고 성능 검증 결과를 논의한다. 마지막으로 5절에서는 결론을 제시한다.

## II. 배경지식 및 관련연구

인공 지능은 컴퓨터가 사람과 유사한 지능을 가지게 하는 기술이다[2]. 인공 지능은 크게 두 가지 방법으로 분류된다. 첫 번째는 지식 기반 시스템으로 사람이 가진 지식을 컴퓨터가 이해할 수 있는 형태로 구현한다. 두 번째인 데이터 기반 시스템은 지식이 포함된 데이터로부터 컴퓨터가 스스로 필요한 지식 또는 정보를 학습한다. 머신 러닝은 데이터 기반 시스템에서 컴퓨터가 학습할 수 있도록 하는 알고리즘과 기술을 개발하는 분야이다[2].

머신러닝의 종류로는 지도 학습(Supervised Learning), 비지도 학습(Unsupervised Learning), 반지도 학습(Semisupervised learning), 강화 학습(Reinforcement Learning) 등이 있다 [3]. 지도 학습은 레이블이 정의된 데이터를 기반으로 학습하기 때문에, 정확도가 높은 데이터를 사용할 수 있다. 비지도 학습은 레이블이 정의되어 있지 않은 데이터를 학습한다. 반지도 학습은 레이블이 정의되어 있지 않은 다수의 데이터와 레이블이 정의된 데이터를 기반으로 학습한다. 강화 학습은 현재의 상태에서 최적의 행동이 무엇인지를 학습한다.

최근에는 머신러닝을 수행할 수 있는 컴퓨팅 성능이 높아짐에 따라 다양한 분야에서 머신러닝을 활용하고 있다[4-6]. 머신러닝은 학습에 사용되는 데이터의 양이 증가할수록 성능이 높아지는 반면, 데이터 정보가 복잡해질수록 정교한 학습 모델을 필요해 성능이 저하된다. 이를 해결하기 위한 학습 모델이 딥러닝이다[2]. 특히 객체 인식 분야에서 활용도가 매우 높다. 그림 3에서 나타나고 있는 Faster R-CNN 알고리즘은 마이크로소프트에서 제안한 객체 인식(Object detection) 모델이다[7]. 이전 알고리즘인 Fast R-CNN 알고리즘의 문제점인 Region Proposal의 병목 현상을 Region Proposal Network(RPN)를 도입하여 보완했다[8]. 딥 컨볼루션 네트워크(Deep Convolution Network)인 Region Proposal Network(RPN)를 도입하여 각 위치의 Object Bounds와 Objectness Score를 동시에 예측한다. Faster R-CNN은 RPN과 Proposed Regions를 사용해 객체를 감지하는 두 개의 네트워크로 구성된다.

김두현 등은 아크 발생 시 전압과 전류들에 대한 데이터 측정과 분석, 저장이 가능한 모니터링 프로그램을 제안하고 개발하였다[9]. 서현욱 등은 주로 사용되는 반파 정류기, 전파 정류기, 벽 컨버터, 인버터 네 개의 회로 환경에서 발생한 아크에 대해서

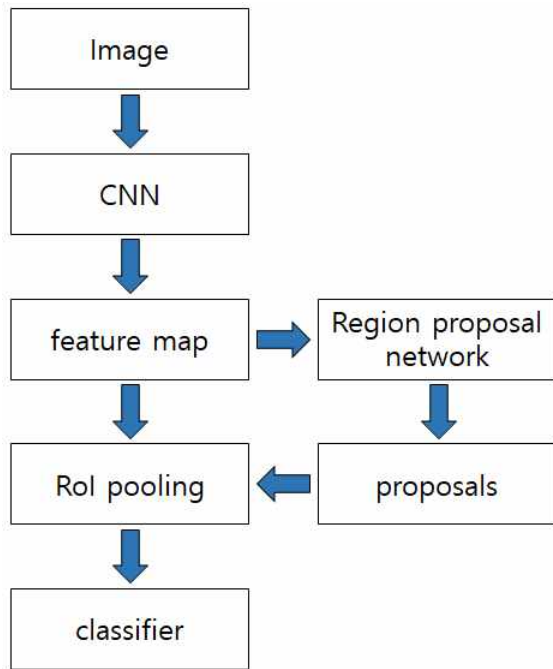


그림 3. Faster R-CNN Algorithm

전압 및 전류의 불규칙 파형을 검토하였다[10]. 그 결과 오동작의 원인을 탐지하여 출력단 커패시터의 조절 방안을 제시하고 검증하였다. 김두현 등은 아크 발생 시 열적 특성과 아크 전력을 분석하였다[11]. 하지만 위의 연구들은 아크 발생 시점을 기준으로 연구를 진행하였으나, 본 연구에서는 아크의 발생 가능성을 기준으로 위험도 예측을 목표로 한다는 점에 차별화된다.

### III. 전기 아크 예측 시스템 설계 및 구현

전기 아크 예측 시스템은 그림 4에서 보여주는 순서대로 동작하도록 구성한다. 그림 5는 수집된 아크 데이터의 예를 나타내고 있다. 각 라인마다 데이터가 수집된 시간과 해당 시간에 수집된 데이터를 나타낸다. 객체 탐지 모델을 사용하기 위해서는 해당 데이터를 이미지로 변환해야 하기 때문에, 수집된 데이터를 기반으로 시간을 x축으로 값을 y축으로 표현하는 라인 그래프를 생성한다. 이 때 다수의 데이터를 사용하여 그래프를 생성하고, 이후 데이터를 기준으로 안전(Safe), 경고(Warning), 위험(Danger)으로 레이블을 생성한다. 하지만 수집한 아크 데이터를 전처리 없이 그래프로 변환할 경우 두 가지 문제가 발생한다. 첫째 아크 데이터 값이 0으로 수집되는 경우, 안전으로 판단하지만, 아크 데이터 값이 0부터 시작하여 그래프를 작성할 수 없다. 따라서 학습을 위한 그래프를 생성하기 위해서 아크 데이터가 0인 경우 0부터 0에 근접한 일정 수의 범위 내에서 균일 분포(Uniform distribution)에 따라 랜덤한 데이터로 변환한다. 둘째 위험을 의미하는 아크 데이터의 상한 값이 매우 크기 때문에



그림 4. 시스템 흐름도

2019/03/12 오후 4:07:37.818425000	102
2019/03/12 오후 4:07:59.802680000	82
2019/03/12 오후 4:08:02.245430000	77
2019/03/12 오후 4:08:26.632471000	89
2019/03/12 오후 4:08:51.045826000	73
2019/03/12 오후 4:09:15.457422000	83
2019/03/12 오후 4:09:39.854251000	90

그림 5. 수집된 아크 데이터

안전 데이터와 경고 데이터의 값이 상대적으로 작아지는 문제가 발생한다. 이를 해결하기 하기 위해서 설정값을 초과하는 아크 데이터는 설정값에 근접한 임의의 데이터로 변환한다. 본 연구에서는 설정값을 150으로 정한다.

전처리 이후, 시간 순서대로 정렬된 데이터를 기반으로 라인 그래프를 생성한다. 이때, 그래프 생성에 사용하는 데이터의 수는 10개, 15개, 20개로 구분하여 생성한다. 그림 6은 15개의 데이터를 기반으로 생성한 그래프 중 하나이며, 다음에 수집될 아크 데이터가 위험(Danger)에 해당한다는 것을 의미한다.

생성된 그래프를 학습하기 위해 그림 7과 같은 XML 파일이 필요하기 때문에 LabelImg를 이용하여 XML 파일을 생성한다 [12]. XML 파일에는 해당하는 이미지 파일에 대한 라벨링

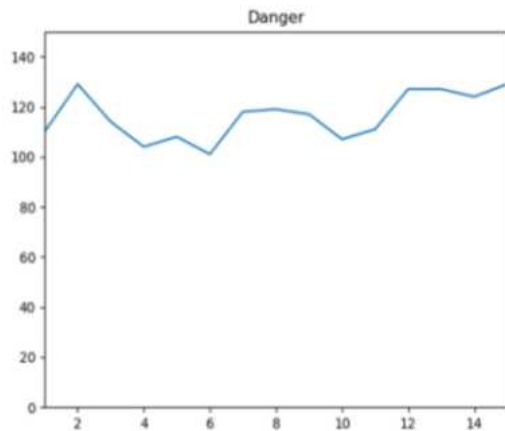


그림 6. 생성된 그래프



그림 7. XML 파일

정보나 이미지 범위 등이 저장된다. LabelImg로 생성된 XML 파일 코드의 특징을 분석하여 XML 코드를 자동 생성하는 프로그램을 개발한다. 프로그램이 동작하면서 그래프 이미지에 대한 각각의 XML 파일을 생성한다.

생성된 그래프 이미지와 XML 파일을 학습시키기 위해 학습 데이터와 테스트 데이터로 구분한다. 학습 데이터와 검증 데이터, 테스트 데이터의 비율을 3:1:1로 구성한다.

이전 단계에서 저장한 csv 파일 정보를 이용하여 TFRecord 파일을 생성한다. TFRecord 파일 생성 시 레이블 정보를 지정하여 생성한다. 이 단계에서 Tensorflow Object Detection API를 활용한다. Tensorflow Object Detection API는 Tensorflow 위에 구축된 프레임워크로 Object Detection Model을 쉽게 구성하고 학습 및 개발을 지원한다[13].

데이터 학습 시 많은 양의 데이터를 처리하기 위해 Training Pipeline을 구성한다. Training Pipeline은 학습 모델과 어떤 매개변수를 사용할지를 결정한다. 진기 아크는 짧은 시간 안에 발생하기 때문에 예측도 빠르게 수행되어야 하므로 본 논문에서 사용하는 학습모델은 Faster\_R-CNN\_Inception\_v2\_coco이다[14]. 매개변수로 사용하는 모델의 학습 횟수, TFRecord를 포함한 각종 파일 경로와 테스트 파일 개수 등을 지정한다. 학습에 사용한 데이터의 수는 200개이다. 배치 사이즈는 1이며 드롭아웃 레이트(dropout rate)는 1이다. 학습률(learning rate)는 0.0002이다. 그래프 생성 시 데이터의 수가 10개와 15개의 training epoch는 1000번이고 20개의 데이터로 그래프를 생성하였을 때의 training epoch는 500번이다. 손실 함수(loss function) 결과가 최소로 유지되고 검증 데이터의 결과에서 예측 정확도가 높게 유지되도록 training epoch 횟수를 결정하였다. 손실 함수는 아래와 같이 Faster R-CNN 모델에서 제안한 손실 함수를 사용한다[7].

$$L(p_i, t_i) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} p_i^* L_{reg}(t_i, t_i^*) \quad (1)$$

$p_i$ 는 앵커의 객체 인식 예측 확률이며,  $t_i$ 는 예측된 바운딩 박스(Bounding box)의 좌표이다.  $p_i^*$ 는 앵커가 양수이면 1이고 음수이면 0인 값이다.  $t_i^*$ 는 양수 앵커에 대한 ground box의 좌표 벡터이다.  $L_{cls}$ 와  $L_{reg}$ 는 classification loss와 regression loss를 의미한다.  $N_{cls}$ ,  $N_{reg}$ ,  $\lambda$ 는 각각 256, 2400, 10으로 지정한다.

데이터 학습을 진행하여 생성된 결과를 바탕으로 Frozen Inference Graph를 생성한다. Inference Graph는 ph 파일 형태로써 개체를 탐지하기 위한 분류기가 포함된다. 그래프 이미지를 입력하면 Inference Graph를 바탕으로 학습된 결과를 탐지한다. 데이터의 수가 20개일 때의 안전, 경고, 위험 예측 결과는 그림 8, 그림 9, 그림 10과 같다. 그래프의 x축은 시간순이며 y 축은 아크 데이터 값이다. 20개의 아크 데이터 값으로 그래프가 그려지며 이 그래프를 바탕으로 아크 위험을 예측한다. 그림 8에서는 두 개의 박스로 태깅 된 결과를 확인할 수 있다. 하나의 박스는 하나의 패턴 기반의 예측값을 의미하기 때문에, 그래프에서는 두 가지 패턴을 인식하여 두 개의 예측값을 보여준다는 것을 의미한다. 결과에서 두 개의 예측값들이 각각 99%와 98%의 확률로 안전(Safe)이라고 다음 아크 데이터의 위험도를 예측하기 때문에 다음 아크 데이터를 안전(Safe)로 예측한다. 위에서 언급한 순서대로 데이터를 학습하고 테스트할 수 있도록 시스템 환경을 표 1과 같이 구성한다. 머신러닝 개발을 위한 라이브러리로써 Tensorflow를 사용하였다[15].

Tensorflow는 구글에서 개발하여 2015년에 오픈소스로 공개한 머신러닝 라이브러리이다. 딥러닝과 머신러닝 분야를 일반인들도 사용하기 쉽도록 다양한 기능을 제공한다. Python, C++ 등 언어를 지원하고 CPU/GPU 모드로 동작하며 본 연구에서는 GPU 모드로 활용한다.

표 1. 구현 환경

개발 및 학습 환경	
OS	Window10x64 / Desktop
GPU	GTX 1080 Ti * 2
Tools	Anaconda3 5.2.0
Language	Python 3.6
Tensorflow	Tensorflow-gpu 1.9.0
CUDA	cudaToolkit 10.0.130
cuDNN	cuDNN 7.3.1
Train Model	faster_rcnn_inception_v2_coco

Tensorflow를 윈도우 환경에서 사용하기 위해 개발 도구로 Anaconda를 사용하였다[16]. Anaconda는 윈도우, 리눅스, 맥 (Mac) OS와 같은 개발 환경에서 Python, R을 이용한 데이터 과학과 머신러닝을 손쉽게 사용할 수 있도록 도와주는 오픈소스 소프트웨어 배포판이다. Anaconda를 사용함으로써 다양한 라이브러리들 간에 종속성 문제를 해결할 수 있다. 본 연구에서는 머신러닝을 사용하여 개발 및 학습을 진행하기 위해 Tensorflow Object Detection API와 Tensorflow Detection Model을 활용하였다. 앞에서 언급한 대로 Tensorflow Object Detection API는 이미지의 객체 인식을 위해서 활용하였다. Tensorflow Detection Model은 구글에서 Tensorflow Object Detection API를 활용하여 머신러닝을 사용하기 위한 학습 모델을 제공한다. 데이터로부터 Object Detection을 수행하여 Boxes 및 Mask 결과를 추론하는 다양한 Common Object in Context (COCO) 모델을 제공한다[17].

#### IV. 성능 평가

본 연구에서 제안하는 시스템의 성능 평가를 수행하였다. 제안하는 예측 시스템의 예측 결과와 실제 결과의 예측 능력을 측정하였다. 그림 7에서 확인할 수 있듯이 제안하는 예측 시스템은 최대 두 개의 결과를 예측하기 때문에, 두 개의 예측 결과 중 하나라도 실제 결과와 일치할 때와 두 개의 예측 결과가 모두 실제 결과와 일치할 때의 예측률 측정을 수행하였다. 먼저 그림 11은 하나의 예측 결과가 맞을 때의 예측률을 나타낸다. 예측 시 사용하는 데이터의 수가 10개, 15개, 20개일 때, 90%, 98%, 94%의 예측률을 보여준다. 데이터의 수가 15개일 때가

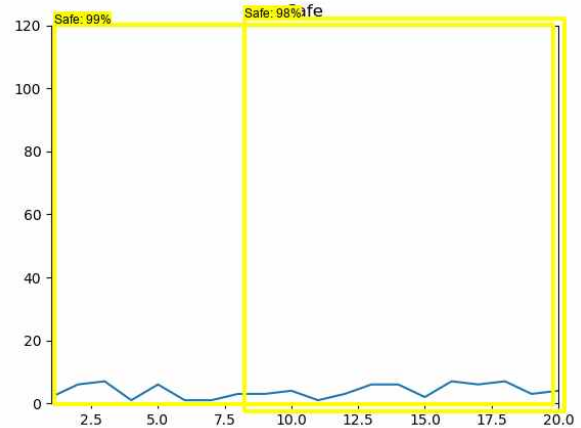


그림 8. 데이터 20개 기반 그래프의 안전 예측

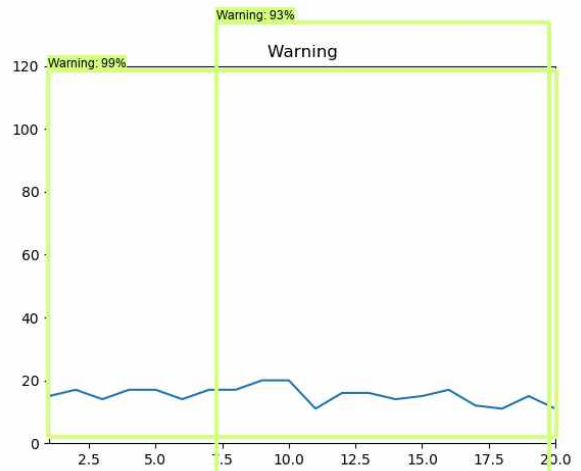


그림 9. 데이터 20개 기반 그래프의 경고 예측

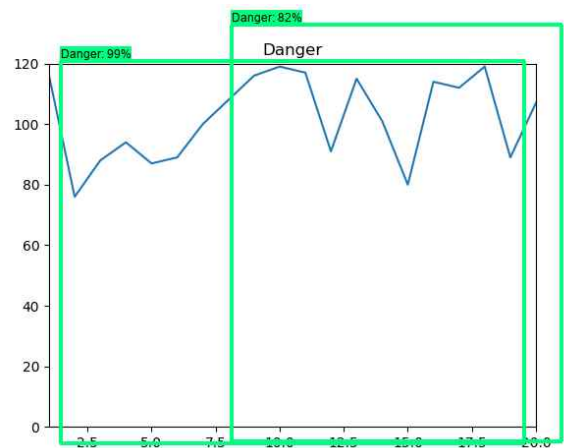


그림 10. 데이터 20개 기반 그래프의 위험 예측

가장 높은 예측률을 측정되었으나 세 경우가 모두 예측률이 90% 이상으로 우수한 예측률을 보인다. 두 번째, 두 개의 예측 결과가 실제 결과와 일치할 때만의 예측률을 측정하였고, 결과는

그림 12에서 나타난다. 예측 시 사용하는 데이터의 수가 20개일 때가 86%의 예측률로 가장 좋은 성능 보인다. 최소 하나의 예측 결과가 일치할 때 가장 좋은 성능을 보였던 데이터의 수가 15개의 예측 결과가 24%로 매우 낮은 성능으로 평가된다. 두 실험의 평가 결과에서 예측 시 사용하는 데이터의 수가 20개일 때 아크 위험성 예측에 가장 적합하다고 판단할 수 있다. 일반적으로 안전에서 경고, 경고에서 위험 등 데이터 값의 차이가 크지 않게 입력될 때에는 준수한 예측 능력을 보였지만, 안전에서 위험으로 갑자기 데이터 값의 차이가 크게 입력될 때에는 저조한 예측 능력을 보였다.

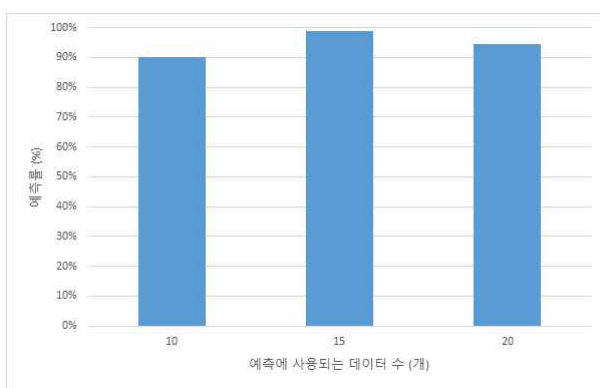


그림 11. 최소 하나의 결과 일치 시 아크 위험성 예측률

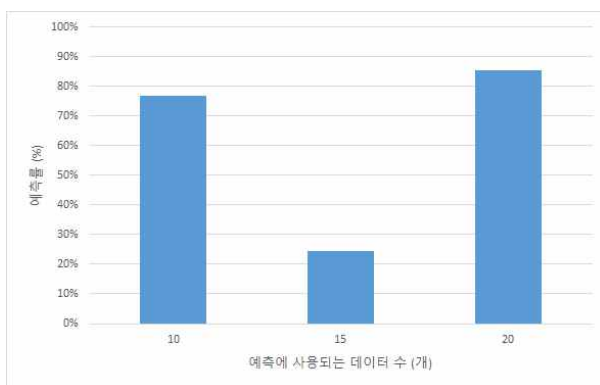


그림 12. 두 개의 결과 일치 시 아크 위험성 예측률

## V. 결론

본 논문에서는 수집한 아크 데이터를 바탕으로 Faster R-CNN과 Tensorflow Object Detection을 사용하여 Anaconda 가상환경에서 실험하였다. 아크 데이터는 안전, 경고, 위험으로 분류하여 학습 및 테스트를 위해 그래프를 순차적인 누적시간을 기준으로 생성하였다. 누적시간 데이터 20개를

기준으로 학습된 데이터로 예측률을 평가하였을 때 약 86% 정도의 예측 능력을 보여주었다. 추가적인 예측률 향상을 위해서 다른 학습 모델의 개발을 추후 연구에서 수행할 계획이다.

## REFERENCES

- [1] 한국전기안전공사, 산업통상자원부, “전기재해 통계 분석. 제25호,” 한국전기안전공사, 2016년
- [2] 김인중, “기계 학습의 발전 동향, 산업화 사례 및 활성화 정책 방향: 딥러닝 기술을 중심으로,” *SPRi Issue Report*, 2016년
- [3] 배재환, “AI 머신러닝을 이용한 자동차 고장진단 시스템 설계 및 개발,” *차세대컨버전스정보서비스 기술논문지*, 제27권, 제2호, 141-148쪽, 2018년
- [4] 김민구, 김진수, 반성범, “사용자 인식을 위한 가상 심전도 신호 생성 기술에 관한 연구,” *스마트미디어저널*, 제8권, 제4호, 33-37쪽, 2019년 12월
- [5] 이현영, 강승식, “음절 단위 임베딩과 딥러닝 기법을 이용한 복합명사 분해,” *스마트미디어저널*, 제8권, 제2호, 74-79쪽, 2019년 6월
- [6] 오정원, 김행곤, 김일태, “머신러닝 적용 과일 수확 시기 예측시스템 설계 및 구현,” *스마트미디어저널*, 제8권, 제1호, 74-81쪽, 2019년 3월
- [7] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, June 2015.
- [8] R. Girshick, “Fast R-CNN,” *IEEE Conf. of Computer Vision (ICCV)*, pp. 1440-1448, Santiago, Chile, Dec. 2015.
- [9] 박종영, 김두현, 김성철, 황동규, 김윤복, “직렬아크 패턴 분석을 위한 모니터링 프로그램 개발,” *대한전기학회 전기설비전문위원회 춘계학술대회 논문집*, 108-110쪽, 2010년
- [10] 서현욱, 최규하, “아크에 의한 전력변환장치의 전기적 영향 분석,” *조명·전기설비학회논문지*, 제27권, 제7호, 8-16쪽, 2013년
- [11] 김두현, 황동규, “전선의 접속부에서 직렬아크에 의한 전기적 및 열적 신호 분석,” *한국안전학회지*, 제30권, 제5호, 1-7쪽, 2015년
- [12] LabelImg, <https://github.com/tzutalin/labelImg/> (accessed Nov. 15, 2019).
- [13] Github Tensorflow Object Detection API, [https://github.com/tensorflow/models/tree/master/research/object\\_detection](https://github.com/tensorflow/models/tree/master/research/object_detection) (accessed Nov. 15, 2019).

- [14] faster\_rcnn\_inception\_v2\_coco, [https://github.com/tensorflow/models/blob/master/research/object\\_detection/samples/configs/faster\\_rcnn\\_inception\\_v2\\_coco.config](https://github.com/tensorflow/models/blob/master/research/object_detection/samples/configs/faster_rcnn_inception_v2_coco.config) (accessed Nov. 15, 2019).
- [15] TensorFlow, <https://www.tensorflow.org/?hl=ko> (accessed Nov. 15, 2019).
- [16] Anaconda, <https://www.anaconda.com/> (accessed Nov. 15, 2019).
- [17] Github Tensorflow detection model zoo, [https://github.com/tensorflow/models/blob/master/research/object\\_detection/g3doc/detection\\_model\\_zoo.md](https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md) (accessed Nov. 15, 2019).

---

저 자 소 개

---



이규빈 (학생회원)

2020년 창원대학교 컴퓨터공학과 학사 졸업.

<주관심분야 : 인공지능, 빅데이터>



김승연 (학생회원)

2020년 창원대학교 컴퓨터공학과 학사 졸업.

<주관심분야 : 응용 소프트웨어, 인공지능, 빅데이터>



안동혁 (정회원)

2006년 한동대학교 전산전자공학부 학사 졸업.

2013년 KAIST 전산학과 박사 졸업.

2013년 성균관대학교 박사후연구원.

2014년 삼성전자 책임연구원.

2015년 계명대학교 컴퓨터공학과 조교수.

2017년 창원대학교 컴퓨터공학과 조교수.

<주관심분야 : 컴퓨터네트워크, 사물인터넷, 스마트 팩토리>