

딥러닝 기반 알고리즘을 활용한 산불 객체 탐지

이 평 화¹ · 송 주 환^{2*}

¹전주대학교 Agro AI학과 석사과정

^{2*}전주대학교 인공지능학과 교수

Forest Fire Object Detection Using Deep Learning-Based Algorithm

Pyeong-Hwa Lee¹ · Ju-Whan Song^{2*}

¹Master's Course, Department of Agro AI, Jeonju University, Jeonju 55069, Korea

^{2*}Professor, Department of Artificial Intelligence, Jeonju University, Jeonju 55069, Korea

[요 약]

우리는 산불로 인한 피해로 직, 간접적인 피해를 받고 있는데 피해를 줄이기 위해서는 초기에 산불을 검출하는 것이 중요하다. 기존 산불 검출 방법으로는 촬영 이미지를 영상처리 기술을 통해 감지하거나 센서를 일정 구간별로 설치하여 모니터링하는 방법을 사용했다. 본 논문은 유지비용이 많이 들어가는 기존 방법을 개선하기 위해 객체 탐지 인공지능 기술, YOLO를 사용하여 객체 검출을 하고 인공지능 모델을 통해 나온 검출 결과를 제공하는 시스템 구현을 진행한다. 연구 결과 주요 성능지표인 mAP가 Proposed LFire 0.959, YOLOv5 0.931, YOLOv4 0.943, R-CNN 0.879가 나왔다. 본 논문에서 제안하는 모델인 Proposed LFire가 가장 높은 것을 통해 산불 검출 데이터에 적합한 객체 검출모델로 볼 수 있다. 향후 보편적으로 사용할 수 있는 RGB 채널 산림 이미지 특징에 맞는 이미지 전처리 및 모델 학습을 통해 검출 성능을 높이는 방향으로 연구할 계획이다.

[Abstract]

We are suffering direct and indirect damage from forest fires, and it is essential to detect them early to reduce the damage. Existing forest fire detection methods used a method of detecting photographed images through image processing technology or installing sensors at certain intervals to monitor them. To improve the existing process that is expensive to maintain, this paper conducts object detection using object detection artificial intelligence technology, YOLO. It implements a system that provides detection results from artificial intelligence models. As a result of the study, the main performance indicators, mAP, were Proposed LFire 0.959, YOLOv5 0.931, YOLOv4 0.943, and R-CNN 0.879. The proposed model in this paper, Proposed LFire, is the highest, which can be seen as an object detection model suitable for forest fire detection data. In the future, we plan to study in the direction of increasing detection performance through image preprocessing and model learning suitable for RGB channel forest image features that can be used universally.

색인어 : 딥러닝, 객체 검출, 분류, 옰로, 산림, 화재

Keyword : Deep Learning, Object Detection, Classification, YOLO, Forest, Fire

<http://dx.doi.org/10.9728/dcs.2022.23.9.1869>



This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 05 August 2022; **Revised** 29 August 2022

Accepted 06 September 2022

***Corresponding Author, Ju-Whan Song**

Tel: +82-63-220-2912

E-mail: jwsong@jj.ac.kr

I. 서론

산림은 대기 중의 이산화탄소방출을 억제하고, 공기 정화, 홍수방지, 수질 정화, 토양 침식방지 등 자연환경의 물질 순환 통로 역할과 재해 예방 기능이 있다[1]. 지구의 심장 역할을 하는 산림이 파괴된다는 것은 지구 환경에서 제공하는 자원들의 공급원이 파괴된다고 보아도 무관할 정도로 많은 자원을 제공한다.

기상변화와 상대적으로 비교하여 분석한 결과 풍속이 강하고 습도가 낮고, 기온이 높은 해의 산불 발생이 많았고 강수일수와의 관계는 미약했으나 강수량과는 밀접한 관계를 보여 강수량이 많은 해의 산불 발생이 적게 나타났다[2]. 세계 각국에서는 지구온난화, 엘니뇨 등 기상이변으로 인한 대형산불이 빈번히 발생해 심각한 환경파괴, 인명 및 재산피해가 발생하고 있다[3]. 미국에서 발생한 산불은 매년 수백 명의 인명을 내고 막대한 재산피해를 발생시킨다. 연방정부의 조사에 따르면 미국 전역에서 산불로 손해 입은 산림, 가옥 등 최소 253억 달러 이상의 경제적 피해가 보고되고 있다. 해외뿐 아니라 전 국토 65%가 산림으로 둘러싸여 있는 우리나라도 산불에 대한 피해 경향이 있다. 2004년 4월 발생한 영동지역 고성, 동해, 강릉, 삼척, 울진 산불은 동해안 지역을 따라 거의 동시다발적으로 발생하여 피해 면적인 23,794ha에 이르고 인명피해 30명, 경제피해액 1,061억 원, 공익적 피해 약 1,851억 원이라는 큰 피해를 기록하였다. 이렇게 산불이 발생하게 되면 수많은 인적, 재산피해뿐 아니라 환경이 파괴되어 자연재해에 대한 발생률이 높아지게 된다. 산불에 대한 기록들은 산불 예방 및 진화에 관한 연구를 통해 우리는 산림을 보존하고 파괴될 수 있는 요소들을 예방해야 한다.

본 논문에서는 산림을 파괴하는 요소 중 산불에 대한 예방과 진화의 중요성에 집중했다. 산불과 관련해서 진화 장비, 진화 시스템, 산불 감지 장비, 산불 감지 시스템 등의 분야에 관한 연구들이 이루어지고 있다. 다양한 매체 및 기관에서 산불 피해 최소화를 위해서는 주야간 24시간 신속하게 화재진화를 위한 전문인력과 장비를 갖춘 전문기관의 초기대응과 주민 안전을 위한 긴급대피와 산불 현장 중심의 대응 체계에 대한 중요성에 대해서 다루고 있다[4]. 산불은 진화가 늦어질수록 급속하게 넓은 면적으로 확대되기 때문에 불이 나고 골든타임 50분 이내에 진화하는 것이 중요하다. 산림청에서는 골든타임을 정하여 대응하려고 노력하고 있다[5].

현재까지는 산림청 관할 통제실에서 모니터링 화면을 통해 산림에 설치된 감시 장비를 통해 감시하는 직원을 통해 산불 감지를 해오고 있다. IoT(Internet of Things) 장비를 이용하더라도 초기 설치 비용 및 유지하기 위한 유지보수 비용 그리고 수리 및 감시 등에 많은 유지비용이 발생하게 됨을 알 수 있다.

유지보수 비용을 줄이고 감시율은 높일 방안으로 딥러닝 기반으로 이미지를 분류하거나 객체 인식 알고리즘을 활용해 객체를 분류하고 위치 정보를 얻는 등의 연구들이 진행되고

있다. 딥러닝 기술이 등장하면서 컴퓨터 비전(Computer Vision) 기술이 빠르게 발전하고 있고 이미지 인식 능력이 사람의 인식 능력을 넘어서고 있다. 이미지의 특성 및 패턴을 처리하는 CNN(Convolution Neural Network)을 사용하여 이미지의 공간 정보를 유지한 채 학습하기에 기존 영상처리 기술의 손실률을 보완하고 인식률은 높게 됐다. CNN은 Localization과 Classification 정보를 포함해 처리하는 객체 검출(Object Detection) 분야에도 적용된다. 객체 검출은 이미지 분류를 포함한 이미지에서 해당 분류 데이터의 위치 검출이 가능하다.

산불 연기 데이터셋에 대한 검출 기술 비교 분석 성능평가를 통해 산불 이미지에 대한 전체적인 오검출이 많은 것으로 볼 수 있다[6]. 본 논문에서는 제안하는 개선 방안으로 Two-stage Detector 기법을 사용하는 객체 검출 알고리즘을 사용하여 개선하고자 한다. 대표적인 Two-stage Detector 기법을 사용하는 YOLO(You only look once) 알고리즘을 사용하였다. 검출에 사용하는 컴퓨터의 사양에 따라서 처리 속도가 달라질 수 있다. 이를 고려하여 훈련 시 입력 사이즈는 416x416으로 훈련하지만, 검출 단계에서는 검출 이미지의 사이즈를 256x256 이하로 변경하여 검출하도록 시스템을 구성했다. 산불 검출 훈련모델에 대한 성능지표로는 mAP, 재현율, 정밀도에 대한 90% 이상의 성능향상을 목표로 했다.

본 논문의 구성은 다음과 같다. 2장에서 관련된 연구를 소개하고, 3장에서 산불 감지 인공지능 모델 학습 및 시스템을 제안하고 4장에서 3장을 바탕으로 연구한 내용과 결과를, 5장에서 본 논문의 결론을 서술하였다.

II. 관련 연구

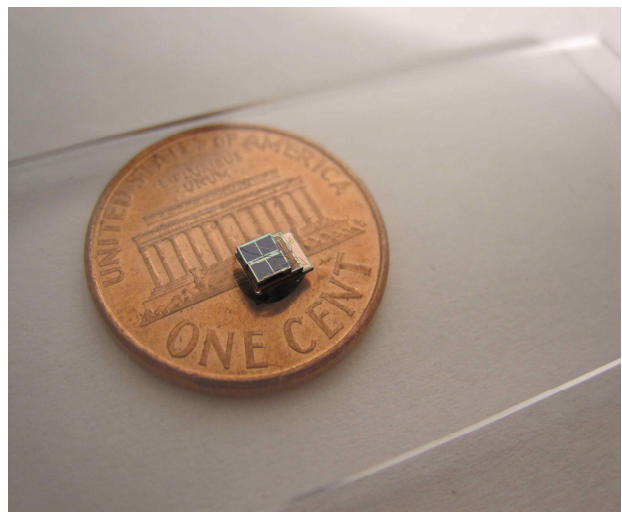


그림 1. IoT 화재 감지 센서 : Smart Dust
Fig. 1. IoT Fire Detection Sensor : Smart Dust

2-1 기존 산불 검출 기술

1) IoT 화재 감지 센서를 이용한 검출

산불이 발생할 때 먼지, 가스 등의 요소들이 사방으로 확산하게 되는데 이를 스마트 센서, 아날로그 센서 등의 센서들을 IoT에 접합하여 서버와 통신하여 임계치 이상의 데이터가 송신될 시 화재 발생 가능성이 큰 것으로 판단하여 관리자에게 알려주는 시스템을 구축하여 사용한다. IoT 화재 감지 센서에는 온도 감지기와 연기 감지기와 열을 감지하는 불꽃 감지기, 네덜란드 Twente 대학의 텔레커뮤니케이션 연구실에서 연구하여 개발된 그림1, 스마트더스트와 같은 스마트 센서도 있다[7]. IoT 화재 감지 센서를 이용한 산불 검출을 위해서는 초기 설치 및 유지보수에 들어가는 비용과 인력에 대한 지속적인 소비를 해야 하는 단점을 가지고 있다.

2) 영상처리 기술을 이용한 검출

산불 감지에 영상처리 기술을 활용하려면 산 곳곳에 센서 및 IoT 장비들을 설치 및 유지보수하지 않고 산림마다 설치되거나 이동하는 무인 드론, 비행체로부터 송신되는 영상을 수신하여 산불을 감지할 수 있다. HSV(Hue Saturation Value) 채널을 이용한 처리 기술에서는 영상을 RGB(Red Green Blue) 채널을 HSV 채널로 변경한 후 배경을 제거하고 미디안 블러(Median Blurring)를 사용해 잡음을 제거했다[8]. 다음으로 HSV 채널에서 붉은 영역만 남게 하여 이동하는 사람과 꾸준히 이동하는 화염만 남게 하여 산불 감지에 활용하게 된다.

YCbCr 채널을 이용한 처리 기술에서는 이미지 사이즈를 정규화하여 연산 속도를 높이고 Color를 정규화하여 불필요한 검출을 줄인 후 RGB 채널을 YCbCr 채널로 변환하여 작은 데이터로 더욱 많은 색상표현을 가지도록 하였다. Color를 해석하고 질감 특징을 분석하여 화재 영역을 검출하는 알고리즘을 제안했다. 이외에도 영상 촬영 장비의 특성이 적용된 영상으로부터 받아온 정보를 영상 처리하여 산불 감지에 활용하기도 한다[9].

3) 인공지능 기술을 이용한 산불 검출

인공지능 분야에서 분류에 많이 사용되는 CNN을 활용한 검출 연구들이 이루어지고 있다. 정성우는 IoT, 영상처리 기술을 이용한 기존 방법이 아닌 이미지 인식과 패턴 감지에 효과적인 CNN을 사용하였다. 기존 FireNet의 일반화 문제점을 개선하여 새로운 경량형 CNN 모델인 i-FireNet 모델을 제안하여 분류 성능이 향상하고 파라미터 개수를 줄여 메모리 크기를 대폭 줄이는 결과를 냈다[10]. 이미지 전처리와 임계화(Thresholding)을 진행한 후 검출 단계로 경계찾기(Contour detection), 분류(Classification) 과정을 차례로 진행하는 One-stage Detection 시스템 구조를 갖는다. 성능 평가로는 Accuracy 90.40%, Precision 93.53%, Recall 86.81%, F1 90.05%의 결과를 나타냈다. 객체 검출로 잘 알

려진 YOLOv3, Faster R-CNN, Retina-Net, Corner-Net, Center-Net에 대한 성능 비교를 보면 Two-stage Detection으로 진행되는 YOLOv3 훈련한 모델 검출 결과가 Precision 48.32%, Recall 64.17%로 논문에서 제시한 성능 평가표에서 가장 높은 결과를 나타냈다[6]. 그러나 산불은 우리에게 큰 피해를 주는 위협 요소로 객체 탐지 및 분류에 대한 인식률이 높아진다. 본 논문에서는 YOLO를 이용해 더 높은 성능 결과의 모델을 개선하고자 한다.

2-2 객체 검출

객체 검출은 Computer Vision과 Image Processing 등을 활용해 분류와 위치 정보(Localization)를 찾는 컴퓨터 기술이다. Localization은 Bounding Box를 찾는 회귀(Regression)를 진행하고 Classification은 Bounding Box 내 물체가 무엇인가 분류하는 문제를 의미한다. 객체 검출은 (1)과 같이 Classification과 Localization에 대한 정보를 가지는 검출 방법을 말한다. 검출 방식으로는 One-stage Detector와 Two-stage Detector로 나뉜다. 그림2는 One-stage와 Two-stage Detector를 비교한 그림이다. One-stage Detector는 분류와 지역화를 동시에 처리하기 때문에 처리 속도가 빠르지만, 정확도가 떨어진다. YOLO, SSD가 대표적인 예이다. Two-stage Detector는 분류와 지역화 작업을 나눠서 차례로 처리하는데 R-FCN, Faster R-CNN, Mask R-CNN 등이 대표적인 예이다.

본 논문에서는 산림에서 발생하는 산불 발생 여부를 분류하고 발생 위치를 식별하여 정보를 제공하는 시스템을 구현하고자 한다. 위치 정보 식별과 분류를 처리할 수 있는 객체 검출 연구를 진행하였다. 분류와 위치 정보를 동시에 처리하여 실시간 분류에 적합한 YOLO 알고리즘을 본 논문의 객체 검출 알고리즘으로 선택하였다.

$$object\ detection = classification + localization \quad (1)$$

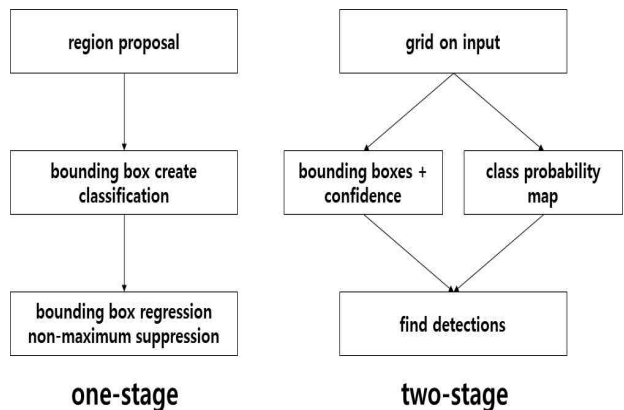


그림 2. 객체 검출 단계
Fig. 2. Object Detector Step

2-3 YOLO

YOLO는 이미지 내에 존재하는 객체와 해당 객체의 위치는 이미지를 한 번만 보고 예측할 수 있는 알고리즘이다[11]. 그림3은 YOLO의 시스템 구조를 나타낸 그림이다. 이는 분류 대상의 객체로 감지하는 대신에, 경계 박스를 다차원적으로 분리하고 클래스 확률(Class Probability)을 적용하여 하나의 회귀 문제로 접근한다. 입력 이미지는 CNN을 통해 텐서(Tensor)의 그리드 형태로 나누고, 각 구간에 따라 객체인 경계 박스와 클래스 확률을 생성하여 해당 구역의 객체를 인식한다. YOLO는 후보 영역을 추출하기 위한 별도의 네트워크를 적용하지 않기 때문에 Faster R-CNN보다 처리시간 측면에서 월등한 성능을 보인다. 그림4는 YOLOv1의 네트워크 구조를 나타내는데, CNN GoogleLeNet 모델 구조를 기반으로 하며 Inception Model을 제외한 Convolution Layer 20 개에 Convolution Layer 4개를 추가하고 분류를 위한 FC Layer 2개를 추가한 구조를 갖는다. 입력 이미지 크기를 조정하고 이미지를 Convolution Network에 입력하여 특징을 추출한다. 그리고 모델의 신뢰도에 따라 표1과 같은 과정을 통해 인코딩한다.

YOLO에서 이미지 처리 방법은 표1과 같다. 입력 이미지의 크기를 설정값에 맞게 조정하고, CNN Model에 적용하여 나온 신뢰도에 따라 인코딩(2)하여 결과 탐지를 진행한다. 다음 과정은 그림5와 같다. 첫 번째로 SxS Grid로 Input Image를 나눈 다음, Bounding Box + Confidence Score 계산과 Conditional Class Probabilities 계산을 동시에 진행한 결과를 가지고 SxS(b*5+ c) Tensor 형태로 인코딩하게 된다. Bounding Box에는 x, y, w, h, Confidence Score가 담긴 총 5개의 요소로 구성되게 된다. x와 y는 Grid cell의 경계를 기준으로 Bounding Box의 중심 좌표를 의미한다. w와 h는 Bounding Box의 너비와 높이를 의미한다. Confidence Score는 Pr(Object) * IOU(truth, pred) 값으로 예측한 Bounding box에 object가 있는지 없는지에 대한 확률값과 예측한 IOU(3)를 구하여 계산한다. Cell에 Object가 존재하지 않는다면 Pr(Object)가 0이 되어 Confidence Score는 0이 된다. Conditional Class Probabilities (4)계산 과정에서는 각 Grid cell에서 class 확률이 높은 class를 할당하게 된다.

$$Pr(Class_i|Object) * Pr(Object) * IOU_{pred}^{truth} = Pr(class_i) * IOU_{pred}^{truth} \quad (2)$$

$$IOU = \frac{Area of \cap}{Area of \cup} \quad (3)$$

$$Pr(Class_i|Object) \quad (4)$$

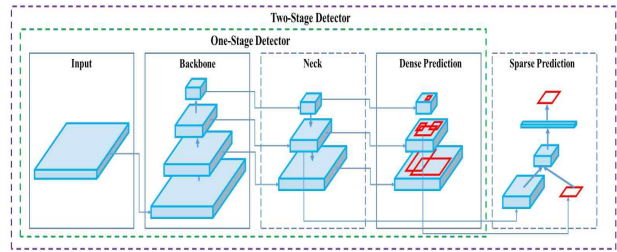


그림 3. 객체 검출
Fig. 3. Object Detector

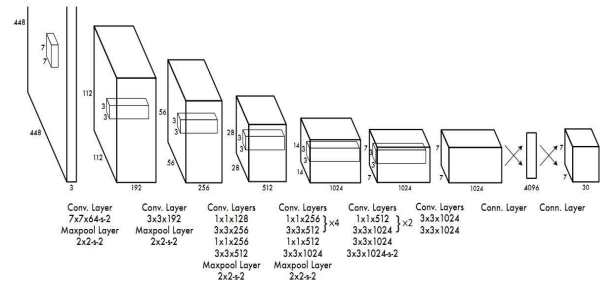


그림 4. YOLO 네트워크
Fig. 4. YOLO Network

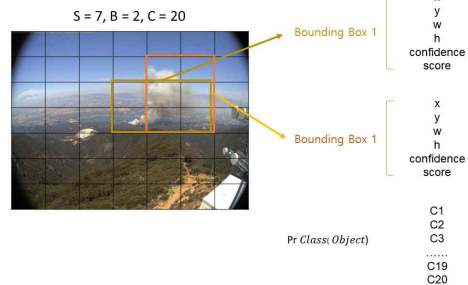


그림 5. 객체 검출 단계
Fig. 5. Object Detector Step

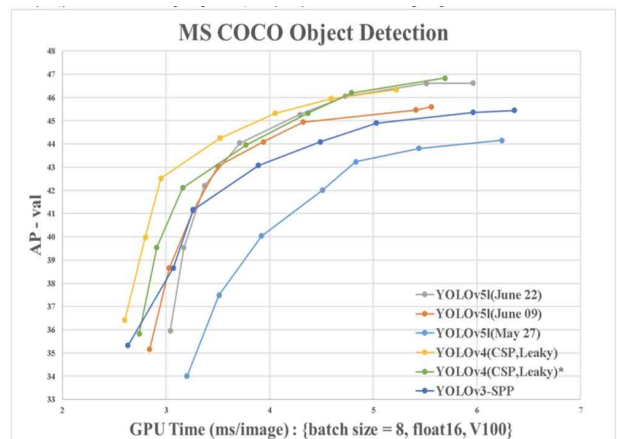


그림 6. COCO 객체 검출 그래프
Fig. 6. COCO Object Detection Graph

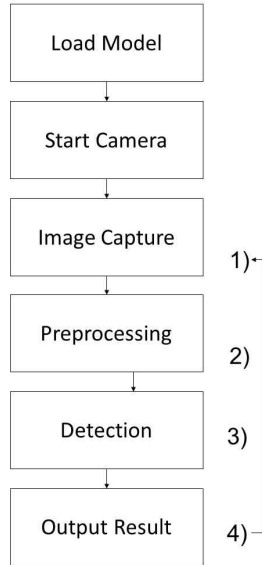


그림 7. 시스템 구조
Fig. 7. System Framework

표 1. YOLO 인코딩 과정
Table 1. YOLO Encoding Process

Stage	Process
1	Image to S x S Grid
2	Calculate the center coordinates x,y of the Binding Box and the width, height of the Box for each Grid cell
	Confidence Score, calculating probability of being an object (Object exists = IOU, object does not exist = 0)
	Assign a class with a high class probability in each Grid cell separately from the Box (calculated only when there is an Object)
3	S x S x (B * 5 + C) Tensor Encoding

표 2. 데이터셋 구성
Table 2. Dataset Configuration

Data	Training	Validation	Test
Count	550	147	40

III. 산불 감지 인공지능 모델 학습 및 시스템 구현

제안하고자 하는 시스템은 기존 IoT 장비를 이용해 지속적인 유지보수가 필요하거나 객체 검출 과정에서 Two-stage 처리 방법을 이용해 처리 속도가 느린 시스템이 아닌

One-stage 처리 방법이 적용된 YOLO 알고리즘을 활용하였다. YOLO 알고리즘 모델을 수정하면서 산불 이미지에 적합한 파라미터들을 찾아 성능을 높였고 학습한 모델을 활용해 검출하는 시스템을 제안하고자 한다.

3-1 데이터셋 구성

컴퓨터 비전(Computer Vision) 기술을 이용해 다양한 애플리케이션에 활용할 수 있도록 지원하는 Roboflow 서비스에서 제공하는 Wildfire Smoke 데이터셋을 Train, Test, Valid 데이터셋으로 선정하였다. 학습을 최적화하기 위해 데이터셋 기존 Image Size를 학습할 Model의 Input Size로 변환하여주고 데이터셋을 Train 550개, Valid 147개, Test 40개로 나누어 학습에 필요한 사전 이미지 준비를 하였다. 학습에 필요한 라벨링 작업을 통해 위치와 Class 정보를 얻어 개체 종류와 위치를 예측에 사용하였다.

3-2 모델 학습

사전에 준비한 데이터셋을 합성곱 신경망 알고리즘 모델에 적용해 산불 객체 검출할 수 있도록 훈련한다. 그림6에서 YOLO 알고리즘에서 AP와 ms/image를 평가한 지표 중 시간 대비 인식률(AP)이 좋은 YOLOv5 버전을 사용하였다.

YOLO에서는 총 24개의 Convolution Layer와 Max-Pooling Layer가 한 쌍으로 이루어 층을 이룬다. 이 레이어들을 반복한 후 2개의 Fully Connected Layer를 통해 Reshape 되어 데이터를 1차원에서 Image의 원본 채널과 같은 3차원으로 변환한다. Class와 Object 위치를 단일 회귀를 통해 분류와 객체 위치를 예측하는 과정을 하나의 과정에서 동시에 수행하는 One-stage Detector를 진행한다. 마지막으로 최적의 Bounding Box를 찾는 비 최대 억제(Non-maximum Suppression)를 통해 검출기(Object Detector)가 예측한 많은 Bounding Box 중 가장 높은 예측치의 위치와 크기의 Bounding Box 하나를 선정한다. 이 과정을 통해 훈련된 모델은 학습을 바탕으로 개체 분류와 위치를 예측해 객체를 검출한다.

YOLO의 검출 구조는 Head와 Backbone으로 구성된다. Head는 추출된 Feature map을 기반으로 물체의 Location을 찾게 된다. 이를 통해 Anchor Box를 설정하고 이를 통해 최종 Bounding Box를 생성한다. Backbone은 이미지로부터 Feature map을 추출하는 역할을 하게 된다.

YOLOv5에서 설정하는 모델 파라미터 중 깊이와 너비 값을 지나치게 크게 설정하면 연산량이 늘어나 속도는 느려지고 정확도가 낮아질 수 있다. 본 논문에서는 YOLOv5 기존 모델 파라미터 중 Backbone 신경망 깊이와 너비를 수정하였고 산불 데이터셋을 YOLOv5 기존 모델에 학습한 결과보다 본 논문에서 개선한 Proposed LFire 모델이 좋은 성능을 보이는 것을 볼 수 있었다.

3-3 객체 검출 시스템 구현

객체 검출 시스템 구현의 과정을 간단히 표현하면 입력 단계 - 전처리 단계 - 검출 단계 - 출력 단계로 나누어 볼 수 있다. 카메라로부터 입력받은 이미지를 Python으로 작성한 객체 검출 시스템에서 읽어와 처리하도록 구현했다. 시스템을 시작하게 되면 라이브러리들을 포함하게 된다. 다음으로 Pytorch에서 제공하는 함수를 활용하여 사전에 훈련한 Proposed LFire 모델을 불러와 검출을 준비한 후 그림7과 같은 단계를 반복하여 검출을 진행하게 된다.

1) 입력 단계

입력 단계에서는 영상 또는 이미지를 훈련모델에서 처리할 수 있는 형태로 읽어와 변환하여주는 단계를 말한다. 본 연구에서는 OpenCV-Python 라이브러리를 이용했다.

2) 전처리 단계

이미지를 검출하기 전 모델에서 정의한 Input Size에 같은 사이즈로 이미지를 변환한다. 본 연구에서는 416x416으로 변환하고 잡음제거(Denoise)를 위해 Normalize, Gaussian Blur를 적용하여 전처리했다.

3) 검출 단계

전처리한 이미지를 사전에 훈련한 모델을 Pytorch에서 다양한 함수를 제공하는데, 모델을 불러오는 함수를 이용해 모델을 불러온다. 불러온 모델에 예측치에 대한 임계값을 설정하여 임계값 이하의 데이터는 제외하도록 설정한다. 이미지와 Image Size를 함께 입력해주면 검출을 진행한다. 반환된 데이터는 Pytorch에서 pandas 등의 Type으로 변환해주는 함수를 제공해주는데 본 구현에서는 pandas Type으로 변환해주는 함수를 사용하였다.

4) 출력 단계

검출 단계에서 나온 pandas type의 결과를 YOLO 라이브러리에서 제공하는 plot_one_box 함수를 사용하여 Bounding Box를 그릴 수 있도록 변환해준 후 Opencv-python을 이용해 입력 이미지에 그림8과 같이 출력한다. 출력 단계에서 나온 결과에 따라 관리자 단말기로 알림을 보내 산불 초기 대응할 수 있도록 서비스를 진행할 수 있다.

IV. 연구내용

4-1 연구 환경

본 연구에서는 Roboflow 서비스에서 제공하는 Wildfire Smoke 데이터셋을 내려받아 훈련할 모델 Input Size와 같이 변환하여 사용하였다[12]. 훈련 데이터 학습은 표3과 같은

환경에서 진행하였다. OS Ubuntu 18.04.5 LTS, RAM 256GB, Intel(R) Xeon(R) Gold 5218R CPU @ 2.10GHz 와 학습에 CPU보다 GPU를 사용하는 것이 좋은 연산 성능을 보여주기 때문에 NVIDIA의 A100-PCIE-40GB를 사용했다. GPU를 사용하기 위해 CRUD 11.5버전을 활용했다. Python은 3.9.12버전으로 Anaconda 가상환경에서 작업을 진행했다. YOLO를 사용하기 위해 Pytorch 1.11.0 버전을 사용하였고 YOLO의 5번째 모델 YOLOv5를 사용한다. 학습 및 테스트에 사용될 이미지의 Object Bounding Box 정보를 입력하는 데 사용한 툴은 Labellmg를 사용하여 Class와 Bounding Box 정보를 txt파일로 저장하여 사용했다[13].

4-2 성능지표

객체 검출에 대한 성능평가 지표로는 Precision, Recall, mAP(mean Average Precision)를 사용하였다[14]. 해당 지표들을 사용하기 위해서는 TP(True Positive), FP(False Positive), FN(False Negative), TN(True Negative)에 대한 이해가 필요하다. 표4를 기준으로 1사분면 FN은 검출되지 말아야 할 값을 검출하지 않을 값으로 분류한 경우, 2사분면 TP는 검출되어야 할 값을 검출 값으로 분류한 경우, 3사분면 FP는 검출되지 말아야 할 값이 검출되어야 할 값으로 분류한 경우, 4사분면 TN은 검출되어야 할 값을 검출하지 않을 값으로 분류한 경우를 의미한다.

$$Precision = \frac{TP}{TP + FP} = \frac{TP}{AllDetection} \tag{5}$$

모델 예측치 중 옳다고 검출한 결과 중 실제로 옳은 검출한 비율을 (5)와 같은 수식을 통해 정밀도(Precision)를 계산한다.

$$Recall = \frac{TP}{TP + FN} = \frac{TP}{AllGround Truth} \tag{6}$$

실제 옳은 값 중 모델이 옳다고 검출한 비율을, (6)과 같은 수식을 통해 재현율(Recall)을 계산하여 성능 평가지표에 사용한다.

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \tag{7}$$

모델 정확도가 높아도 재현율이 낮으면 객체 검출의 일부만 검출할 가능성이 있고 재현율이 높아도 정밀도가 낮으면 옳지 않은 객체를 옳다고 검출할 수 있는 오류를 발생시킬 수 있다. 산불이 발생하지 않았는데 발생하였다고 하여 소방 인력이 출동하는 비용이 발생하는 것도 손실이지만 산불이 발생하였는데 산불이 발생하지 않았다고 검출하면 더욱 큰 손해를 발생시킬 수 있다. 그러므로 AP(Average Precision)을

사용하도록 한다. AP는 알고리즘의 성능을 하나의 값으로 표하는데 x값을 Recall, y 값을 Precision으로 그래프로 나타낸 Precision-Recall 곡선의 아래 면적을 의미한다. 각각의 class에 대한 AP의 평균을 구한 mAP를 (7)과 같이 식을 통해 구하여 지표로 사용하게 된다. 본 연구에서는 mAP를 주요 성능지표로 선정해 사용하였다.

표 3. 실험환경

Table 3. Experiment Environment

Equipment	Model
Processor	Intel(R) Xeon(R) Gold 5218R CPU @ 2.10GHz
	NVIDIA A100-PCIe-40GB
RAM	256GB
OS	Ubuntu 18.04.5 LTS

표 4. 오차 행렬

Table 4. Confusion Matrix

Ground truth	Predict Result	
	Positive	Negative
Positive	TP (True Positive)	FN (False Negative)
Negative	FP (False Positive)	TN (True Negative)

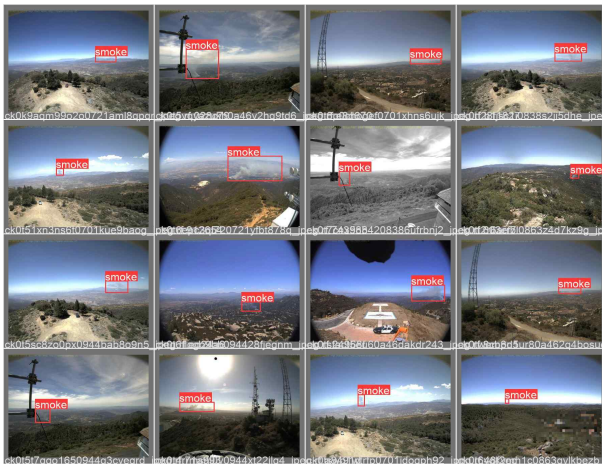


그림 8. 실험 이미지

Fig. 8. Experiment Image

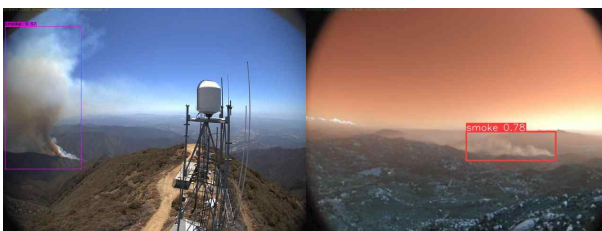


그림 9. YOLOv4, YOLOv5 객체 검출 결과

Fig. 9. YOLOv4, YOLOv5 Object Detection Result

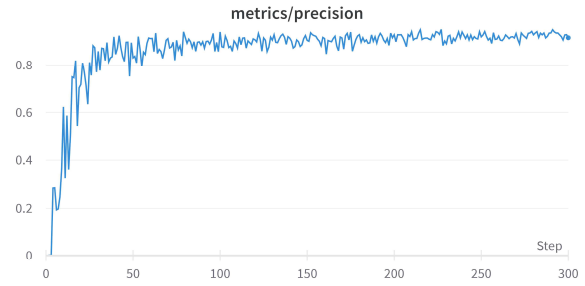


그림 10. 정밀도 결과

Fig. 10. Precision Result

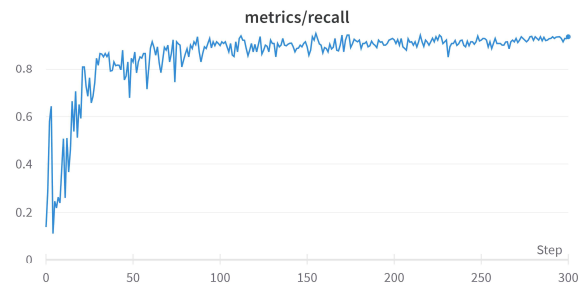


그림 11. 재현율 결과

Fig. 11. Recall Result

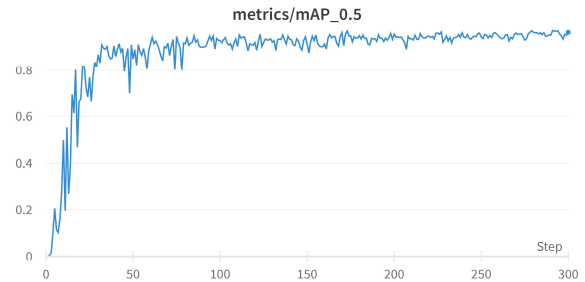


그림 12. mAP 결과

Fig. 12. mAP Result

표 5. 알고리즘 성능 비교 결과

Table 5. Algorithm Performance Comparison Results

	Precision	Recall	mAP
Proposed LFire	0.905	0.925	0.959
YOLOv5	0.907	0.936	0.931
YOLOv4	0.920	0.920	0.943
R-CNN	0.876	0.918	0.879

V. 결론

본 논문은 실시간 산불 감지를 위해 인공지능 분야의 CNN 모델을 산불 감지 모델에 적용하였다. 산불 검출에 사용할 모

델을 학습하기 위해 산불 이미지 데이터셋에서 산불 영역에 Bounding Box와 Class를 기재하는 라벨링을 진행하였다. 객체 검출에서 Localization과 Classification을 처리하게 되는데, 차례로 처리하는 One-stage Detector 기법이 아닌 동시에 처리하는 Two-stage Detector 기법을 사용하는 YOLO 알고리즘을 사용하였다. YOLO 알고리즘은 객체 검출을 실시간 영상에서 검출할 수 있다. 기존 YOLOv5 모델을 산불 데이터에 적합한 파라미터를 찾아 수정하여 Proposed LFire 모델로 훈련하였다. 산불 검출 시스템을 구현하기 위해 빅데이터 및 인공지능 분야에서 인기 있는 Python 프로그래밍언어로 코드를 작성하였다. 컴퓨터 비전과 관련된 라이브러리 OpenCV를 이용해 영상을 읽어와 Normalize, Gaussian Blur를 적용하는 이미지 전처리를 하였다. 사전 학습한 Proposed LFire 모델을 불러와 검출을 진행해 결과를 그림9와 같이 출력해주는 과정을 반복하는 시스템 구현을 진행했다. 성능평가는 본 논문에서 제안하는 Proposed LFire 그리고 객체 검출 알고리즘에서 YOLOv5, YOLOv4와 R-CNN에 대한 평가를 진행했다. 그림10은 Proposed LFire 훈련모델의 Precision 수치를 그래프로 그린 결과이고 그림11은 Proposed LFire 훈련모델의 Recall수치를 그래프로 그린 결과, 그림12는 Proposed LFire 훈련모델의 mAP 수치를 그래프로 그린 결과이다. 표5를 보면 본 논문에서 실험한 결과를 볼 수 있는데, Precision은 Proposed LFire 0.905, YOLOv5 0.907, YOLOv4 0.92, R-CNN 0.876, Recall은 Proposed LFire 0.925, YOLOv5 0.937, YOLOv4 0.92, R-CNN 0.918, mAP는 Proposed LFire 0.959, YOLOv5 0.931, YOLOv4 0.943, R-CNN 0.879로 전체적인 성능지표가 Proposed LFire가 높은 것을 볼 수 있다. 실험 결과를 통해 Proposed LFire가 산불 검출 알고리즘으로 적합한 것을 알 수 있다.

지역별 산지에 설치된 CCTV나 감시용 카메라, 촬영용 드론 등 영상기기가 부착된 곳으로부터 받은 이미지에 대해 이미지 전처리를 통해 잡음제거를 하고 제안하는 훈련모델을 통해 산불 검출을 진행하여 산불 발생 여부와 산불 발생 시 발생 위치 결과를 출력하여 초기 산불 진화에 효과적으로 활용될 수 있다. 보편적으로 사용할 수 있는 RGB 채널 산불 이미지 특징에 맞는 이미지 전처리 및 모델 학습을 통해 검출 성능을 높이는 방향으로 향후 연구 계획이다.

참고문헌

[1] Jae-Gun Seo, Research on improvement plan for wildland fire precaution and extinguishment system in Korea, Urban Science, University of Seoul, February, 2008.
 [2] Ji-Eun Kim, A Study on Meteorological Elements Effecting on Large-scale Forest Fire during Spring Time in Gangwon Young-dong Region, Disaster Prevention, Kangwon

National University, February, 2011,
<https://doi.org/10.9798/KOSHAM.2011.11.1.037>
 [3] Mi-Gyeom Kim, "A Review of Mountain Fire Disaster in Korea", Disaster Management, Kwandong University, 2008.
 [4] Young-guk Ju, A Study on the problems and improvement by analyzing, Degree in Fire Science, Graduate School of Wonkwang University, October, 2021.
 [5] Korea Forest Service, Comprehensive Plan on Domestic Fire Protection in 2020 [Internet]. Available:
<https://www.forest.go.kr..>
 [6] Kwang-Ju Kim, In-Su Jang, Kil-Taek Lim, Construction of Wild-fire Smoke Data-set and Comparative Analysis of Detection Method based on Deep Neural Network, Proceedings of the Korean Institute of Communication Sciences Conference, KICS Winter Conference 2021, pp.1172-1173, February, 2021
 [7] Towards Smart Dust [Internet]. Available:<http://uatrobotics.blogspot.com/2010/03/towards-smart-dust.html>.
 [8] Min-Seok Seo, Choong Ho Lee, "A Forest Fire Detection Algorithm Using Image Information", Journal of the Institute of Convergence Signal Processing, Vol. 20, No. 3, pp. 159~164, September, 2019.
 [9] Chang, Dae-Woong, Studies on fire detection using a YCbCr color model and textur, Department of Electronic and Telecommunication Engineering, Graduate school of Industry, Pukyong National University, February, 2011.
 [10] Seong-Woo Jeong and Joonhyuk Yoo, "i-FireNet: A Lightweight CNN to Increase Generalization Performance for Real-Time Detection of Forest Fire in Edge AI Environments", Journal of Institute of Control, Robotics and Systems, Vol 26, No 9, pp. 802-810, September, 2020. <https://doi.org/10.5302/j.icos.2020.20.0033>
 [11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779-788, May 2016. <https://doi.org/10.1109/CVPR.2016.91>
 [12] Roboflow Dataset for Object Detection [Internet]. Available: <https://roboflow.com/>.
 [13] Labeling for Object Detection [Internet]. Available: <https://www.makesense.ai/>.
 [14] Jin-Gu Kima, Ji-Hun Kangb, "Development of a Normal/Abnormal RBC Detection Algorithm using YOLOv5", Journal of the Korean Society of Manufacturing Technology Engineers, Vol. 31, No. 2, pp 94-100, April, 2022, <https://doi.org/10.7735/ksmte.2022.31.2.94>

이평화(Pyeong-Hwa Lee)



2022년 : 전주대학교 스마트미디어학과 (학사졸업)

2022년: 전주대학교 스마트미디어학과

2022년~현 재: 전주대학교 일반대학원 Agro AI학과 재학 중

※관심분야: 웹서비스, 모바일 서비스, 인공지능, 스마트팜

송주환(Ju-Whan Song)



1997년 : 전북대학교 대학원 (이학석사)

2003년 : 전북대학교 대학원 (이학박사-컴퓨터그래픽스)

2006년~현 재: 전주대학교 인공지능학과 교수

2019년~현 재: 전주대학교 일반대학원 Agro AI학과 교수

※관심분야: 인공지능, 스마트팜, 영상처리, 사물인터넷(IoT), 디지털콘텐츠, 모바일 프로그래밍