

Deep-Learning Based Real-time Fire Detection Using Object Tracking Algorithm

Jonghyuk Park*, Dohyun Park*, Donghwan Hyun*, Youmin Na*, Soo-Hong Lee*

*Student, Dept. of Mechanical Engineering, Yonsei University, Seoul, Korea

*Student, Dept. of Mechanical Engineering, Yonsei University, Seoul, Korea

*Student, Dept. of Mechanical Engineering, Yonsei University, Seoul, Korea

*Student, Dept. of Mechanical Engineering, Yonsei University, Seoul, Korea

*Professor, Dept. of Mechanical Engineering, Yonsei University, Seoul, Korea

[Abstract]

In this paper, we propose a fire detection system based on CCTV images using an object tracking technology with YOLOv4 model capable of real-time object detection and a DeepSORT algorithm. The fire detection model was learned from 10800 pieces of learning data and verified through 1,000 separate test sets. Subsequently, the fire detection rate in a single image and fire detection maintenance performance in the image were increased by tracking the detected fire area through the DeepSORT algorithm. It is verified that a fire detection rate for one frame in video data or single image could be detected in real time within 0.1 second. In this paper, our AI fire detection system is more stable and faster than the existing fire accident detection system.

▶ **Key words:** Fire Detection, DeepSORT, Tracking Algorithm, Real-time, YOLOv4

[요 약]

본 논문에서는 실시간 객체 탐지(Real-time Object Detection)가 가능한 YOLOv4 모델과 DeepSORT 알고리즘을 활용한 객체 추적(Object Tracking) 기술을 활용하여 CCTV 영상 이미지 기반의 화재 탐지 시스템을 제안한다. 화재 탐지 모델은 10800장의 학습용 데이터로부터 학습되었으며 1000장의 별도 테스트 셋을 통해 검증되었다. 이후 DeepSORT 알고리즘을 통해 탐지된 화재 영역을 추적하여 단일 이미지 내의 화재 탐지율과 영상 내에서의 화재 탐지 유지성능을 증가시켰다. 영상 내의 한 프레임 혹은 단일 이미지에 대한 화재 탐지 속도는 장당 0.1초 이내로 실시간 탐지가 가능함을 확인하였으며 본 논문의 AI 화재 탐지 시스템은 기존의 화재 사고 탐지 시스템 보다 안정적이고 빠른 성능을 지니고 있어 화재현장에 적용 시 화재를 조기 발견하여 빠른 대처 및 발화단계에서의 진화가 가능할 것으로 예상된다.

▶ **주제어:** 화재 탐지, DeepSORT 알고리즘, 객체 추적, 실시간 탐지, YOLOv4

- First Author: Jonghyuk Park, Corresponding Author: Soo-Hong Lee
- *Jonghyuk Park (hyukize@yonsei.ac.kr), Dept. of Mechanical Engineering, Yonsei University
- *Dohyun Park (pdh7183@yonsei.ac.kr), Dept. of Mechanical Engineering, Yonsei University
- *Donghwan Hyun (donghwan9411@gmail.com), Dept. of Mechanical Engineering, Yonsei University
- *Youmin Na (umuna@yonsei.ac.kr), Dept. of Mechanical Engineering, Yonsei University
- *Soo-Hong Lee (shlee@yonsei.ac.kr), Dept. of Mechanical Engineering, Yonsei University
- Received: 2021. 11. 16, Revised: 2022. 01. 14, Accepted: 2022. 01. 17.

I. Introduction

화재 사고는 첫 발화로부터 이른 시간 내에 넓은 범위로 번져 수많은 인명 피해를 초래할 수 있어 조기에 이를 감지하여 신속하게 대응하는 것이 중요하다. 대부분의 실내 환경에는 화재경보기가 설치되어 화재로부터 발생하는 연기와 열 등을 감지하여 이를 조기에 발견할 수 있다. 하지만, 실외 환경 및 천장이 높은 실내구조에서는 화재의 규모가 상당히 커진 후에야 이를 감지, 발견할 수 있다. 이를 해결하기 위해 모든 실외 공간에 화재경보기를 설치하는 것은 그 실질적 효용성이 떨어질뿐더러 상당한 비용적 문제를 포함하게 된다. 따라서 기존에 설치되어있는 시설과 설비를 활용하여 원거리에서도 화재를 탐지한다면 실외 환경 등에서도 최소한의 비용으로 넓은 범위의 화재를 감지할 수 있기에 CCTV 영상 이미지를 활용한 화재 탐지 기법이 최근까지 활발히 연구되고 있다. 하지만 기존의 화재 탐지 기법들은 연산량이 적은 가벼운 모델의 경우 초기 화재에 대한 탐지 정확도가 지나치게 낮으며, 무거운 모델의 경우 높은 지연율을 보유하고 있어 실시간 탐지(Real-time Detection)가 불가능하였다. 특히 화재를 탐지하는 과정 중간에서 불씨가 흔들리면서 화재를 안정적으로 탐지하지 못하고, 해당 모델의 탐지 결과로서 이미지 내에 화재 영역을 표시하는 상자(Bounding Boxes)가 깜빡이는 현상(Flickering Effect)이 존재하였다. 이는 화재의 형태가 일정하지 않기에 발생하는 현상으로 이를 해결하기 위해 동적 물체 추적 알고리즘(Object Tracking Algorithm of Moving Object)을 적용할 필요가 있다.

본 논문에서는 딥러닝 기반의 실시간 객체 탐지 모델인 YOLOv4[1]와 객체 추적 기법인 DeepSORT[2] 알고리즘을 적용하여 실시간 화재 탐지 모델을 구현하였다. 실시간

화재 탐지를 위해서 약 10800장의 화재 이미지를 수집하여 YOLOv4 모델을 학습하였고 해당 모델의 탐지 결과를 바탕으로 DeepSORT 알고리즘을 통해 화재 탐지 및 추적 모델을 완성하였다.

II. Preliminaries

1. YOLOv4

YOLO는 현시점에서 가장 많이 쓰이는 딥러닝 기반 실시간 객체 탐지 모델(Deep-Learning Based Real-Time Object Detection Model)이다. “You Only Look Once”의 약자로 줄여서 흔히 YOLO라고 부른다. YOLO가 등장하기 이전에도 DPM(Deformable Part Model)[3]과 R-CNN(Regions with Convolutional Neural Network)[4]같은 딥러닝 모델을 이용하여 객체 탐지를 수행하는 방법이 존재해왔으나 YOLO는 기존 모델과 지향하는 바가 다르다. YOLO는 더 높은 정확도를 추구하는 것이 아닌, 근접한 정확도를 가지면서 더 많은 양의 이미지를 처리할 수 있는 실시간 객체 탐지를 목표로 한다. 기존의 R-CNN 모델 등의 경우에는 이미지를 일정한 규칙으로 여러 장으로 분할하고 CNN(Convolutional Neural Network)[5] 모델을 통과시키기 때문에 한 장의 이미지에서 객체 탐지를 수행해도 실제로는 수 천장의 이미지를 모델에 통과시킨다. 반면, YOLO는 이미지 데이터를 CNN 모델에 한 번만 투입 시키는 방식을 사용하여 기존 방식보다도 훨씬 더 빠른 처리속도를 갖는다는 장점이 있다. 이에 더하여, 다른 객체 탐지 모델의 경우 다양한 전처리 과정과 인공 신경망(Neural Network)을 결합해서 사용하지만, YOLO는 단 하나의 인공신경망에서 이를 전부 처리한다.

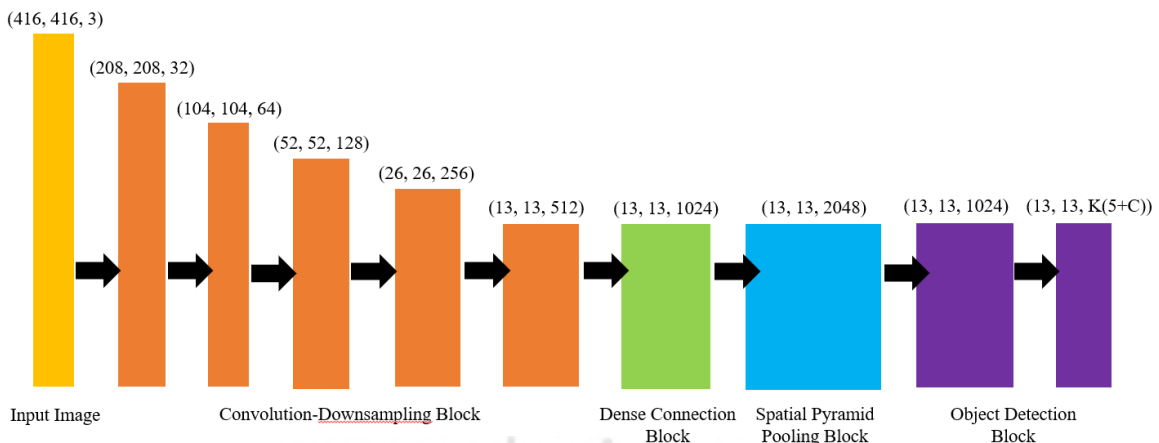


Fig. 1. YOLOv4 architecture

이러한 특성들에 기인하여 YOLO는 실시간 객체 탐지 (Real-time Object Detection) 분야에서 가장 널리 사용되는 인공 신경망 모델이 되었다. 실제로, 일반적으로 Fast R-CNN이 0.5 FPS(초당 프레임 수)의 성능을 보유하고 있는 데에 비해, YOLO의 경우 45 FPS의 성능을 갖는다.

2. DeepSORT Algorithm

2.1 Kalman Filter[6]

칼만 필터(Kalman Filter)는 재귀 필터의 한 종류로 노이즈를 포함하고 있는 데이터를 기반으로 선형 역학계의 상태를 추정한다. 이는 과거의 측정값을 바탕으로 현재의 상태 변수의 결합분포를 추정하는 방식으로 이루어진다. 칼만 필터의 알고리즘은 예측 단계, 업데이트 단계로 세분화할 수 있는데, 예측 단계에서는 현재 상태 변수에 대한 값을 예측하고 해당 예측의 정확도를 측정한다. 이후, 업데이트 단계에서는 실제 측정치와 예측한 값 사이의 차이를 반영하여 현재 상태 변수의 업데이트를 진행하게 된다.

2.2 Mahalanobis Distance[7]

마할로노비스 거리(Mahalanobis Distance)는 다변량 거리의 기본 형태로, 다변량의 데이터에서 분포의 형태를 고려하여 거리를 재는 척도이다. 아래에 제시된 Equation 1. 은 다변량의 데이터 u 와 v 의 마할로노비스 거리를 구하는 식이다. 이는, covariance matrix의 inverse matrix를 곱하여 거리를 재는 방식으로 구현되어 있기에 이를 통해 변수 간의 correlation과 같은 분포를 고려한 거리를 도출해 낼 수 있다.

$$d(u, v) = \sqrt{(u - v)\Sigma^{-1}(u - v)^T}$$

Equation 1. Mahalanobis Distance Between u, v

2.3 Assignment Problem & Hungarian Algorithm[8]

다수의 공급처와 수요처가 존재하는 상황에서 각자 다른 수송비용을 가지고 있다면, 총 수송비용의 합이 최소가 되는 최적 해를 찾는 문제를 할당 문제(Assignment Problem)라고 한다. 이때 한 공급처에서 반드시 한 수요처로만 수송이 이루어져야만 한다. 헝가리안 알고리즘(Hungarian algorithm)은 이러한 할당 문제에 대한 해결책으로 제시되었다.

2.4 SORT Algorithm

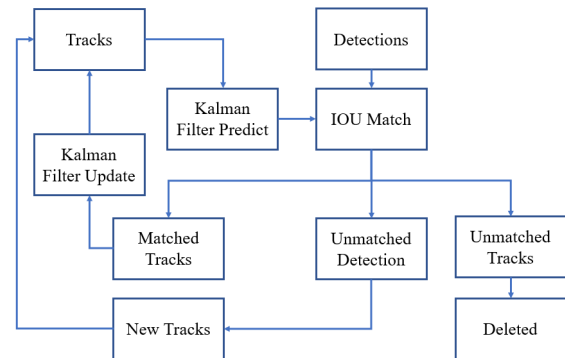


Fig. 2. Schematic flow diagram of SORT algorithm

SORT Algorithm은 실시간 추적을 위해 객체(Object) 들을 효율적으로 상호 연관시키는 MOT(Multi Object Tracking)[9] 기술이다. SORT는 Simple Online and Realtime Tracking의 약자로 여기서 나타나는 Online Tracking 방식은 미래 프레임에 대한 정보 없이 과거와 현재 프레임의 객체 탐지 정보만을 사용하여 연관 관계에 대한 추적(Tracking)을 수행하는 방식이다. Fig 2. 에 SORT 알고리즘의 흐름도가 제시되어 있다. 가장 먼저 Detections는 프레임에서 개체 탐지한 것을 나타내며 이 과정에서는 대부분 YOLO 모델을 사용하게 된다. Estimation에서는 칼만 필터를 통해 개체를 추적하기 위한 측정치를 예측하고 업데이트하는 과정이 진행된다. 칼만 필터는 Fig. 2.에서 확인할 수 있듯이 예측값과 실제 측정치를 통해 업데이트하며 다음 프레임의 값과 다시 IOU(Intersection over Union)[10] 값을 측정하는 재귀 필터의 형태를 보인다. Data Association은 다중 객체 추적(MOT) 방법을 기반으로 한 tracking-by-detection의 핵심 단계로, 할당에 관한 분기처리로 Hungarian algorithm을 사용하면서 IOU를 Metric으로 사용하여 Fig. 2.에서는 IOU Match라고 나타나 있다. 이 단계에서는 IOU 유사도를 구한 후, 추적되고 있던 개체와 아닌 개체를 분류한다. 추적 중인 개체는 칼만 필터를 통해 다음 개체를 추적하기 위한 측정치를 업데이트하게 된다. 하지만 SORT 알고리즘의 경우 실제 상황에서 발생하는 개체 중첩(Occlusion)이나 ID Switching에는 불안정한 특성을 보유하고 있다. 개체 중첩(Occlusion)은 개체가 어떤 상황에 의해 가려지는 현상을 일컫는 말로 해당 현상이 발생하면 중첩된 대상 사물의 Tracking을 기존과 이어서 할 수 없게 된다. 이 과정에서 이전에 가지고 있던 ID가 변경되고 해당 현상을 ID Switching이라고 한다. 따라서 이를

해결하기 위해서는 각각의 개체마다 특성을 파악해두고 중첩이 생길 때마다 해당 특성을 비교해가며 개체를 추적할 필요가 있다.

2.5 DeepSORT Algorithm

DeepSORT는 SORT를 확장한 개념으로, DeepSORT의 가장 큰 특징은 Deep Appearance Descriptor로 Re-identification(ReID)[11] 모델을 적용하여 ID Switching 문제를 해결했다는 점을 꼽을 수 있다. 또한, Matching Cascade[12] 로직을 사용하여 기존 SORT 알고리즘 대비 정확한 추적 성능을 보유하고 있다. Fig. 3.에서 확인할 수 있듯이 DeepSORT 알고리즘에서는 먼저 칼만 필터를 가지고 다음 프레임에 대해 연결되는 개체를 예측하고 해당 결과에 따라 Matching Cascade로 개체의 상태를 추출한다. Matching Cascade는 유사도 행렬 M을 계산하는 방법으로 유사도를 계산하기 위해 마할로노비스 거리와 코사인 거리(Cosine Distance)[13]의 가중 평균으로 비용 행렬(Cost Matrix)[14]을 구한다. 코사인 거리는 칼만 필터만으로 설명되지 않는 움직임을 위해 도입되었으며 이때, 할당 문제(Assignment Problem)를 위해 비용 함수를 입력값으로 하는 헝가리안 알고리즘을 사용한다.

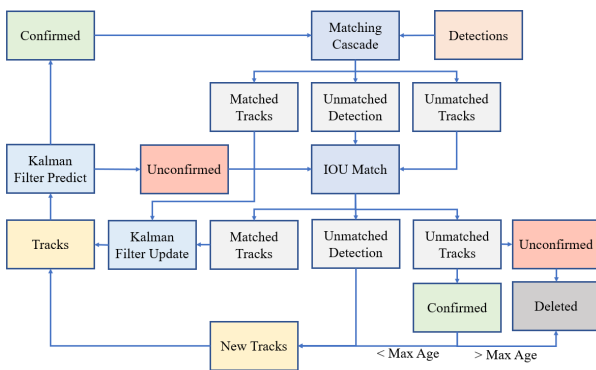


Fig. 3. Schematic flow diagram of DeepSORT algorithm

해당 과정을 통해 Unmatched Tracks, Unmatched Detections, Matched Tracks 상태 배열로 개체들을 나눈다. 각 상태에 대해 정의하면 다음과 같다.

- **Matched Tracks** : 계속해서 추적 중인 개체로, 계속해서 칼만 필터를 업데이트 한다.
- **Unmatched Detections** : 새롭게 등장한 개체로서 새로운 Track으로서 개체를 정의하지만 잠정적인(Tentative) 상태로 우선 구별하였다가 개체가 3번 등장할 때 최종 확인 상태(Confirmed)로 분류된다.

- **Unmatched Tracks** : 추적되던 개체를 발견하지 못했을 때의 상태로 개체 추적이 불가능한 상황에서 개체를 추적 리스트에서 바로 삭제하는 대신 개체가 다시 나타날 가능성을 고려하여 잠정적인 상태를 할당한 뒤 개체가 다시 나타날 때까지 대기한다.

Unmatched Tracks와 Unmatched Detections의 상태를 가진 개체들을 정확히 구분하기 위해 IOU Matching을 진행하여 개체들의 상태에 따라 분기 처리될 수 있도록 한다. 이는 갑작스러운 개체의 특징 변화(Appearance Change)에 대처하는 데 도움을 주는 동시에 부분적인 개체 중첩 문제를 해결할 수 있다. 지금까지 서술된 방법론에 기인하여 DeepSORT 알고리즘은 여러 장점을 보유하고 있다. 타 추적 알고리즘 대비 빠른 처리속도를 바탕으로 실시간 객체 탐지 및 추적(Real-time Object Detection & Tracking)에 적용할 수 있고 높은 추적 정확도를 보이며, SORT 알고리즘에 비해 ID Switching이 감소하였다. 반면 몇 가지 단점도 존재한다. SORT 알고리즘 대비 계산이 필요한 과정이 많아졌기에 CPU만으로는 처리하기에는 다소 큰 계산복잡도를 가지고 있어 GPU를 병행 사용하는 것이 일반적이다. 이에 따라 계산자원의 사용이 늘어남에 따라 처리 비용이 소폭 늘어나는 점을 예로 들 수 있다.

3. Real-time Detection

한국정보통신기술협회(TTA, Telecommunications Technology Association)의 정보통신용어사전에 따르면 실시간(實時間, real-time)이란, ‘실제로 컴퓨터에 관련된 일들이 발생하는 시간. 컴퓨터에서 관련되는 일들, 즉 프로세서나 통신이 발생하는 동안에 이에 관련되는 연산이 행해질 때, 그 컴퓨터는 실시간 조작이 되고 있다고 한다. 실시간 조작이 되는 컴퓨터는 연산 속도가 매우 빨라서 실제로 프로세서나 통신이 진행되고 있을 때 그것에 대한 연산 결과가 곧바로 실제로 발생하고 있는 프로세서나 통신에 영향을 미칠 수 있어야 된다.’ 라고 명시되어있다.

4. Research Trends

박경민 외 1인은 불꽃과 연기를 동시에 검출하며 이를 위해 딥러닝 알고리즘인 합성곱 신경망을 이용하여 선박 기관실에서 화재 영상을 녹화한 데이터로 실험을 수행하였다. 불꽃과 연기의 특징을 외곽 상자 추출한 후 합성곱 신경망 중 하나인 옴로(Tiny-YOLOv2)를 이용하여 학습하고 결과를 테스트하였다. 실험 결과를 검출률, 오검출률, 정확도로 평가하였다[15].

김정수 외 4인은 딥러닝 모델의 학습을 위한 화재 데이터 확보가 어렵고 열악한 영상 조건 및 화재로 오인할 수 있는 객체가 많은 지하공동구 내 화재에 대해 화재 탐지 모델을 제안하였다. 기존 합성곱 인공신경망에 GoogleNet의 Inception block과 ResNet의 skip connection을 조합하여 어두운 환경에서 발생하는 화재 탐지를 위한 모델 구조를 제안하였으며, 제안된 모델을 효과적으로 학습시키기 위한 방법도 함께 제시하고 모델의 화재 탐지 성능을 정량적으로 평가하였다[16].

유시영 외 1인은 네트워크 감시 카메라를 이용해 실시간 화재 감지를 위한 인공지능 시스템을 제안하였다. 화재를 감지하기 위해서 화염의 특징을 사용하여 화재 후보 영역을 인식하며 화재와 유사한 색을 갖는 물체를 구분하기 위해서 매우 빠른 화재 분류기를 개발하였다. 또한 화재를 감지하는데 단서가 될 수 있는 연기 감지 알고리즘도 제안하였다. 화재 및 연기 감지 알고리즘의 결과가 융합하여 최종결정을 한다[17].

위에서 언급한 접근은 연산량이 적은 가벼운 탐지 모델을 사용해 초기 화재에 대한 탐지 정확도가 낮거나, 연산량이 큰 무거운 탐지 모델을 사용해 높은 지연율을 보유하고 있어 실시간 탐지가 불가능하다. 또한, 화재를 탐지하는 과정 중간에서 불씨가 흔들리면서 화재를 안정적으로 탐지하지 못하는 현상을 개선시킬 동적 물체 추적 알고리즘이 적용되지 않았다.

III. Experiment

1. Dataset

공개 데이터로부터 수집한 화재 이미지 10800장을 사용하여 학습 데이터셋을 구성하였다. 모든 이미지 데이터셋에 대하여 Bounding Box Labeling을 진행하였으며 화재 영역이 구분이 불명확한 경우 연기를 제외한 실제 불꽃 부분만 포함하도록 Policy를 설정하였다. 학습(Train Set)과 검증 셋(Validation Set)의 비율은 8:2로 설정하였으며 Train Dataset의 경우 실제 학습에 투입되는 과정에서 Horizontal, Vertical Flip, 100~110%의 확대, 0~5%의 Sheer Transform, 0~45°의 회전을 통해 데이터 증강(Data Augmentation)을 진행했다. 해당 데이터로 학습된 모델의 테스트 셋은 학습 및 검증 셋과 별도로 구성하였으며 실제 CCTV 영상을 포함한 1000장의 이미지로 구성하였다. 또한, 추적 알고리즘 적용 전후 비교를 위해 1분 내외의 화재가 포함된 영상 10개를 테스트 셋에 추가하였다.

이에 더하여, 실시간 객체 탐지를 위한 FHD 웹캠을 스트리밍하여 해당 영상에 대해서도 실시간 객체 탐지 및 추적을 진행하였다. 웹캠의 경우 프린팅된 화재 이미지와 가스라이터의 불꽃을 테스트 화재 데이터로써 사용하였다. 실제 사용된 데이터 예시는 Fig. 4.-6.에서 확인할 수 있다.



Fig. 4. Fire detection dataset example



Fig. 5. Fire detection dataset example

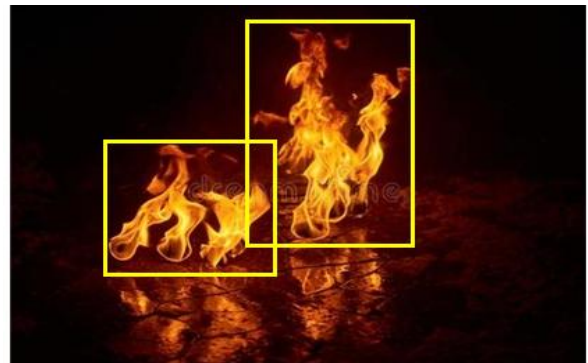


Fig. 6. Fire detection dataset example

CCTV에서 탐지하는 화재 상황의 경우 대부분 실내 화재 상황이다. 이러한 실내 화재 상황의 이미지만 사용하기에는 그 조건에 만족하는 이미지 데이터의 수가 너무 적어 학습이 제대로 진행되지 않는다. 그렇기 때문에 불꽃의 feature를 확실하게 학습시켜 탐지하도록 하는 것이 성능

향상에는 더 효과적이다. 하지만 인위적인 불꽃이나 실외의 불꽃의 경우 모델이 학습하여 실제 화재 상황이 아니라도 화재로 탐지할 가능성이 있다. 이는 검사 모델이 갖는 일종의 trade-off 문제이지만 화재 탐지의 경우 오류의 종류가 거짓 양성(False positive)일 때보다 거짓 음성(False negative)일 때가 더 위험한 task이기 때문에 화재를 미탐할 확률을 최대한 줄여야 한다. 모델 훈련 시 불꽃의 특징을 잘 학습하기 위해서 CCTV에서 잡힐 수 있는 실내 화재 상황 외의 이미지 데이터도 포함시켰다.

모든 화재 탐지 데이터는 실제 화재 이미지로 구성하였고, QVGA(320x240) 해상도부터 HD1080(1920x1080)의 다양한 해상도로 이루어져 있다. 또한 일부 데이터는 연기로 인해 전체적으로 선명하지 않다. 이런 여러 카메라 성능 및 주변 배경 상황을 모두 고려해 학습용 데이터와 테스트 데이터를 구축했기 때문에 특정 상황에 대한 성능 저하의 우려가 매우 낮다.

2. YOLOv4 Model Train Configuration

YOLOv4 모델을 학습하는 과정에서 사용된 Hyper parameters는 Table 1.에 정리되어 있다.

Table 1. Hyperparameters of YOLOv4 configuration

Hyper parameter	Value
Input Size	(608, 608)
Batch Size	16
Subdivisions	16
Initial Learning Rate	0.0013
Momentum	0.949
Decay	0.0005
Anchors	12, 16, 19, 36, 40, 28, 36, 75, 76, 55, 72, 146, 142, 110, 192, 243, 459, 401
IoU Threshold	0.213
Max Batches	10000

3. Results

YOLOv4 모델만을 사용하여 화재 이미지를 학습하였을 때는 0.68의 mAP 값을 얻을 수 있었다. 해당 결과는 이미지 단위의 성능 평가 결과로 연속된 프레임 기반의 비디오 데이터를 대상으로 평가했을 때는 Bounding Box의 깜빡임 현상(Flickering Effect)를 관찰할 수 있었다. 외부환경의 변화가 크지 않은 정적 영상의 경우 YOLOv4의 경우 10프레임 기준 1.8 프레임의 미탐지율을 보였고 YOLOv4 + DeepSORT 보강 모델에 대해서는 동일 프레임 수 대비 1.3프레임의 미탐지율을 보였다. 동적 환경 내에서의 영상 데이터의 경우 바람 등에 의한 화재의 형태 변화로 평균 10프레임 기준 2.3 프레임에 대한 탐지가 실패하였고 특

히 CCTV 영상과 같은 저화질 영상의 경우 깜빡임의 정도가 10프레임 기준 4.2프레임으로 증가함을 확인하였다. 학습된 YOLOv4 모델 이후에 DeepSORT 추적 알고리즘을 적용한 후 동일 비디오 데이터 셋에 테스트를 진행하였을 때에는 평균 10프레임 기준 1.0프레임의 깜빡임 현상이 감지되었고 저화질의 CCTV 영상의 경우 1.9프레임의 깜빡임 현상을 관측하였다. 웹캠 스트리밍 데이터의 경우 화면이 크게 흔들리지 않아 멈춰있는 불꽃의 경우 YOLOv4만 사용했을 때와 추적 알고리즘을 사용했을 때의 두 경우에 대한 탐지율의 차이가 없었지만 불꽃이 움직일 때(가스라이터 혹은 프린팅된 화재 이미지를 좌우로 흔들어 움직임)의 탐지율은 유의미한 차이를 보였다. YOLOv4만 사용한 경우 10프레임 중 1.2프레임의 미탐지율을 보였지만 추적 알고리즘을 적용한 경우는 거의 모든 프레임에 대하여 불꽃을 탐지하는 데 성공하였다. 동일 계산 성능을 가진 시스템 하에서의 평균 탐지 FPS는 YOLOv4 단독 사용 모델의 경우 34FPS, DeepSORT 추가 모델의 경우 32FPS로 큰 차이가 발생하지 않았다.

본 논문에서는 실시간 객체 탐지가 가능한 YOLOv4와 화재를 안정적으로 탐지할 수 있는 객체 추적 기법인 DeepSORT 알고리즘을 적용하였다. 이를 통해 기존의 알고리즘에서 발생하는 깜빡임 현상을 개선하여 탐지 성능을 높였다. 깜빡임 현상이 발생할 경우 화재 발생 중간에 탐지 정보를 잃기 때문에 연속적인 탐지가 불가능하다. 탐지 후 알림으로 이어질 때 이러한 간헐적 끊김이 단점으로써 치명적으로 작용할 수 있다. 기존의 율로를 포함한 객체 탐지 모델을 사용한 연구들의 경우 단순히 정확도 및 속도 등에 집중한 반면 본 연구에서는 객체 추적 알고리즘을 사용하여 객체 탐지 모델만을 사용했을 때와 깜빡임 현상의 발생 정도를 비교하였다.

본 화재 탐지 시스템은 30FPS 미만의 영상을 Input size (608, 608)로 받을 시 YOLOv4 모델과 DeepSORT 추적 알고리즘을 거친 뒤, 화재가 탐지된 이미지 포출 및 저장까지 평균 0.209초가 소요되며, 평균 탐지 FPS는 4.79FPS의 결과를 보인다. 영상의 Input size가 (416, 416)일 시 소요시간은 0.117초로 감소하고, 평균 탐지 FPS는 8.50FPS이다.

따라서 본 시스템의 이미지 획득부터 탐지에 이르는 지연 시간은 0.209초와 0.117초로, 충분히 위 한국정보통신기술협회의 실시간 정의를 충족시킨다고 할 수 있다.

본 실험에서 화재 탐지 테스트 시 주변이 밝은 낮의 화재 이미지들, 연기로 인해 주변이 흐릿한 화재 이미지들 그리고 주변이 어두운 밤 이미지들로 나누어 화재 탐지 mAP를 살펴본 결과 각각 0.68, 0.67, 0.69의 값으로 큰 차이가 없었다.

Table 2. The number of flickering frame per 10 frames

	YOLOv4	YOLOv4 + DeepSORT
Static Video	2.3	1.0
Dynamic Video	1.8	1.3
Static Video (Webcam)	0.7	0.6
Dynamic Video (Webcam)	1.2	0.5
Lo-Fi Video (CCTV)	4.2	1.9

IV. Conclusions

YOLOv4 알고리즘으로 화재 데이터를 학습시켜 CCTV 영상으로부터 화재를 실시간으로 탐지하였고, 이후 DeepSORT 추적 알고리즘을 적용하여 화재 탐지 성능을 보강하였다. 10프레임을 기준으로 평균 미탐지율을 평가하였을 때 DeepSORT 알고리즘을 YOLOv4에 추가로 사용한 경우 YOLOv4 모델만 단독으로 사용한 경우보다 정확도가 2배 가까이 증가함을 확인하였다. Input Video FPS를 30FPS 미만으로 설정할 경우 DeepSORT 알고리즘을 추가하여도 실시간 탐지가 나오는 충분한 탐지 속도를 보여주었으며, 탐지 정확도의 증가 대비 탐지 속도의 하락은 무의미한 정도로 확인되었다.

결과적으로 기존의 YOLOv4 알고리즘만을 사용하여 학습시켰을 때 발생하는 깜빡임 현상을 크게 개선하였다. 깜빡임 현상은 불꽃과 같이 형태가 일정하지 않은 물체를 탐지하여야 할 때 쉽게 발생하고 탐지가 간헐적으로 끊기기 때문에 탐지 정보를 연속적으로 사용하기 어렵게 만든다. 본 연구의 방법을 통해 보다 안정적이고 효과적으로 화재 상황을 탐지할 수 있도록 하였다.

다만 여전히 약간의 깜빡임 현상이 발생하기 때문에 이를 알고리즘 상에서 연속적인 탐지로 처리하는 방식의 추가 연구 및 개발이 필요할 것으로 보인다. 기존의 방법보다 그 횟수가 현저히 줄어든 만큼 추가적인 리소스 사용을 최소로 하여 성능을 보다 개선할 수 있을 것으로 보인다.

주변 환경에 따른 화재 탐지 성능에 차이가 크지 않았기 때문에 추가적인 학습을 통해 탐지율 성능을 개선시키지는 않았다. 화재 환경에 대한 케이스를 분류해 학습을 시킨다면 탐지 정확도는 당연히 올라갈 확률이 크지만, 그만큼 모델이 무거워지고 모델의 화재 탐지 시간이 늘어난

다. 골든 타임 확보를 위한 모델의 예측 시간을 위해서 현재의 모델을 표본 모델로 두고 추후 연구를 통해 화재 상황의 경우의 수를 늘려가면서 화재 탐지율과 지연시간에 관한 추가 연구를 진행할 예정이다.

본 연구는 현재에도 진행 중인 연구로서, 화재 데이터의 전처리 과정 추가와, 추적 알고리즘의 최적화를 통해 YOLOv4 모델의 탐지 성능 지표 중 하나인 mAP를 올리고 비디오 데이터 내에서의 깜빡임 현상을 줄여나갈 것이다. 이에 더하여, 더 넓은 범위의 환경에 대한 화재 데이터를 수집하여 시야 확보가 어려운 상황(극심한 저화질, 야간 상황 등)에서의 추적 알고리즘이 탐지 모델 성능 향상에 미치는 영향에 대해서도 파악할 예정이다. 또한, 현재 임시 테스트 베드로서 설치된 웹캠을 대신하여 실제 CCTV와 연동하여 현재의 테스트 환경보다 실제 상황에 가까운 환경 내에서의 추가 테스트를 계획 중에 있다.

ACKNOWLEDGEMENT

This work was supported by Korea Agency for Infrastructure Technology Advancement(KAIA) (21NSPS-B159110-03).

REFERENCES

- [1] BOCHKOVSKIY, Alexey; WANG, Chien-Yao; LIAO, Hong-Yuan Mark. Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934, 2020.
- [2] VEERAMANI, Balaji; RAYMOND, John W.; CHANDA, Pritam. DeepSort: deep convolutional networks for sorting haploid maize seeds. BMC bioinformatics, 2018, 19.9: 1-9.
- [3] FELZENSZWALB, Pedro; MCALLESTER, David; RAMANAN, Deva. A discriminatively trained, multiscale, deformable part model. In: 2008 IEEE conference on computer vision and pattern recognition. Ieee, 2008. p. 1-8.
- [4] XIONG, Huan, et al. On the number of linear regions of convolutional neural networks. In: International Conference on Machine Learning. PMLR, 2020. p. 10514-10523.
- [5] ALBAWI, Saad; MOHAMMED, Tareq Abed; AL-ZAWI, Saad. Understanding of a convolutional neural network. In: 2017 International Conference on Engineering and Technology (ICET). Ieee, 2017. p. 1-6.
- [6] WELCH, Greg, et al. An introduction to the Kalman filter. 1995.
- [7] MCLACHLAN, Geoffrey J. Mahalanobis distance. Resonance,

1999, 4.6: 20-26.

- [8] MILLS-TETTEY, G. Ayorkor; STENTZ, Anthony; DIAS, M. Bernardine. The dynamic hungarian algorithm for the assignment problem with changing costs. Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-07-27, 2007.
- [9] MILAN, Anton, et al. MOT16: A benchmark for multi-object tracking. arXiv preprint arXiv:1603.00831, 2016.
- [10] REZATOFI, Hamid, et al. Generalized intersection over union: A metric and a loss for bounding box regression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019. p. 658-666.
- [11] ZHENG, Liang, et al. Scalable person re-identification: A benchmark. In: Proceedings of the IEEE international conference on computer vision. 2015. p. 1116-1124.
- [12] PANG, Jiahao, et al. Cascade residual learning: A two-stage convolutional neural network for stereo matching. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. 2017. p. 887-895.
- [13] SENOUSAOUI, Mohammed, et al. A study of the cosine distance-based mean shift for telephone speech diarization. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2013, 22.1: 217-227.
- [14] RONQUIST, Fredrik. Three-dimensional cost-matrix optimization and maximum cospeciation. Cladistics, 1998, 14.2: 167-172.
- [15] Kyung-Min Park., et al. A Study on Fire Detection in Ship Engine Rooms Using Convolutional Neural Network. Journal of the Korean Society of Marine Environment and Safety Vol.25 No.4 pp.476-481
- [16] Jeongsoo Kim et al. (2020). Development of Fire Detection Model for Underground Utility Facilities Using Deep Learning : Training Data Supplement and Bias Optimization. Journal of Korea Academia-Industrial cooperation Society, 21(12), 320-330.
- [17] Siyeong Ryoo, Soong-hwan Ro. A Study on a Video-Based Deep Learning Fusion Algorithm for Fire Detection Systems. Journal of Korea Academia-Industrial cooperation Society, 46(9), 1487-1496.

Authors



Jonghyuk Park received the B.S. in Mechanical Engineering from Yonsei University, Korea, in 2021. Mr. Park is currently a M.S. student in Mechanical Engineering at Yonsei University.

His research interests include AI, Computer Vision, and Deep-Learning.



Dohyun Park received the B.S. degree in Mechanical Engineering from Yonsei University, Korea, in 2020. Mr. Park is currently a M.S. student in Mechanical Engineering at Yonsei University.

His research interests include IoT and Computer Vision.



Donghwan Hyun received the B.S., M.S. degree in Mechanical Engineering from Yonsei University, Korea, in 2020, 2022, respectively. His research interests include System Design and Computer Vision.



Youmin Na received the B.S., M.S. degree in Mechanical Engineering from Ajou University, Korea, in 2019, Yonsei University, Korea in 2022, respectively. Mr.Na's research interests include Mechanical

Design and Computer Vision.



Soo-Hong Lee received the B.S. degree in Mechanical Engineering from Seoul National University, Seoul, Korea, in 1981. He received the M.S. degree in Mechanical Design from Seoul National University,

Seoul, Korea, in 1983. He received the Ph.D. degree in Mechanical Engineering from Stanford University, USA, in 1991. Soo-Hong Lee is currently a professor at the department of Mechanical Engineering at Yonsei University. His research interests include System Design, Process Planning Design, Embedded Software and IoT.