



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2025년06월25일
(11) 등록번호 10-2825407
(24) 등록일자 2025년06월23일

- (51) 국제특허분류(Int. Cl.)
G06V 10/82 (2022.01) G06N 3/04 (2023.01)
G06N 3/08 (2023.01) G06V 10/774 (2022.01)
G06V 10/776 (2022.01)
- (52) CPC특허분류
G06V 10/82 (2022.01)
G06N 3/04 (2023.01)
- (21) 출원번호 10-2024-0198872
(22) 출원일자 2024년12월27일
심사청구일자 2024년12월27일
- (56) 선행기술조사문헌
Kavya, "Lightweight and Efficient YOLOv8 With Residual Attention Mechanism for Precise Leukemia Detection and Classification", IEEE Access(2024.10.23.) 1부.*
Manar, "Squirrel Search Optimization with Deep Transfer Learning-Enabled Crop Classification Model on Hyperspectral Remote Sensing Imagery" (2022.06.02.) 1부.*
*는 심사관에 의하여 인용된 문헌

- (73) 특허권자
국립창원대학교 산학협력단
경상남도 창원시 의창구 창원대로 20(퇴촌동)
- (72) 발명자
권오설
경상남도 창원시 의창구 중동로 78, 411동 2702호(중동, 창원 중동 유니시티4단지)
- 서정환
경상남도 창원시 의창구 사림로158번길 17-1, 205호(사림동)
- (74) 대리인
양성보

전체 청구항 수 : 총 10 항

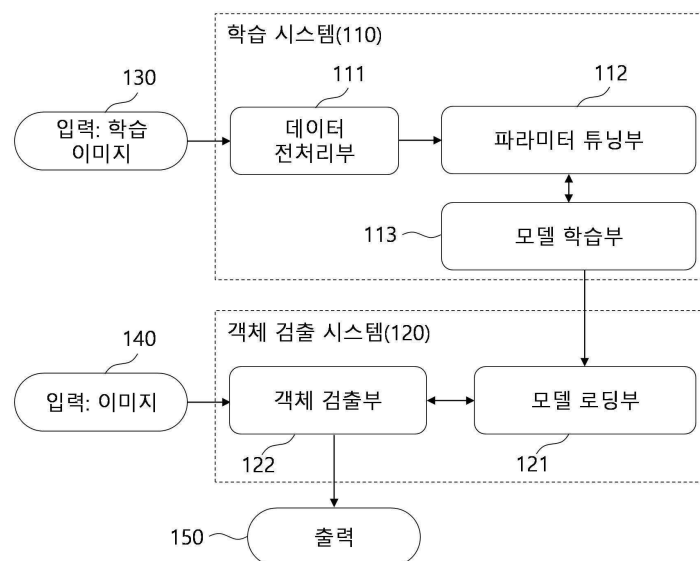
심사관 : 정수진

(54) 발명의 명칭 원격 탐지 영상의 객체 검출 방법 및 시스템

(57) 요약

원격 탐지 영상의 객체 검출 방법 및 시스템을 개시한다. 일실시예에 따른 객체 검출 방법은, SSA(Squirrel Search Algorithm)에 기반하여 튜닝된 하이퍼 파라미터에 따라 사전 학습된 YOLO(You Only Look once) 모델을 로딩하는 단계, 객체 검출을 위한 이미지를 입력받는 단계, 및 상기 이미지를 상기 YOLO 모델에 입력하여 상기 YOLO 모델이 출력하는 상기 이미지가 포함하는 객체에 대한 정보를 제공하는 단계를 포함할 수 있다.

대표도 - 도1



(52) CPC특허분류
G06N 3/08 (2023.01)
G06V 10/774 (2023.08)
G06V 10/776 (2023.08)
G06T 2207/20081 (2013.01)
G06T 2207/20084 (2013.01)

이 발명을 지원한 국가연구개발사업

과제고유번호 2710008858
과제번호 RS-2024-00438409
부처명 과학기술정보통신부
과제관리(전문)기관명 정보통신기획평가원
연구사업명 정보통신방송혁신인재양성사업(대학ICT연구센터)
연구과제명 인공지능 기반 위성영상 고도화를 통한 실감화 및 응용 연구
과제수행기관명 국립창원대학교 산학협력단
연구기간 2024.07.01 ~ 2031.12.31

이 발명을 지원한 국가연구개발사업

과제고유번호 1345370810
과제번호 2021RIS-003
부처명 교육부
과제관리(전문)기관명 한국연구재단
연구사업명 지자체-대학 협력기반 지역혁신 사업
연구과제명 (울산·경남지역혁신플랫폼, 미래모빌리티사업단) 스마트 센싱 기반 협력적 군집주

행 모빌리티 플랫폼 개발

과제수행기관명 국립창원대학교
연구기간 2021.09.01 ~ 2025.01.31

공지예외적용 : 있음

명세서

청구범위

청구항 1

적어도 하나의 컴퓨터 장치에 의해 구현되는 객체 검출 시스템의 객체 검출 방법에 있어서,

상기 적어도 하나의 컴퓨터 장치는 적어도 하나의 프로세서를 포함하고,

상기 객체 검출 방법은,

상기 적어도 하나의 프로세서에 의해, SSA(Squirrel Search Algorithm)에 기반하여 튜닝된 하이퍼 파라미터에 따라 사전 학습된 YOLO(You Only Look once) 모델을 로딩하는 단계;

상기 적어도 하나의 프로세서에 의해, 객체 검출을 위한 이미지를 입력받는 단계; 및

상기 적어도 하나의 프로세서에 의해, 상기 이미지를 상기 YOLO 모델에 입력하여 상기 YOLO 모델이 출력하는 상기 이미지가 포함하는 객체에 대한 정보를 제공하는 단계

를 포함하고,

상기 YOLO 모델은 상기 SSA에서 사용되는 다람쥐의 위치에 상기 하이퍼 파라미터를 대응시켜 상기 하이퍼 파라미터에 대한 적합도 평가를 통해 선택된 하이퍼 파라미터에 따라 사전 학습되고,

상기 적합도 평가는 정밀도(precision), 재현율(recall), 평균 정밀도의 합에서 손실 지표를 빼는 적합도 함수에 따른 적합도에 기반하고,

상기 평균 정밀도는 IoU(Intersection over Union)가 0.5 이상인 경우를 기준으로 계산된 제1 평균 정밀도, 및 IoU가 0.5에서 0.95까지 0.05 간격으로 증가하는 여러 임계값에서 계산된 제2 평균 정밀도를 포함하고,

상기 손실 지표는, 경계 상자의 위치 및 크기 예측에서 발생하는 오류를 나타내는 경계 상자 손실(Bounding Box Loss), 상기 YOLO 모델이 각 객체의 클래스 레이블을 예측할 때 발생하는 오류를 나타내는 분류 손실(Classification Loss), 및 객체의 형태 변형을 처리하는 데 사용되는 손실을 나타내는 변형 손실(Deformation Loss)을 포함하는 것

을 특징으로 하는 객체 검출 방법.

청구항 2

제1항에 있어서,

상기 YOLO 모델은 CloU(Complete Intersection over Union) 손실 함수, DFL(Distance Focal Loss) 손실 함수, 및 이미지에 대한 시맨틱 분할 마스크를 예측하는 시맨틱 분할 모델을 포함하는 것을 특징으로 하는 객체 검출 방법.

청구항 3

삭제

청구항 4

제1항에 있어서,

상기 하이퍼 파라미터는 학습률(learning rate), 배치 크기(batch size), 및 에포크(epoch) 중 적어도 하나를 포함하는 것을 특징으로 하는 객체 검출 방법.

청구항 5

삭제

청구항 6

삭제

청구항 7

삭제

청구항 8

적어도 하나의 컴퓨터 장치에 의해 구현되는 학습 시스템의 학습 방법에 있어서,

상기 적어도 하나의 컴퓨터 장치는 적어도 하나의 프로세서를 포함하고,

상기 학습 방법은,

상기 적어도 하나의 프로세서에 의해, 복수의 학습 이미지를 포함하는 데이터 셋을 전처리하는 단계;

상기 적어도 하나의 프로세서에 의해, SSA(Squirrel Search Algorithm)에 기반하여 YOLO(You Only Look once) 모델의 하이퍼 파라미터를 튜닝하는 단계; 및

상기 전처리된 데이터 셋을 이용하여 상기 튜닝된 하이퍼 파라미터에 따라 상기 YOLO 모델을 학습시키는 단계를 포함하고,

상기 하이퍼 파라미터를 튜닝하는 단계는,

상기 SSA에서 사용되는 다람쥐의 위치에 상기 하이퍼 파라미터를 대응시켜 상기 하이퍼 파라미터에 대한 적합도 평가를 통해 하이퍼 파라미터를 선택하고,

상기 YOLO 모델을 학습시키는 단계는,

상기 선택된 하이퍼 파라미터에 따라 상기 YOLO 모델을 학습시키고,

상기 적합도 평가는 정밀도(precision), 재현율(recall), 평균 정밀도의 합에서 손실 지표를 빼는 적합도 함수에 따른 적합도에 기반하고,

상기 평균 정밀도는 IoU(Intersection over Union)가 0.5 이상인 경우를 기준으로 계산된 제1 평균 정밀도, 및 IoU가 0.5에서 0.95까지 0.05 간격으로 증가하는 여러 임계값에서 계산된 제2 평균 정밀도를 포함하고,

상기 손실 지표는, 경계 상자의 위치 및 크기 예측에서 발생하는 오류를 나타내는 경계 상자 손실(Bounding Box Loss), 상기 YOLO 모델이 각 객체의 클래스 레이블을 예측할 때 발생하는 오류를 나타내는 분류 손실(Classification Loss), 및 객체의 형태 변형을 처리하는 데 사용되는 손실을 나타내는 변형 손실(Deformation Loss)을 포함하는 것

을 특징으로 하는 학습 방법.

청구항 9

제8항에 있어서,

상기 YOLO 모델은 CloU(Complete Intersection over Union) 손실 함수, DFL(Distance Focal Loss) 손실 함수, 및 이미지에 대한 시맨틱 분할 마스크를 예측하는 시맨틱 분할 모델을 포함하는 것을 특징으로 하는 학습 방법.

청구항 10

삭제

청구항 11

제8항에 있어서,

상기 하이퍼 파라미터는 학습률(learning rate), 배치 크기(batch size), 및 에포크(epoch) 중 적어도 하나를 포함하는 것을 특징으로 하는 학습 방법.

청구항 12

삭제

청구항 13

삭제

청구항 14

삭제

청구항 15

컴퓨터 장치와 결합되어 제1항, 제2항, 제4항, 제8항, 제9항, 또는 제11항 중 어느 한 항의 방법을 컴퓨터 장치에 실행시키기 위해 컴퓨터 판독 가능한 기록매체에 저장된 컴퓨터 프로그램.

청구항 16

제1항, 제2항, 제4항, 제8항, 제9항, 또는 제11항 중 어느 한 항의 방법을 컴퓨터 장치에 실행시키기 위한 컴퓨터 프로그램이 기록되어 있는 컴퓨터 판독 가능한 기록매체.

청구항 17

적어도 하나의 컴퓨터 장치에 의해 구현되는 객체 검출 시스템에 있어서,

상기 적어도 하나의 컴퓨터 장치는 적어도 하나의 프로세서를 포함하고,

상기 적어도 하나의 프로세서에 의해,

SSA(Squirrel Search Algorithm)에 기반하여 튜닝된 하이퍼 파라미터에 따라 사전 학습된 YOLO(You Only Look once) 모델을 로딩하고,

객체 검출을 위한 이미지를 입력받고,

상기 이미지를 상기 YOLO 모델에 입력하여 상기 YOLO 모델이 출력하는 상기 이미지가 포함하는 객체에 대한 정보를 제공하고,

상기 YOLO 모델은 상기 SSA에서 사용되는 다람쥐의 위치에 상기 하이퍼 파라미터를 대응시켜 상기 하이퍼 파라미터에 대한 적합도 평가를 통해 선택된 하이퍼 파라미터에 따라 사전 학습되고,

상기 적합도 평가는 정밀도(precision), 재현율(recall), 평균 정밀도의 합에서 손실 지표를 빼는 적합도 함수에 따른 적합도에 기반하고,

상기 평균 정밀도는 IoU(Intersection over Union)가 0.5 이상인 경우를 기준으로 계산된 제1 평균 정밀도, 및 IoU가 0.5에서 0.95까지 0.05 간격으로 증가하는 여러 임계값에서 계산된 제2 평균 정밀도를 포함하고,

상기 손실 지표는, 경계 상자의 위치 및 크기 예측에서 발생하는 오류를 나타내는 경계 상자 손실(Bounding Box Loss), 상기 YOLO 모델이 각 객체의 클래스 레이블을 예측할 때 발생하는 오류를 나타내는 분류 손실(Classification Loss), 및 객체의 형태 변형을 처리하는 데 사용되는 손실을 나타내는 변형 손실(Deformation Loss)을 포함하는 것

을 특징으로 하는 객체 검출 시스템.

청구항 18

삭제

청구항 19

적어도 하나의 컴퓨터 장치에 의해 구현되는 학습 시스템에 있어서,

상기 적어도 하나의 컴퓨터 장치는 적어도 하나의 프로세서를 포함하고,

상기 적어도 하나의 프로세서에 의해,
 복수의 학습 이미지를 포함하는 데이터 셋을 전처리하고,
 SSA(Squirrel Search Algorithm)에 기반하여 YOLO(You Only Look once) 모델의 하이퍼 파라미터를 튜닝하고,
 상기 전처리된 데이터 셋을 이용하여 상기 튜닝된 하이퍼 파라미터에 따라 상기 YOLO 모델을 학습시키고,
 상기 하이퍼 파라미터를 튜닝하기 위해, 상기 적어도 하나의 프로세서에 의해,
 상기 SSA에서 사용되는 다람쥐의 위치에 상기 하이퍼 파라미터를 대응시켜 상기 하이퍼 파라미터에 대한 적합도 평가를 통해 하이퍼 파라미터를 선택하고,
 상기 YOLO 모델을 학습시키기 위해, 상기 적어도 하나의 프로세서에 의해,
 상기 선택된 하이퍼 파라미터에 따라 상기 YOLO 모델을 학습시키고,
 상기 적합도 평가는 정밀도(precision), 재현율(recall), 평균 정밀도의 합에서 손실 지표를 빼는 적합도 함수에 따른 적합도에 기반하고,
 상기 평균 정밀도는 IoU(Intersection over Union)가 0.5 이상인 경우를 기준으로 계산된 제1 평균 정밀도, 및 IoU가 0.5에서 0.95까지 0.05 간격으로 증가하는 여러 임계값에서 계산된 제2 평균 정밀도를 포함하고,
 상기 손실 지표는, 경계 상자의 위치 및 크기 예측에서 발생하는 오류를 나타내는 경계 상자 손실(Bounding Box Loss), 상기 YOLO 모델이 각 객체의 클래스 레이블을 예측할 때 발생하는 오류를 나타내는 분류 손실(Classification Loss), 및 객체의 형태 변형을 처리하는 데 사용되는 손실을 나타내는 변형 손실(Deformation Loss)을 포함하는 것
 을 특징으로 하는 학습 시스템.

청구항 20

삭제

발명의 설명

기술 분야

[0001] 본 발명은 교육부와 한국연구재단의 재원으로 지원을 받아 수행된 3단계 산학협력 선도대학 육성사업(LINC 3.0)의 연구결과이다.

[0002] 아래의 설명은 원격 탐지 영상의 객체 검출 방법 및 시스템에 관한 것이다.

배경 기술

[0003] 원격 탐지 영상 분석에서 객체 탐지는 중요한 작업 중 하나로, 다양한 산업에서 활용되고 있다. 특히 위성 영상, 감시 카메라, 드론 촬영 등에서 매우 유용하게 사용된다. 이러한 객체 탐지 기술은 관심 객체가 속한 범주를 식별해야 할 뿐만 아니라 경계 상자(Bounding Box)를 사용해 객체의 위치를 찾아야 하므로 작업의 난이도가 어려워지고 알고리즘의 요구 사항도 더 복잡해진다. 현재 객체 감지 분야에서 가장 널리 사용되는 모델 중 하나는 YOLO(You Only Look once) 시리즈이다. 이 모델은 실시간으로 이미지 내의 여러 객체를 한 번에 탐지할 수 있는 장점이 있지만, 성능은 주로 매개변수 설정에 크게 좌우된다. 하이퍼 파라미터 최적화(Hyperparameter Optimization)는 모델에 대한 깊은 이해와 충분한 시간, 계산 자원을 필요로 하는 기술이다. 모든 가능한 하이퍼 파라미터를 비교하려면 많은 계산 시간과 비용이 소요될 뿐만 아니라 최종 모델도 과도하게 복잡해질 수 있다. 따라서 실무에서 제한된 자원과 시간 내에서 최적의 하이퍼 파라미터를 효율적으로 찾기 위해 다양한 알고리즘이 개발되었다.

[0004] 메타 휴리스틱 알고리즘(Meta-heuristic)은 자연 현상을 모방하여 복잡한 최적화 문제를 해결하는 데 널리 사용된다. 각기 다른 탐색 전략을 통해 최적해를 찾는데, PSO(Particle Swarm Optimization), ABC(Artificial Bee Colony) BA(Bat Algorithm)와 FF(Firefly Algorithm) 등이 있다. 그러나 이들 알고리즘은 탐색 다양성 부족이나 성능 제한 등의 한계가 있다. 이와 같은 한계점들을 보완하기 위해 SSA(Squirrel Search Algorithm)는 여러 단계의 전략과 무작위 이동, 계절적 조건 등을 도입하여 탐색 능력을 강화한다. 특히 다양한 환경 조건에 적응

하여 지역 최적해에 빠지지 않고 더 넓은 공간을 효율적으로 탐색할 수 있다. 이러한 접근 방식은 글로벌 최적해를 찾아내어 모델의 성능을 극대화할 수 있다.

[0005] 객체 인식을 위한 하이퍼 파라미터 최적화 방법에 대한 연구는 지속적으로 이루어지고 있다. 예를 들어, 유전 알고리즘(GA)을 기반으로 한 YOLOv5 최적화 연구에서는 크로스오버와 돌연변이 연산자를 통해 새로운 개체를 생성하여 적합도 함수를 극대화하는 최적의 하이퍼 파라미터 조합을 찾는다. 특히 돌연변이는 80%의 확률로 적용되어 최적화를 돕는다. 이 최적화 결과 mAP와 재현율은 향상되었으나 정밀도는 다소 낮은 결과를 보였다. "Real-Time Flying Object Detection with YOLOv8" 연구에서는 W&B 플랫폼을 활용한 베이지안 최적화 기법을 통해 하이퍼 파라미터를 효과적으로 조정하는 방법을 제안한다. 이 기법은 학습률, 배치 크기, 이미지 크기 등을 최적화하여 mAP50 성능을 목표로 하며, YOLOv8의 감지 정확도를 크게 향상시킨다.

[0006] [선행문헌번호]

[0007] 한국등록특허 제10-2462966호

발명의 내용

해결하려는 과제

[0008] 원격 탐지 영상의 객체 검출 방법 및 시스템을 제공한다.

과제의 해결 수단

[0009] 적어도 하나의 컴퓨터 장치에 의해 구현되는 객체 검출 시스템의 객체 검출 방법에 있어서, 상기 적어도 하나의 컴퓨터 장치는 적어도 하나의 프로세서를 포함하고, 상기 객체 검출 방법은, 상기 적어도 하나의 프로세서에 의해, SSA(Squirrel Search Algorithm)에 기반하여 튜닝된 하이퍼 파라미터에 따라 사전 학습된 YOLO(You Only Look once) 모델을 로딩하는 단계; 상기 적어도 하나의 프로세서에 의해, 객체 검출을 위한 이미지를 입력받는 단계; 및 상기 적어도 하나의 프로세서에 의해, 상기 이미지를 상기 YOLO 모델에 입력하여 상기 YOLO 모델이 출력하는 상기 이미지가 포함하는 객체에 대한 정보를 제공하는 단계를 포함하는 것을 특징으로 하는 객체 검출 방법을 제공한다.

[0010] 일측에 따르면, 상기 YOLO 모델은 CloU(Complete Intersection over Union) 손실 함수, DFL(Distance Focal Loss) 손실 함수, 및 이미지에 대한 시맨틱 분할 마스크를 예측하는 시맨틱 분할 모델을 포함하는 것을 특징으로 할 수 있다.

[0011] 다른 측면에 따르면, 상기 YOLO 모델은 상기 SSA에서 사용되는 다람쥐의 위치에 상기 하이퍼 파라미터를 대응시켜 상기 하이퍼 파라미터에 대한 적합도 평가를 통해 선택된 하이퍼 파라미터에 따라 사전 학습되는 것을 특징으로 할 수 있다.

[0012] 또 다른 측면에 따르면, 상기 하이퍼 파라미터는 학습률(learning rate), 배치 크기(batch size), 및 에포크(epoch) 중 적어도 하나를 포함하는 것을 특징으로 할 수 있다.

[0013] 또 다른 측면에 따르면, 상기 적합도 평가는 적합도 함수에 따른 적합도에 기반하고, 상기 적합도 함수는 정밀도(precision), 재현율(recall), 평균 정밀도, 및 손실 지표를 향으로 포함하는 것을 특징으로 할 수 있다.

[0014] 또 다른 측면에 따르면, 상기 손실 지표는, 경계 상자의 위치 및 크기 예측에서 발생하는 오류를 나타내는 경계 상자 손실(Bounding Box Loss), 상기 YOLO 모델이 각 객체의 클래스 레이블을 예측할 때 발생하는 오류를 나타내는 분류 손실(Classification Loss), 및 객체의 형태 변형을 처리하는 데 사용되는 손실을 나타내는 변형 손실(Deformation Loss) 중 적어도 하나를 포함하는 것을 특징으로 할 수 있다.

[0015] 또 다른 측면에 따르면, 상기 평균 정밀도는 IoU(Intersection over Union)가 0.5 이상인 경우를 기준으로 계산된 제1 평균 정밀도, 및 IoU가 0.5에서 0.95까지 0.05 간격으로 증가하는 여러 임계값에서 계산된 제2 평균 정밀도 중 적어도 하나를 포함하는 것을 특징으로 할 수 있다.

[0016] 적어도 하나의 컴퓨터 장치에 의해 구현되는 학습 시스템의 학습 방법에 있어서, 상기 적어도 하나의 컴퓨터 장치는 적어도 하나의 프로세서를 포함하고, 상기 학습 방법은, 상기 적어도 하나의 프로세서에 의해, 복수의 학습 이미지를 포함하는 데이터 셋을 전처리하는 단계; 상기 적어도 하나의 프로세서에 의해, SSA(Squirrel Search Algorithm)에 기반하여 YOLO(You Only Look once) 모델의 하이퍼 파라미터를 튜닝하는 단계; 및 상기

전처리된 데이터 셋을 이용하여 상기 튜닝된 하이퍼 파라미터에 따라 상기 YOLO 모델을 학습시키는 단계를 포함하는 것을 특징으로 하는 학습 방법을 제공한다.

- [0017] 일측에 따르면, 상기 YOLO 모델은 CloU(Complete Intersection over Union) 손실 함수, DFL(Distance Focal Loss) 손실 함수, 및 이미지에 대한 시맨틱 분할 마스크를 예측하는 시맨틱 분할 모델을 포함하는 것을 특징으로 할 수 있다.
- [0018] 다른 측면에 따르면, 상기 하이퍼 파라미터를 튜닝하는 단계는, 상기 SSA에서 사용되는 다람쥐의 위치에 상기 하이퍼 파라미터를 대응시켜 상기 하이퍼 파라미터에 대한 적합도 평가를 통해 하이퍼 파라미터를 선택하고, 상기 YOLO 모델을 학습시키는 단계는, 상기 선택된 하이퍼 파라미터에 따라 상기 YOLO 모델을 학습시키는 것을 특징으로 할 수 있다.
- [0019] 또 다른 측면에 따르면, 상기 하이퍼 파라미터는 학습률(learning rate), 배치 크기(batch size), 및 에포크(epoch) 중 적어도 하나를 포함하는 것을 특징으로 할 수 있다.
- [0020] 또 다른 측면에 따르면, 상기 적합도 평가는 적합도 함수에 따른 적합도에 기반하고, 상기 적합도 함수는 정밀도(precision), 재현율(recall), 평균 정밀도, 및 손실 지표를 향으로 포함하는 것을 특징으로 할 수 있다.
- [0021] 또 다른 측면에 따르면, 상기 손실 지표는, 경계 상자의 위치 및 크기 예측에서 발생하는 오류를 나타내는 경계 상자 손실(Bounding Box Loss), 상기 YOLO 모델이 각 객체의 클래스 레이블을 예측할 때 발생하는 오류를 나타내는 분류 손실(Classification Loss), 및 객체의 형태 변형을 처리하는 데 사용되는 손실을 나타내는 변형 손실(Deformation Loss) 중 적어도 하나를 포함하는 것을 특징으로 할 수 있다.
- [0022] 또 다른 측면에 따르면, 상기 평균 정밀도는 IoU(Intersection over Union)가 0.5 이상인 경우를 기준으로 계산된 제1 평균 정밀도, 및 IoU가 0.5에서 0.95까지 0.05 간격으로 증가하는 여러 임계값에서 계산된 제2 평균 정밀도 중 적어도 하나를 포함하는 것을 특징으로 할 수 있다.
- [0023] 컴퓨터 장치와 결합되어 상기 방법을 컴퓨터 장치에 실행시키기 위해 컴퓨터 판독 가능한 기록매체에 저장된 컴퓨터 프로그램을 제공한다.
- [0024] 상기 방법을 컴퓨터 장치에 실행시키기 위한 컴퓨터 프로그램이 기록되어 있는 컴퓨터 판독 가능한 기록매체를 제공한다.
- [0025] 적어도 하나의 컴퓨터 장치에 의해 구현되는 객체 검출 시스템에 있어서, 상기 적어도 하나의 컴퓨터 장치는 적어도 하나의 프로세서를 포함하고, 상기 적어도 하나의 프로세서에 의해, SSA(Squirrel Search Algorithm)에 기반하여 튜닝된 하이퍼 파라미터에 따라 사전 학습된 YOLO(You Only Look once) 모델을 로딩하고, 객체 검출을 위한 이미지를 입력받고, 상기 이미지를 상기 YOLO 모델에 입력하여 상기 YOLO 모델이 출력하는 상기 이미지가 포함하는 객체에 대한 정보를 제공하는 것을 특징으로 하는 객체 검출 시스템을 제공한다.
- [0026] 적어도 하나의 컴퓨터 장치에 의해 구현되는 학습 시스템에 있어서, 상기 적어도 하나의 컴퓨터 장치는 적어도 하나의 프로세서를 포함하고, 상기 적어도 하나의 프로세서에 의해, 복수의 학습 이미지를 포함하는 데이터 셋을 전처리하고, SSA(Squirrel Search Algorithm)에 기반하여 YOLO(You Only Look once) 모델의 하이퍼 파라미터를 튜닝하고, 상기 전처리된 데이터 셋을 이용하여 상기 튜닝된 하이퍼 파라미터에 따라 상기 YOLO 모델을 학습시키는 것을 특징으로 하는 학습 시스템을 제공한다.

발명의 효과

- [0027] 원격 탐지 영상의 객체 검출 방법 및 시스템을 제공할 수 있다.

도면의 간단한 설명

- [0028] 도 1은 본 발명의 일실시예에 있어서, YOLO 모델의 최적화를 위한 시스템 및 최적화된 YOLO 모델을 이용한 객체 검출을 위한 시스템의 개괄적인 모습의 예를 도시한 도면이다.
- 도 2는 본 발명의 일실시예에 있어서, YOLO 모델의 최적화 과정의 예를 도시한 흐름도이다.
- 도 3은 본 발명의 일실시예에 있어서, DIOR 데이터 셋이 포함하는 20개의 객체 클래스의 예를 도시한 도면이다.
- 도 4는 본 발명의 일실시예에 있어서, 학습 및 검증용 데이터 셋의 예를 도시한 도면이다.

도 5는 본 발명의 일실시예에 있어서, 테스트용 데이터 셋의 예를 도시한 도면이다.

도 6은 본 발명의 일실시예에 있어서, 원격 탐사 객체 탐지의 예측 결과의 예를 도시한 도면이다.

도 7 내지 도 11은 본 발명의 일실시예에 있어서, 모델 학습의 결과의 예를 도시한 도면들이다.

도 12는 본 발명의 일실시예에 따른 학습 시스템의 학습 방법의 예를 도시한 흐름도이다.

도 13은 본 발명의 일실시예에 따른 객체 검출 시스템의 객체 검출 방법의 예를 도시한 흐름도이다.

도 14는 본 발명의 일실시예에 따른 컴퓨터 장치의 예를 도시한 블록도이다.

발명을 실시하기 위한 구체적인 내용

- [0029] 이하, 실시예를 첨부한 도면을 참조하여 상세히 설명한다.
- [0030] 본 발명의 실시예들에서는 자연에서 영감을 받은 메타휴리스틱 최적화 기법인 SSA(Squirrel Search Algorithm)를 적용하여 YOLO(You Only Look once) 모델(일례로, YOLOv8 모델)의 하이퍼 파라미터 최적화를 수행할 수 있다. SSA를 사용한 최적화는 다양한 장면에서 보다 정확한 객체 감지를 가능하게 하며, 과정과 그 결과를 상세히 분석하고 이를 통해 원격 객체 탐지의 성능을 향상시킬 수 있다.
- [0031] YOLOv8 네트워크의 구조
- [0032] 앞서 설명한 바와 같이 본 발명의 실시예들에서는 YOLOv8 모델의 하이퍼 파라미터 최적화를 위해 SSA를 사용하여 YOLOv8의 성능을 향상시킬 수 있다.
- [0033] 도 1은 본 발명의 일실시예에 있어서, YOLO 모델의 최적화를 위한 시스템 및 최적화된 YOLO 모델을 이용한 객체 검출을 위한 시스템의 개괄적인 모습의 예를 도시한 도면이다. 도 1의 실시예에서는 학습 시스템(110), 및 객체 검출 시스템(120)을 나타내고 있다. 학습 시스템(110), 및 객체 검출 시스템(120) 각각은 적어도 하나의 컴퓨터 장치에 의해 구현될 수 있다.
- [0034] 학습 시스템(110)은 데이터 전처리부(111), 파라미터 튜닝부(112), 및 모델 학습부(113)를 포함할 수 있다. 데이터 전처리부(111)는 학습 시스템(110)의 입력(130)인 학습 이미지들을 전처리할 수 있다. 일례로, 학습 이미지들을 포함하는 학습 데이터의 디렉토리 내의 주석 파일은 YOLO 모델과 호환되지 않는 형식(일례로 파스칼 VOC(Pascal Visual Object Classes) 형식)일 수 있으며, 데이터 전처리부(111)는 이러한 주석 파일의 형식을 YOLO 형식으로 변환할 수 있다. 또한, 파라미터 튜닝부(112)는 SSA를 이용하여 YOLO 모델의 하이퍼 파라미터를 튜닝할 수 있다. 모델 학습부(113)는 전처리된 학습 이미지들을 이용하여 튜닝된 하이퍼 파라미터에 따라 YOLO 모델을 학습할 수 있다. 이미 설명한 바와 같이 학습 시스템(110)은 적어도 하나의 컴퓨터 장치에 의해 구현될 수 있으며, 적어도 하나의 컴퓨터 장치는 적어도 하나의 프로세서를 포함할 수 있다. 여기서, 데이터 전처리부(111), 파라미터 튜닝부(112), 및 모델 학습부(113)는 적어도 하나의 프로세서의 기능적 표현들(functional expressions)일 수 있다.
- [0035] 객체 검출 시스템(120)은 모델 로딩부(121), 및 객체 검출부(122)를 포함할 수 있다. 모델 로딩부(121)는 학습 시스템(110)에 의해 사전 학습되어 제공되는 YOLO 모델을 객체 검출 시스템(120)을 구현하는 적어도 하나의 컴퓨터 장치의 메모리에 로딩할 수 있다. 객체 검출부(122)는 이미지를 입력(140)받을 수 있으며, 입력된 이미지를 로딩된 YOLO 모델에 입력하여 이미지에 포함된 객체를 검출할 수 있다. 객체 검출부(122)를 통해 제공되는 출력(150)은 입력(140)인 이미지에서 검출된 객체에 대한 정보를 포함할 수 있다. 이미 설명한 바와 같이 객체 검출 시스템(120)은 적어도 하나의 컴퓨터 장치에 의해 구현될 수 있으며, 적어도 하나의 컴퓨터 장치는 적어도 하나의 프로세서를 포함할 수 있다. 여기서, 모델 로딩부(121), 및 객체 검출부(122)는 적어도 하나의 프로세서의 기능적 표현들일 수 있다.
- [0036] 일례로, 데이터 전처리부(111)는 학습 데이터를 불러오기 위해 데이터 셋을 다운로드하여 데이터 파일들을 준비할 수 있으며, 그 다음 데이터 학습 결과를 평가하기 위해 전체 데이터 셋을 학습용, 테스트용, 그리고 검증용으로 재분배할 수 있다. 일실시예로, 전체 데이터 셋의 50%를 학습용으로, 40%를 테스트용으로, 그리고 10%를 검증용으로 각각 재분배할 수 있다.
- [0037] YOLO 모델은 객체 감지 요소인 위치 추정과 분류를 하나의 단계로 통합하여 처리하는 1단계 검출기이다. 대량의 이미지에 대해 위치 파악 및 분류를 동시에 수행함으로써 실시간 객체 감지에 효과적이지만 다양한 크기의 객체와 복잡한 장면을 처리하는데 어려움이 있다. YOLO의 최신 버전인 YOLOv8은 크게 백본(Backbone), 넥

(Neck), 헤드(Head) 세 부분으로 구성되어 있다. YOLOv8의 백본은 CSPDarknet53 구조를 기반으로 한다. 이 구조의 핵심은 C2f 모듈(Cross-Stage Partial Bottleneck with Two Convolutions)로, 고수준의 특징을 추출하고 문맥 정보를 결합해 탐지 정확도를 향상시키는 역할을 한다. 이 모듈은 두 개의 합성곱을 사용해 효율적으로 정보를 처리한다. 넥은 다양한 해상도의 특징을 결합하여 목표물의 위치와 크기를 정확히 파악할 수 있도록 돕는다. C2f 모듈이 이 부분에서도 활용되어, 높은 해상도의 특징과 낮은 해상도의 특징을 효과적으로 통합한다. 이를 통해 모델은 다양한 크기의 객체를 더 잘 탐지할 수 있다. 헤드는 앵커 프리(Anchor-Free) 모델로 설계되어, 객체 탐지를 위한 복잡성을 줄이고 성능을 높인다. 디커플드 헤드(Decoupled Head)를 사용하여, 객체성, 분류, 회귀 작업을 독립적으로 처리할 수 있다. 이로 인해 각 작업이 보다 정확하게 수행되며, 모델의 전반적인 정확도가 향상된다. 객체성 점수는 시그모이드 함수를 통해, 클래스 확률은 소프트맥스 함수를 통해 계산된다. 또한 YOLOv8은 CloU(Complete Intersection over Union)와 DFL(Distance Focal Loss) 손실 함수를 사용해 작은 객체 탐지 성능을 강화했으며, YOLOv8-seg라는 시맨틱 분할 모델도 포함하여, 이미지에 대한 시맨틱 분할 마스크를 예측할 수 있다. 이 모든 요소들이 결합되어 YOLOv8은 기존 버전보다 더 높은 AP와 빠른 속도를 기록하며, 더 뛰어난 성능을 보여준다.

[0038] 도 2는 본 발명의 일실시예에 있어서, YOLO 모델의 최적화 과정의 예를 도시한 흐름도이다. 일례로, 학습 시스템(110)의 파라미터 튜닝부(112)는 YOLOv8 모델의 학습률(learning rate), 배치 크기(batch size), 에포크(epoch)에 대한 초기 하이퍼 파라미터를 설정(Parameter initialization (210))할 수 있다. 이때, 파라미터 튜닝부(112)는 주어진 파라미터를 사용해 YOLOv8 모델을 학습시키고, 그 결과를 바탕으로 검증 손실을 최소화하는 목적 함수를 정의(Defining Objective Function (220))할 수 있다. 목적 함수는 모델의 성능을 평가하고, 파라미터 최적화를 위한 기준으로 사용될 수 있다.

[0039] 다람쥐 탐색 알고리즘(Squirrel Search Algorithm, SSA)

[0040] 메타휴리스틱(Meta-heuristic) 최적화 방법은 특정 문제에 한정되지 않고, 어떤 문제에 대해서도 범용적으로 대응할 수 있도록 설계된 알고리즘의 기본적인 정의이다. 자연에는 풍부한 메커니즘과 원리가 존재하기 때문에 일부 생물학적 행동이나 물리적 현상을 모방한 자연에서 영감을 받은 최적화 알고리즘이 등장하였다. SSA는 다람쥐의 포식 활동을 모델링한 메타휴리스틱 최적화 기법이다. 이 알고리즘은 날다람쥐의 계절별 움직임과 먹이 찾기 전략을 모방해 복잡한 문제에 대한 최적의 해를 탐색한다. SSA는 글로벌 서치(Global Search)와 로컬 서치(Local Search) 전략을 통해 초기 단계에서는 넓은 범위를 탐색하여 가능성 있는 새로운 영역을 찾고, 최적화가 진행되면서 후반 단계에서는 이미 성과를 보인 영역을 더 깊이 탐색하여 최적해를 정교화한다. 또한 자원 가용성에 따라 다양한 전략을 사용하는 계절 개념을 도입함으로 최적화 과정을 더욱 세밀하게 조정한다.

[0041] 먼저 수학적 모델 단순화를 위한 가정을 한다. 숲에는 n 마리의 날다람쥐가 있고, 각 날다람쥐 한 마리 당 한 그루의 나무에 있다. 각 날다람쥐는 개별적으로 먹이를 탐색하며, 동적 채집 행동을 통해 이용 가능한 자원을 최적 활용한다. 숲에는 일반나무, 참(도토리)나무, 히코리 나무의 세 종류의 나무가 있는데, 고려하는 숲 지역에는 세 개의 참나무와 한 개의 히코리 나무가 있다고 가정한다. 연구 설정에서 날다람쥐의 수(n)는 50마리이다. 전체 50그루의 나무 중에서 4개의 영양가 있는 먹이 자원 M_{fs} 이 있는데 1개의 히코리 나무와 3개의 도토리 나무가 있다. 나머지 46개의 나무에는 먹이 자원이 없다. 먹이 자원의 수는 $1 < M_{fs} < n$ 제한 조건에 따라 달라질 수 있으며 이 중 하나는 최적의 겨울 먹이 자원인 히코리 너트를 포함한다. SSA는 다른 군집 기반 알고리즘과 유사하게 날다람쥐의 초기 위치를 무작위로 설정하는 것으로 시작한다. 날다람쥐의 위치는 차원의 탐색 공간에서 벡터로 표현될 수 있다. 따라서 날다람쥐는 1차원, 2차원, 3차원 또는 초차원 탐색 공간에서 글라이딩하며 위치 벡터를 변경할 수 있다. 숲에 있는 날다람쥐의 수 n 마리의 날다람쥐(FS)가 숲에 있으며, i 번째 날다람쥐의 위치는 아래 수학적 식 1과 같이 벡터로 표현될 수 있다.

수학적 식 1

$$FS = \begin{bmatrix} FS_{1,1} & FS_{1,2} & \cdots & FS_{1,d} \\ FS_{2,1} & FS_{2,2} & \cdots & FS_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ FS_{n,1} & FS_{n,2} & & FS_{n,d} \end{bmatrix}$$

[0042]

[0043] 이때, $FS_{i,j}$ 는 i 번째 날다람쥐의 j 번째 차원에서의 위치를 나타낼 수 있다. 이때, i 번째 날다람쥐의 위치 FS_i 는 아래 수학적 식 2와 같이 표현될 수 있다.

수학적 식 2

$$FS_i = FS_L + U(0,1) * (FS_U - FS_L)$$

[0044]

[0045] 여기서, FS_L , FS_U 는 각각 i 번째 날다람쥐의 j 번째 차원에서의 상한과 하한을 나타낼 수 있고, $U(0,1)$ 은 $[0,1]$ 범위에서 균등 분포된 난수를 나타낼 수 있다.

[0046] 또한, 각 날다람쥐의 위치에 대한 사용자 정의 적합도 함수가 계산되어 다음 수학적 식 3과 같은 배열에 저장될 수 있다.

수학적 식 3

$$f = \begin{bmatrix} f_1([FS_{1,1}, FS_{1,2}, \dots, FS_{1,d}]) \\ f_1([FS_{2,1}, FS_{2,2}, \dots, FS_{2,d}]) \\ \vdots \\ f_1([FS_{n,1}, FS_{n,2}, \dots, FS_{n,d}]) \end{bmatrix}$$

[0047]

[0048] 날다람쥐의 위치에 따른 적합성 가치(fitness value)는 그 위치에서 탐색된 먹이원의 질을 나타낼 수 있다. 다시 말해, 최적의 식량 공급원(히코리 나무), 보통의 식량 공급원(도토리 나무), 먹이원이 없는 경우(보통 나무)를 의미할 수 있으며, 이로 인해 생존 확률도 달라질 수 있다.

[0049] SSA를 이용한 YOLOv8의 하이퍼 파라미터 최적화

[0050] 파라미터 튜닝부(112)는 다양한 매개변수 조합을 가진 날다람쥐 군집을 무작위로 초기화(Initialize Squirrel Population with random hyperparameters (230))할 수 있다. 각 다람쥐는 학습률, 배치 크기, 에포크 수와 같은 하이퍼 파라미터를 나타낼 수 있다. 이때, 파라미터 튜닝부(112)는 각 다람쥐의 하이퍼 파라미터 조합으로 YOLOv8 모델을 학습시키고, 모델의 성능을 평가할 수 있다. 이 과정은 적합도를 평가하는 과정(Evaluating Fitness (240))을 포함할 수 있으며, 이때 적합도 함수는 평균 정밀도(mean average precision, mAP) 값을 중심으로 설정될 수 있다. 일례로, 파라미터 튜닝부(112)는 아래 수학적 식 4와 같이 정밀도(P), 재현율(R), mAP_{50} , mAP_{50-95} 값의 합에서 손실 지표(loss metrics, L_{box} , L_{cls} , L_{dft})를 뺀 함수를 하이퍼 파라미터 최적화 성능의 평가 기준으로 적용할 수 있다. 여기서, mAP_{50} 은 IoU(Intersection over Union)가 0.5 이상인 경우를 기준으로 계산된 평균 정밀도를, mAP_{50-95} 은 IoU가 0.5에서 0.95까지 0.05 간격으로 증가하는 여러 임계값에서 계산된 평균 정밀도를 각각 의미할 수 있다. 계산 손실 지표에서 L_{box} (경계 상자 손실(Bounding Box Loss))는 경계 상자의 위치 및 크기 예측에서 발생하는 오류를 나타낼 수 있다. L_{box} 손실은 모델이 실제 객체의 경계 상자과 예측된 경계 상자 간의 차이를 최소화하기 위해 사용될 수 있으며, IoU 손실이나 변형된 형태의 손실 함수를 사용하여 계산될 수 있다. 또한, L_{cls} (분류 손실(Classification Loss))는 모델이 각 객체의 클래스 레이블을 예측할 때 발생하는 오류를 나타낼 수 있다. 이러한 L_{cls} 손실은 교차 엔트로피 손실을 사용하여 계산될 수 있으며, 모델이 객체의 정확한 클래스를 올바르게 분류할 수 있도록 학습을 도울 수 있다. 마지막으로 L_{dft} (변형 손실(Deformation Loss))는 객체의 형태 변형을 처리하는 데 사용되는 손실 요소로서, 객체의 형태가 다소 변형되거나 비정형적인 경우, 이러한 변형을 모델이 얼마나 잘 예측하는지를 평가할 수 있다.

수학식 4

$$F(\theta) = P + R + 2 * (mAP_{50} + mAP_{50-95}) - L$$

[0051]

[0052]

여기서 θ 는 하이퍼 파라미터들의 집합을 의미할 수 있다. 다시 말해, θ 는 다람쥐들의 위치(P_i)에 함수 $F(\theta)$ 는 적합도 함수 $f(P_i)$ 에 대응할 수 있다. 파라미터 튜닝부(112)는 적합도 평가 결과에 따라 다람쥐들의 위치(P_i)를 조정할 수 있다. 이 과정에서 더 나은 하이퍼 파라미터 조합을 찾기 위해 탐색 공간 내에서 위치를 지속적으로 업데이트(Update Squirrel Positions (250))할 수 있으며, 계절 변화에 따른 다람쥐의 행동을 반영하여 위치 업데이트를 최적화할 수 있다. 파라미터 튜닝부(112)는 반복적으로 하이퍼 파라미터를 최적화하며, 적합도 함수 $f(P_i)$ 가 아래 수학식 5와 같은 특정 수렴 기준을 충족할 때까지 과정을 반복(Convergence check (260))할 수 있다. 최적의 하이퍼 파라미터 조합이 발견되면, 해당 조합을 선택할 수 있다.

수학식 5

$$f(t+1) - f(t) < \epsilon$$

[0053]

[0054]

파라미터 튜닝부(112)는 수렴 조건이 만족되지 못하면 다람쥐들의 위치를 조정하는 과정(Update Squirrel Positions (250)) 및 수렴 검사의 과정(Convergence check (260))을 반복할 수 있고, 만족되면 과정이 종료될 수 있다. 파라미터 튜닝부(112)는 최적화된 하이퍼 파라미터를 사용해 YOLOv8 모델을 최종 학습시킬 수 있으며, 검증 데이터 셋을 통해 성능을 분석하고 평가(Evaluate Performance (270)) 및 Result Analysis & validation (280))할 수 있다.

[0055]

실험 및 결과

[0056]

본 실험예에서는 구글 코랩(Google Colab) 환경에서 수행되었다. 소프트웨어 구성으로는 PyTorch 2.3.1 프레임워크와 Python 3.10.12 버전이 사용되었으며, 하드웨어 구성으로는 Ubuntu 22.04.3 LTS 운영체제에서 Intel(R) Xeon(R) CPU @ 2.20GHz와 NVIDIA T4 GPU가 사용되었다. 이러한 환경 구성은 딥러닝 모델의 학습과 추론 과정에서 높은 성능과 효율성을 제공하였다. 아래 표 1은 구현 환경 구성의 예를 나타내고 있다.

표 1

Software (S/W)		Hardware (H/W)	
Platform	Google Colab	OS	Ubuntu 22.04.3 LTS
Framework	Pytorch 2.3.1	CPU	Intel(R) Xeon(R) CPU @ 2.20GHz
Language	Python 3.10.12	GPU	T4 GPU (Colab)

[0057]

[0058]

본 실험예에서는 DIOR(detection in optical remote sensing images) 데이터 셋을 이용하여 객체 감지를 수행했다. 이 데이터 셋에는 23,463개의 이미지와 192,472개의 인스턴스가 포함되어 있으며 건물, 자동차, 배, 항공기 등 총 20개의 서로 다른 객체 클래스를 포함한다. 도 3은 본 발명의 일실예에 있어서, DIOR 데이터 셋이 포함하는 20개의 객체 클래스의 예를 도시한 도면이다. DIOR 데이터 셋은 다양한 객체 클래스와 복잡한 배경까지 포함하고 있어 탐지 알고리즘을 평가하고 개발하는데 매우 유용하다. DIOR 전체 데이터 셋을 50%는 학습용, 40%는 테스트용, 그리고 10%는 검증용으로 재분배하였다. 도 4는 본 발명의 일실예에 있어서, 학습 및 검증용 데이터 셋의 예를 도시한 도면이고, 도 5는 본 발명의 일실예에 있어서, 테스트용 데이터 셋의 예를 도시한 도면이다. 또한, 도 6은 본 발명의 일실예에 있어서, 원격 탐사 객체 탐지의 예측 결과의 예를 도시한 도면이고, 아래 표 2는 YOLOv8 모델과 SSA가 추가된 YOLOv8 모델의 성능을 네 가지 지표로 비교하고 있으며, 표 3 및 표 4는 YOLOv8 모델과 SSA가 추가된 YOLOv8 모델의 손실을 비교하고 있다.

표 2

Model	mAP50	mAP50-95	Precision	Recall
YOLOv8	0.6540	0.4470	0.7650	0.5750
Proposed method	0.788	0.569	0.871	0.703

표 3

YOLOv8 + SSA						
epoch	train / box_loss	train / cls_loss	train / dfl_loss	val / box_loss	val / cls_loss	val / dfl_loss
1	2.9760	4.2680	3.0960	2.8210	3.2400	2.6230
5	1.3010	1.4980	1.2840	1.2780	1.3570	1.1660
10	1.1890	1.0420	1.0910	1.1580	1.0150	1.0720
15	1.0690	0.8640	1.0130	1.0970	0.9210	0.9520
20	0.9480	0.8310	0.9780	1.0540	0.8360	0.9950
25	0.8260	0.7320	0.9500	1.0540	0.7840	0.9760

표 4

YOLOv8						
epoch	train / box_loss	train / cls_loss	train / dfl_loss	val / box_loss	val / cls_loss	val / dfl_loss
1	3.5006	5.0211	3.6421	3.3188	3.8123	3.0856
5	1.5309	1.76300	1.5110	1.5036	1.5964	1.3714
10	1.3992	1.2256	1.2844	1.3629	1.1947	1.2618
15	1.2576	1.0159	1.1924	1.2913	1.0840	1.1205
20	1.1154	0.9765	1.1504	1.2402	0.9838	1.1698
25	0.9721	0.8615	1.1178	1.2407	0.9222	1.1487

mAP₅₀에서 YOLOv8+SSA 모델은 0.788로 YOLOv8 모델의 0.6540에 비해 10% 향상된 성능을 보인다. 이는 SSA가 추가됨으로써 모델이 50% IoU 임계값에서 객체를 더 정확하게 탐지할 수 있음을 의미한다. 또한 mAP₅₀₋₉₅에서 YOLOv8+SSA가 0.5690, YOLOv8이 0.4470으로, SSA가 다양한 IoU 임계값에서 SSA 모델의 성능을 전반적으로 개선했음을 나타낸다. 정밀도(Precision)에서 YOLOv8+SSA는 0.8710, YOLOv8은 0.7650 값이 나타나면서 SSA가 추가되었을 때 모델이 잘못된 탐지를 줄이면서 정확한 예측이 나타남을 확인할 수 있다. 재현율(Recall)에서 YOLOv8+SSA는 0.7030, YOLOv8은 0.5750으로 YOLOv8+SSA에서 더 높은 값을 기록하고 있다. 이는 SSA가 모델의 탐지 능력을 강화하고 놓친 객체를 줄였음을 의미한다. 모델의 모든 평가 지표에서 기본 YOLOv8 모델보다 성능이 향상된 것을 확인할 수 있다. 특히 YOLOv8+SSA 모델은 평균 정밀도(mAP)에서 최대 12% 개선된 것을 확인할 수 있다. 이는 SSA가 모델의 탐지 정확도를 높이는 데 효과적임을 보여준다. 도 7 내지 도 11은 본 발명의 일 실험예에 있어서, 모델 학습의 결과의 예를 도시한 도면들이다. 도 7의 그래프들의 x축은 에포크를, y축은 각 그래프의 대응하는 값을 의미할 수 있다. 도 8은 모델이 예측한 클래스와 실제 클래스 간의 관계를 시각적으로 나타내는 혼동 행렬(Confusion Matrix)의 예를 나타내고 있으며, 도 9는 모델의 정밀도(Precision)와 재현율(Recall) 사이의 관계를 시각화하여 모델의 전반적인 성능을 분석하기 위해, F1 신뢰 곡선의 예를 나타내고 있다. 도 9에서 x축은 임계값을, y축은 F1 점수를 각각 나타낼 수 있다. 도 10은 모델의 정밀도(Precision)와 그 신뢰 구간을 시각화하는 정밀도 신뢰 곡선의 예를 나타내고 있으며, 도 11은 모델의 재현율(Recall)과 그 신뢰 구간을 시각화하는 재현율 신뢰 곡선의 예를 각각 나타내고 있다.

이처럼, 본 발명의 실시예들에서는 다람쥐 탐색을 적용하여 YOLOv8 모델의 하이퍼 파라미터 최적화하기 위해 YOLO 모델을 학습하고, 최적화된 YOLO 모델을 통해 객체를 검출하는 방법 및 시스템에 대해 설명하였다. SSA는 탐색 능력이 뛰어나며, 다양한 환경 조건에 적응하여 글로벌 최적해를 찾는 데 효율적이다. 이로 인해 모델이 원격 탐지 영상 분석에서 더욱 높은 객체 탐지 성능을 발휘할 수 있었다. 실험예의 결과에서는 DIOR 데이터 셋

실험에서 SSA 기반 최적화된 YOLOv8 모델이 mAP_{50} 과 정밀도가 각각 10%, 9% 향상되는 등 성능 개선을 확인하였다. 이는 SSA가 실시간 객체 탐지의 성능을 크게 향상할 수 있음을 시사하며, 본 발명의 실시예들에 따른 SSA 기반 최적화된 YOLO 모델이 다양한 원격 탐지와 실시간 객체 감지 응용 분야에 적용될 수 있음을 의미할 수 있다.

[0064] 도 12는 본 발명의 일실시예에 따른 학습 시스템의 학습 방법의 예를 도시한 흐름도이다. 본 실시예에 따른 학습 방법은 도 1을 통해 설명한 학습 시스템(110)에 의해 수행될 수 있다. 이미 설명한 바와 같이, 학습 시스템(110)은 적어도 하나의 컴퓨터 장치에 의해 구현될 수 있다. 이때, 적어도 하나의 컴퓨터 장치가 포함하는 적어도 하나의 프로세서는 메모리가 포함하는 운영체제의 코드나 적어도 하나의 컴퓨터 프로그램의 코드에 따른 제어 명령(instruction)을 실행하도록 구현될 수 있다. 여기서, 적어도 하나의 프로세서는 적어도 하나의 컴퓨터 장치에 저장된 코드가 제공하는 제어 명령에 따라 동작하여 적어도 하나의 컴퓨터 장치에 의해 구현되는 학습 시스템(110)이 도 12의 방법이 포함하는 단계들(1210 내지 1230)을 수행하도록 학습 시스템(110)을 제어할 수 있다. 이때, 학습 시스템(110)이 포함하는 데이터 전처리부(111), 파라미터 튜닝부(112), 및 모델 학습부(113)는 적어도 하나의 프로세서가 도 12의 방법이 포함하는 단계들(1210 내지 1230)을 수행하기 위한 적어도 하나의 프로세서의 기능적 표현들일 수 있다.

[0065] 단계(1210)에서 데이터 전처리부(111)는 복수의 학습 이미지를 포함하는 데이터 셋을 전처리할 수 있다. 전처리는 앞서 설명한 바와 같이 데이터 셋의 디렉토리 내의 주석 파일의 형식을 YOLO 모델과 호환되는 형식으로 변환하는 과정과 전체 데이터 셋을 학습용, 테스트용, 그리고 검증용으로 재분배하는 과정을 포함할 수 있다.

[0066] 단계(1220)에서 파라미터 튜닝부(112)는 SSA에 기반하여 YOLO 모델의 하이퍼 파라미터를 튜닝할 수 있다. 여기서, YOLO 모델은 CIoU 손실 함수, DFL 손실 함수, 및 이미지에 대한 시멘틱 분할 마스크를 예측하는 시멘틱 분할 모델을 포함하는 모델일 수 있다. 앞서 이러한 모델의 예로서 YOLOv8 모델에 대해 설명한 바 있다. 이때, 파라미터 튜닝부(112)는 SSA에서 사용되는 다람쥐의 위치에 하이퍼 파라미터를 대응시켜 하이퍼 파라미터에 대한 적합도 평가를 통해 하이퍼 파라미터를 선택할 수 있다. 여기서, 하이퍼 파라미터는 학습률, 배치 크기, 및 에포크 중 적어도 하나를 포함할 수 있다. 한편, 적합도 평가는 적합도 함수에 따른 적합도에 기반할 수 있다. 이러한 적합도 함수는 정밀도(precision), 재현율(recall), 평균 정밀도, 및 손실 지표를 향으로 포함할 수 있다. 손실 지표는, 경계 상자의 위치 및 크기 예측에서 발생하는 오류를 나타내는 경계 상자 손실, YOLO 모델이 각 객체의 클래스 레이블을 예측할 때 발생하는 오류를 나타내는 분류 손실, 및 객체의 형태 변형을 처리하는 데 사용되는 손실을 나타내는 변형 손실 중 적어도 하나를 포함할 수 있다. 또한, 평균 정밀도는 IoU가 0.5 이상인 경우를 기준으로 계산된 제1 평균 정밀도, 및 IoU가 0.5에서 0.95까지 0.05 간격으로 증가하는 여러 임계값에서 계산된 제2 평균 정밀도 중 적어도 하나를 포함할 수 있다. 이러한 적합도 함수에 대한 예를 앞서 수학식 4를 통해 설명한 바 있다.

[0067] 단계(1230)에서 모델 학습부(113)는 전처리된 데이터 셋을 이용하여 튜닝된 하이퍼 파라미터에 따라 YOLO 모델을 학습시킬 수 있다. 이때, 모델 학습부(113)는 단계(1230)에서 파라미터 튜닝부(112)에 의해 선택된 하이퍼 파라미터에 따라 YOLO 모델을 학습시킬 수 있다.

[0068] 도 13은 본 발명의 일실시예에 따른 객체 검출 시스템의 객체 검출 방법의 예를 도시한 흐름도이다. 본 실시예에 따른 객체 검출 방법은 도 1을 통해 설명한 객체 검출 시스템(120)에 의해 수행될 수 있다. 이미 설명한 바와 같이, 객체 검출 시스템(120)은 적어도 하나의 컴퓨터 장치에 의해 구현될 수 있다. 이때, 적어도 하나의 컴퓨터 장치가 포함하는 적어도 하나의 프로세서는 메모리가 포함하는 운영체제의 코드나 적어도 하나의 컴퓨터 프로그램의 코드에 따른 제어 명령(instruction)을 실행하도록 구현될 수 있다. 여기서, 적어도 하나의 프로세서는 적어도 하나의 컴퓨터 장치에 저장된 코드가 제공하는 제어 명령에 따라 동작하여 적어도 하나의 컴퓨터 장치에 의해 구현되는 객체 검출 시스템(120)이 도 13의 방법이 포함하는 단계들(1310 내지 1330)을 수행하도록 객체 검출 시스템(120)을 제어할 수 있다. 이때, 객체 검출 시스템(120)이 포함하는 모델 로딩부(121), 및 객체 검출부(122)는 적어도 하나의 프로세서가 도 13의 방법이 포함하는 단계들(1310 내지 1330)을 수행하기 위한 적어도 하나의 프로세서의 기능적 표현들일 수 있다.

[0069] 단계(1310)에서 모델 로딩부(121)는 SSA에 기반하여 튜닝된 하이퍼 파라미터에 따라 사전 학습된 YOLO 모델을 로딩할 수 있다. 여기서, YOLO 모델은 CIoU 손실 함수, DFL 손실 함수, 및 이미지에 대한 시멘틱 분할 마스크를 예측하는 시멘틱 분할 모델을 포함하는 모델일 수 있다. 앞서 이러한 모델의 예로서 YOLOv8 모델에 대해 설명한 바 있다. 이때, YOLO 모델은 SSA에서 사용되는 다람쥐의 위치에 하이퍼 파라미터를 대응시켜 하이퍼 파라미터에 대한 적합도 평가를 통해 선택된 하이퍼 파라미터에 따라 사전 학습될 수 있다. 앞서 도 12의 단계

(1220)에서 파라미터 튜닝부(112)가 SSA에서 사용되는 다람쥐의 위치에 하이퍼 파라미터를 대응시켜 하이퍼 파라미터에 대한 적합도 평가를 통해 하이퍼 파라미터를 선택할 수 있음에 대해 설명하였다. 여기서, 하이퍼 파라미터는 학습률, 배치 크기, 및 에포크 중 적어도 하나를 포함할 수 있다. 한편, 적합도 평가는 적합도 함수에 따른 적합도에 기반할 수 있다. 이러한 적합도 함수는 정밀도(precision), 재현율(recall), 평균 정밀도, 및 손실 지표를 항으로 포함할 수 있다. 손실 지표는, 경계 상자의 위치 및 크기 예측에서 발생하는 오류를 나타내는 경계 상자 손실, YOLO 모델이 각 객체의 클래스 레이블을 예측할 때 발생하는 오류를 나타내는 분류 손실, 및 객체의 형태 변형을 처리하는 데 사용되는 손실을 나타내는 변형 손실 중 적어도 하나를 포함할 수 있다. 또한, 평균 정밀도는 IoU가 0.5 이상인 경우를 기준으로 계산된 제1 평균 정밀도, 및 IoU가 0.5에서 0.95까지 0.05 간격으로 증가하는 여러 임계값에서 계산된 제2 평균 정밀도 중 적어도 하나를 포함할 수 있다. 이러한 적합도 함수에 대한 예를 앞서 수학적식 4를 통해 설명한 바 있다.

[0070] 단계(1320)에서 객체 검출부(122)는 객체 검출을 위한 이미지를 입력받을 수 있다. 이미지는 위성 영상, 항공 영상, 감시 카메라 영상, 드론 촬영 영상 등과 같은 원격 탐지 영상을 포함할 수 있다.

[0071] 단계(1320)에서 객체 검출부(122)는 이미지를 YOLO 모델에 입력하여 YOLO 모델이 출력하는 이미지가 포함하는 객체에 대한 정보를 제공할 수 있다. 앞서, SSA를 통해 튜닝된 하이퍼 파라미터에 따라 사전 학습된 YOLO 모델의 실시간 객체 탐지 성능이 크게 향상되었음을 설명한 바 있으며, 이러한 성능 향상에 따라 객체 검출부(122)는 입력된 이미지에 대해 보다 정확한 객체 탐지 기능을 제공할 수 있다.

[0072] 이처럼, 본 발명의 실시예들에 따르면, 원격 탐지 영상의 객체 검출 방법 및 시스템을 제공할 수 있다.

[0073] 도 14는 본 발명의 일실시예에 따른 컴퓨터 장치의 예를 도시한 블록도이다. 일례로, 학습 시스템(110) 및 객체 검출 시스템(120) 각각은 적어도 하나의 컴퓨터 장치에 의해 구현될 수 있고, 적어도 하나의 컴퓨터 장치 각각이 도 14의 컴퓨터 장치(Computer device, 1400)에 대응할 수 있다. 컴퓨터 장치(1400)는 도 14에 도시된 바와 같이, 메모리(Memory, 1410), 프로세서(Processor, 1420), 통신 인터페이스(Communication interface, 1430) 그리고 입출력 인터페이스(I/O interface, 1440)를 포함할 수 있다. 메모리(1410)는 컴퓨터에서 판독 가능한 기록매체로서, RAM(random access memory), ROM(read only memory) 및 디스크 드라이브와 같은 비소멸성 대용량 기록장치(permanent mass storage device)를 포함할 수 있다. 여기서 ROM과 디스크 드라이브와 같은 비소멸성 대용량 기록장치는 메모리(1410)와는 구분되는 별도의 영구 저장 장치로서 컴퓨터 장치(1400)에 포함될 수도 있다. 또한, 메모리(1410)에는 운영체제와 적어도 하나의 프로그램 코드가 저장될 수 있다. 이러한 소프트웨어 구성요소들은 메모리(1410)와는 별도의 컴퓨터에서 판독 가능한 기록매체로부터 메모리(1410)로 로딩될 수 있다. 이러한 별도의 컴퓨터에서 판독 가능한 기록매체는 플로피 드라이브, 디스크, 테이프, DVD/CD-ROM 드라이브, 메모리 카드 등의 컴퓨터에서 판독 가능한 기록매체를 포함할 수 있다. 다른 실시예에서 소프트웨어 구성요소들은 컴퓨터에서 판독 가능한 기록 매체가 아닌 통신 인터페이스(1430)를 통해 메모리(1410)에 로딩될 수도 있다. 예를 들어, 소프트웨어 구성요소들은 네트워크(Network, 1460)를 통해 수신되는 파일들에 의해 설치되는 컴퓨터 프로그램에 기반하여 컴퓨터 장치(1400)의 메모리(1410)에 로딩될 수 있다.

[0074] 프로세서(1420)는 기본적인 산술, 로직 및 입출력 연산을 수행함으로써, 컴퓨터 프로그램의 명령을 처리하도록 구성될 수 있다. 명령은 메모리(1410) 또는 통신 인터페이스(1430)에 의해 프로세서(1420)로 제공될 수 있다. 예를 들어 프로세서(1420)는 메모리(1410)와 같은 기록 장치에 저장된 프로그램 코드에 따라 수신되는 명령을 실행하도록 구성될 수 있다.

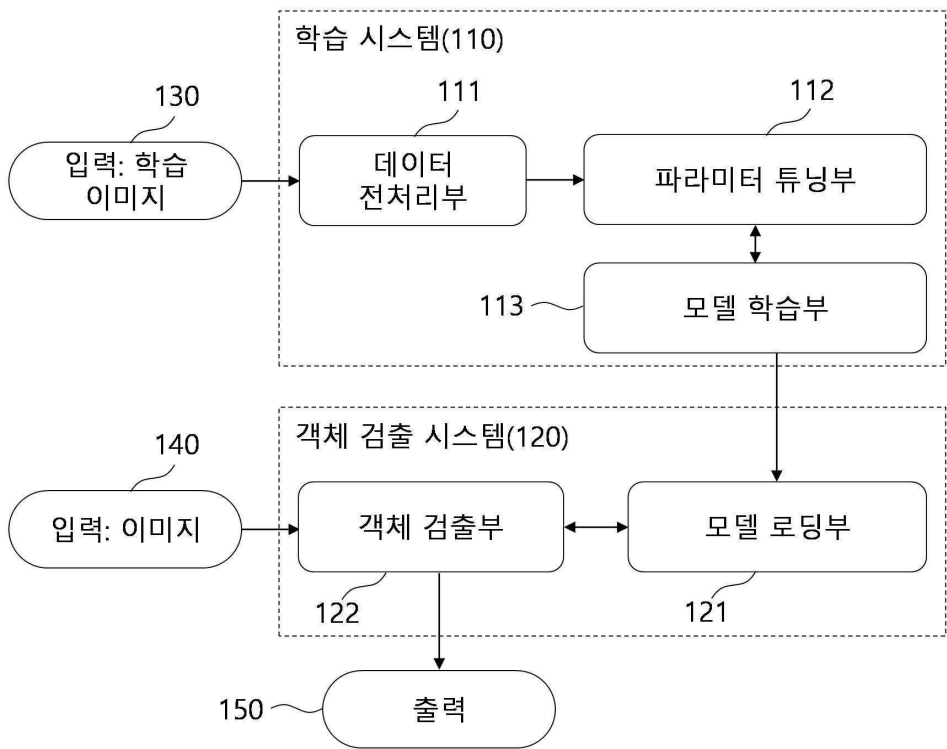
[0075] 통신 인터페이스(1430)는 네트워크(1460)를 통해 컴퓨터 장치(1400)가 다른 장치와 서로 통신하기 위한 기능을 제공할 수 있다. 일례로, 컴퓨터 장치(1400)의 프로세서(1420)가 메모리(1410)와 같은 기록 장치에 저장된 프로그램 코드에 따라 생성한 요청이나 명령, 데이터, 파일 등이 통신 인터페이스(1430)의 제어에 따라 네트워크(1460)를 통해 다른 장치들로 전달될 수 있다. 역으로, 다른 장치로부터의 신호나 명령, 데이터, 파일 등이 네트워크(1460)를 거쳐 컴퓨터 장치(1400)의 통신 인터페이스(1430)를 통해 컴퓨터 장치(1400)로 수신될 수 있다. 통신 인터페이스(1430)를 통해 수신된 신호나 명령, 데이터 등은 프로세서(1420)나 메모리(1410)로 전달될 수 있고, 파일 등은 컴퓨터 장치(1400)가 더 포함할 수 있는 저장 매체(상술한 영구 저장 장치)로 저장될 수 있다.

[0076] 입출력 인터페이스(1440)는 입출력 장치(I/O device, 1450)와의 인터페이스를 위한 수단일 수 있다. 예를 들어, 입력 장치는 마이크, 키보드 또는 마우스 등의 장치를, 그리고 출력 장치는 디스플레이, 스피커와 같은 장치를 포함할 수 있다. 다른 예로 입출력 인터페이스(1440)는 터치스크린과 같이 입력과 출력을 위한 기능이 하나로 통합된 장치와의 인터페이스를 위한 수단일 수도 있다. 입출력 장치(1450)는 컴퓨터 장치(1400)와 하나의 장치로 구성될 수도 있다.

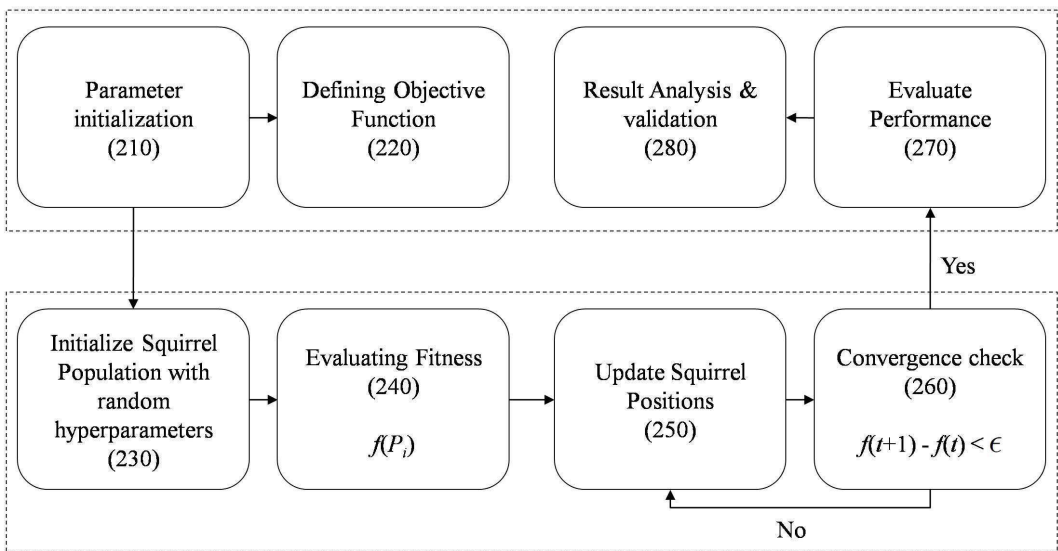
- [0077] 또한, 다른 실시예들에서 컴퓨터 장치(1400)는 도 14의 구성요소들보다 더 적은 혹은 더 많은 구성요소들을 포함할 수도 있다. 그러나, 대부분의 종래기술적 구성요소들을 명확하게 도시할 필요성은 없다. 예를 들어, 컴퓨터 장치(1400)는 상술한 입출력 장치(1450) 중 적어도 일부를 포함하도록 구현되거나 또는 트랜시버(transceiver), 데이터베이스 등과 같은 다른 구성요소들을 더 포함할 수도 있다.
- [0078] 이상에서 설명된 시스템 또는 장치는 하드웨어 구성요소, 또는 하드웨어 구성요소 및 소프트웨어 구성요소의 조합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치 및 구성요소는, 예를 들어, 프로세서, 콘트롤러, ALU(arithmetic logic unit), 디지털 신호 프로세서(digital signal processor), 마이크로컴퓨터, FPGA(field programmable gate array), PLU(programmable logic unit), 마이크로프로세서, 또는 명령(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 하나 이상의 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다. 처리 장치는 운영 체제(OS) 및 상기 운영 체제 상에서 수행되는 하나 이상의 소프트웨어 어플리케이션을 수행할 수 있다. 또한, 처리 장치는 소프트웨어의 실행에 응답하여, 데이터를 접근, 저장, 조작, 처리 및 생성할 수도 있다. 이해의 편의를 위하여, 처리 장치는 하나가 사용되는 것으로 설명된 경우도 있지만, 해당 기술분야에서 통상의 지식을 가진 자는, 처리 장치가 복수 개의 처리 요소(processing element) 및/또는 복수 유형의 처리 요소를 포함할 수 있음을 알 수 있다. 예를 들어, 처리 장치는 복수 개의 프로세서 또는 하나의 프로세서 및 하나의 콘트롤러를 포함할 수 있다. 또한, 병렬 프로세서(parallel processor)와 같은, 다른 처리 구성(processing configuration)도 가능하다.
- [0079] 소프트웨어는 컴퓨터 프로그램(computer program), 코드(code), 명령(instruction), 또는 이들 중 하나 이상의 조합을 포함할 수 있으며, 원하는 대로 동작하도록 처리 장치를 구성하거나 독립적으로 또는 결합적으로(collectively) 처리 장치를 명령할 수 있다. 소프트웨어 및/또는 데이터는, 처리 장치에 의하여 해석되거나 처리 장치에 명령 또는 데이터를 제공하기 위하여, 어떤 유형의 기계, 구성요소(component), 물리적 장치, 가상장치(virtual equipment), 컴퓨터 저장 매체 또는 장치에 구체화(embody)될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 하나 이상의 컴퓨터 판독 가능 기록매체에 저장될 수 있다.
- [0080] 실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 매체는 컴퓨터로 실행 가능한 프로그램을 계속 저장하거나, 실행 또는 다운로드를 위해 임시 저장하는 것일 수도 있다. 또한, 매체는 단일 또는 수개 하드웨어가 결합된 형태의 다양한 기록수단 또는 저장수단일 수 있는데, 어떤 컴퓨터 시스템에 직접 접속되는 매체에 한정되지 않고, 네트워크 상에 분산 존재하는 것일 수도 있다. 매체의 예시로는, 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체, CD-ROM 및 DVD와 같은 광기록 매체, 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical medium), 및 ROM, RAM, 플래시 메모리 등을 포함하여 프로그램 명령어가 저장되도록 구성된 것이 있을 수 있다. 또한, 다른 매체의 예시로, 애플리케이션을 유통하는 앱 스토어나 기타 다양한 소프트웨어를 공급 내지 유통하는 사이트, 서버 등에서 관리하는 기록매체 내지 저장매체도 들 수 있다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다.
- [0081] 이상과 같이 실시예들이 비록 한정된 실시예와 도면에 의해 설명되었으나, 해당 기술분야에서 통상의 지식을 가진 자라면 상기의 기재로부터 다양한 수정 및 변형이 가능하다. 예를 들어, 설명된 기술들이 설명된 방법과 다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성요소들이 설명된 방법과 다른 형태로 결합 또는 조합되거나, 다른 구성요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다.
- [0082] 그러므로, 다른 구현들, 다른 실시예들 및 청구범위와 균등한 것들도 후술하는 청구범위의 범위에 속한다.

도면

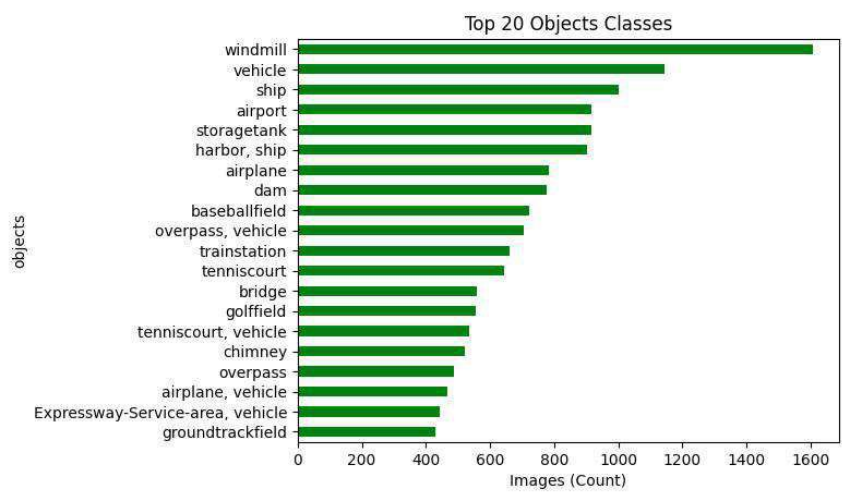
도면1



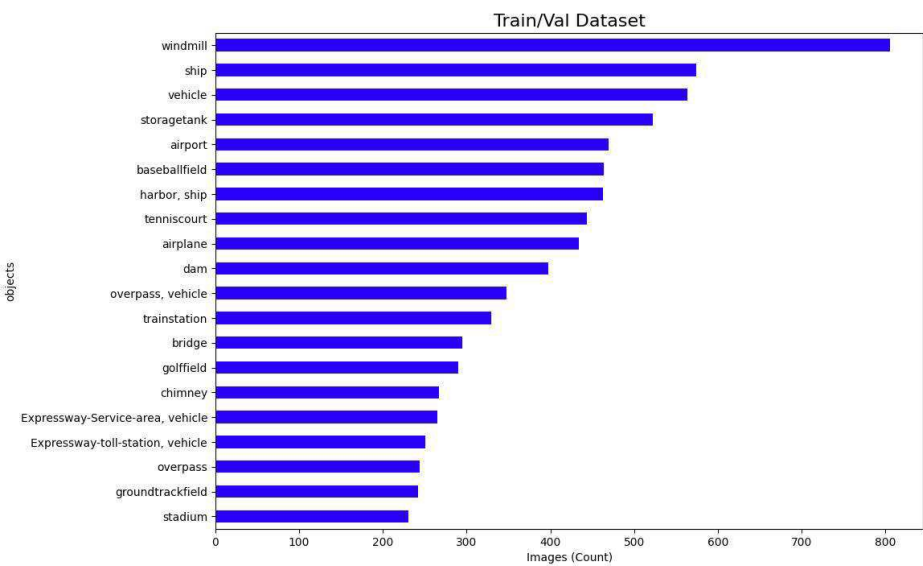
도면2



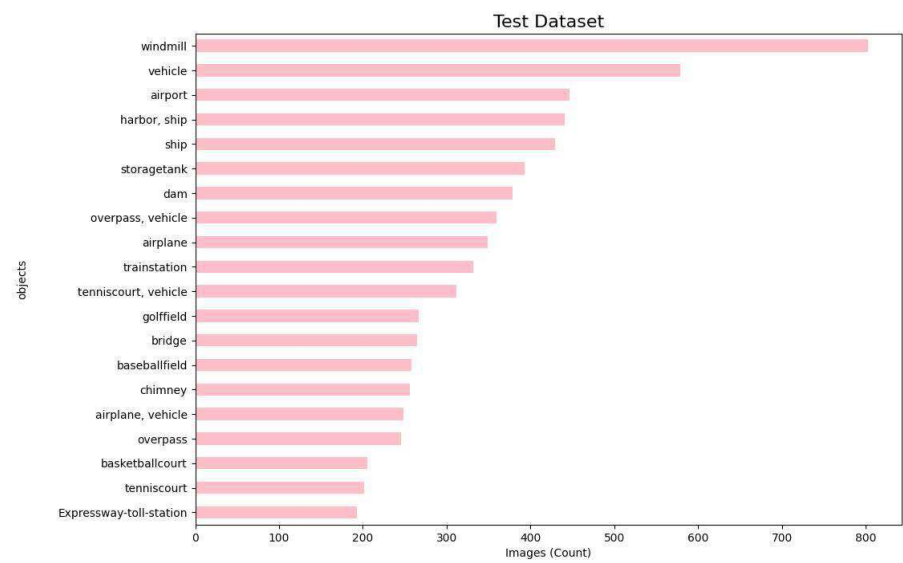
도면3



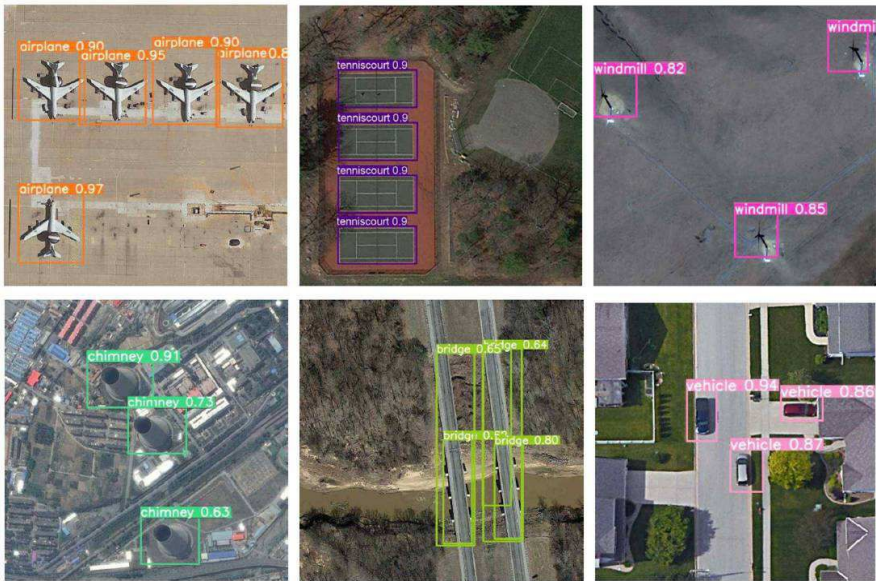
도면4



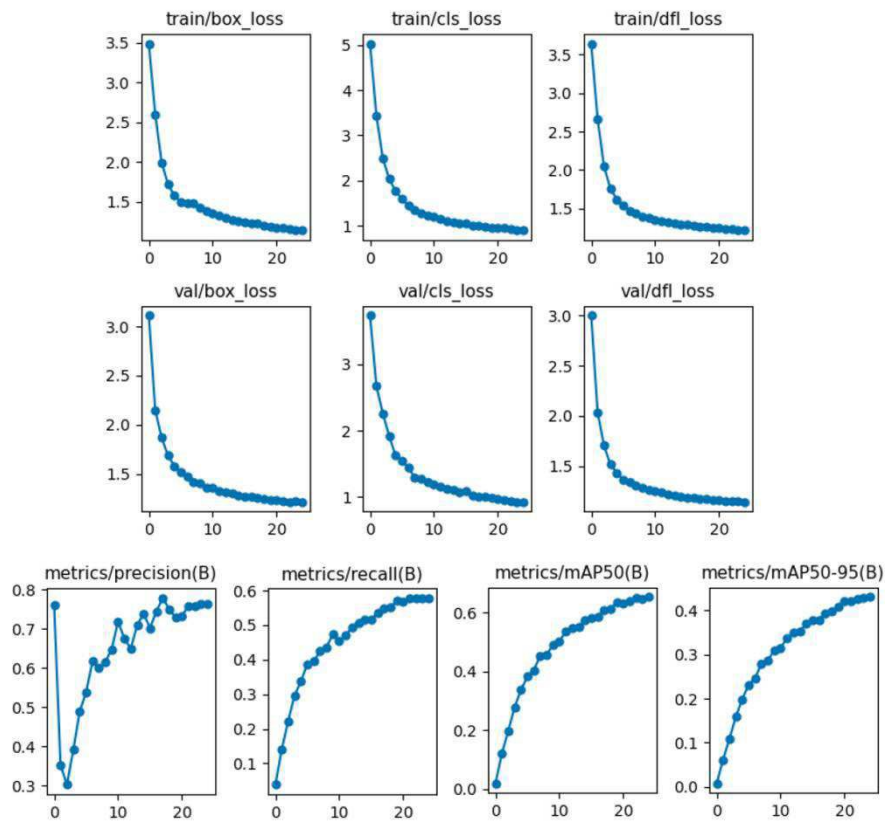
도면5



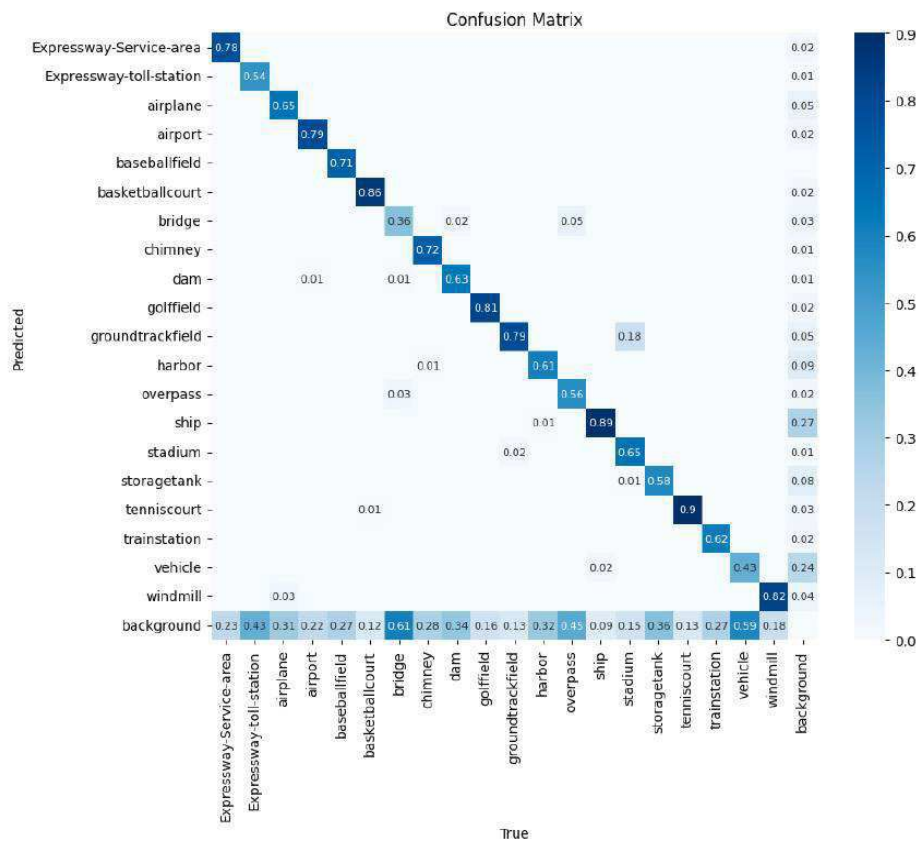
도면6



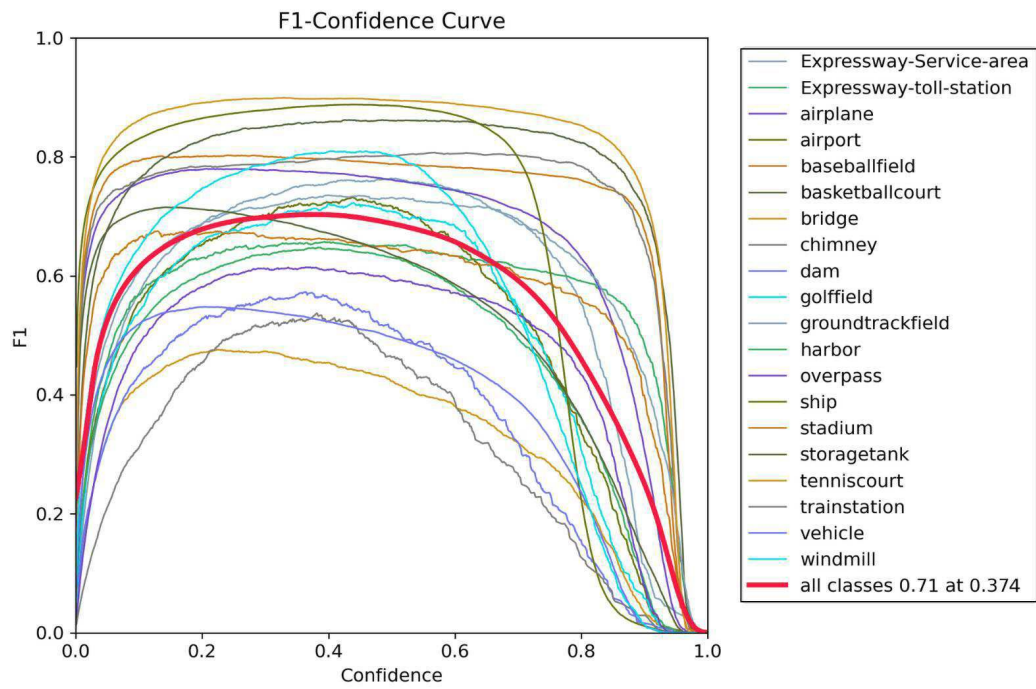
도면7



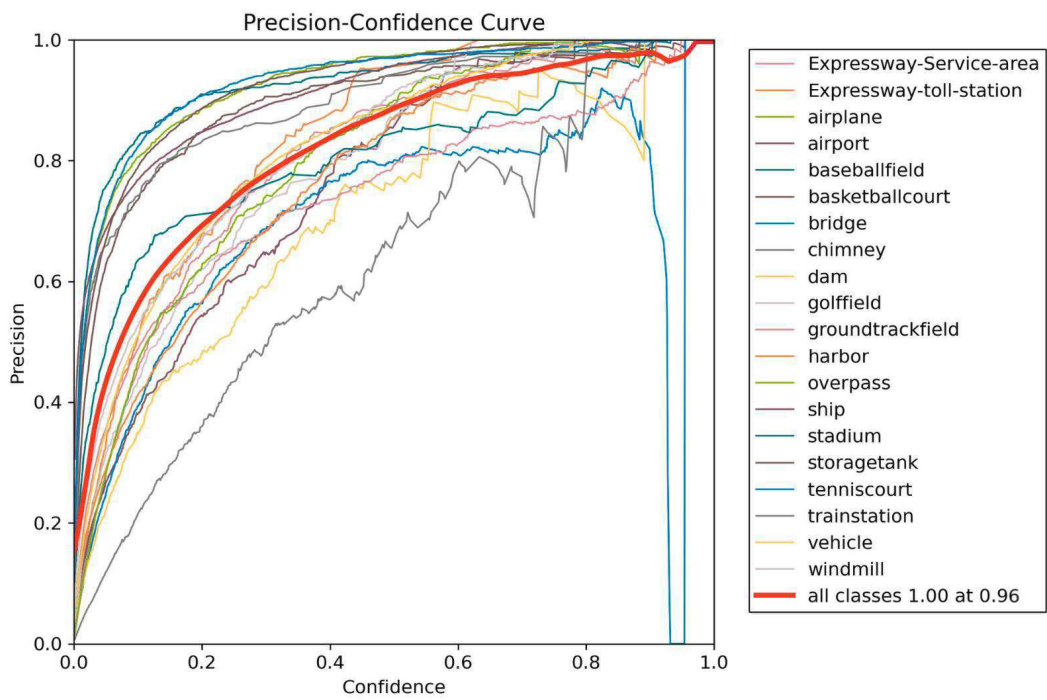
도면8



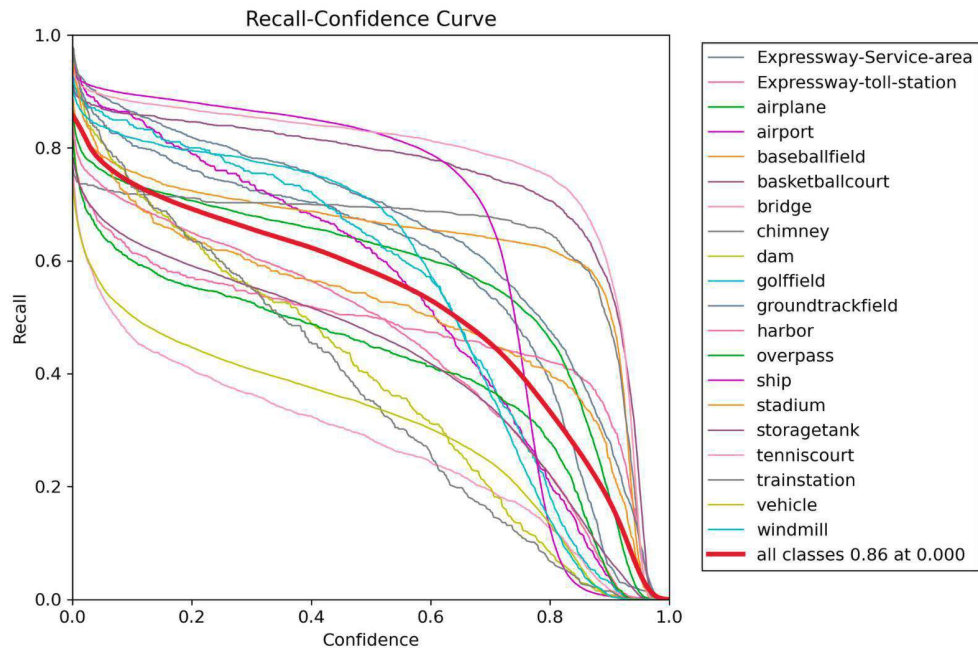
도면9



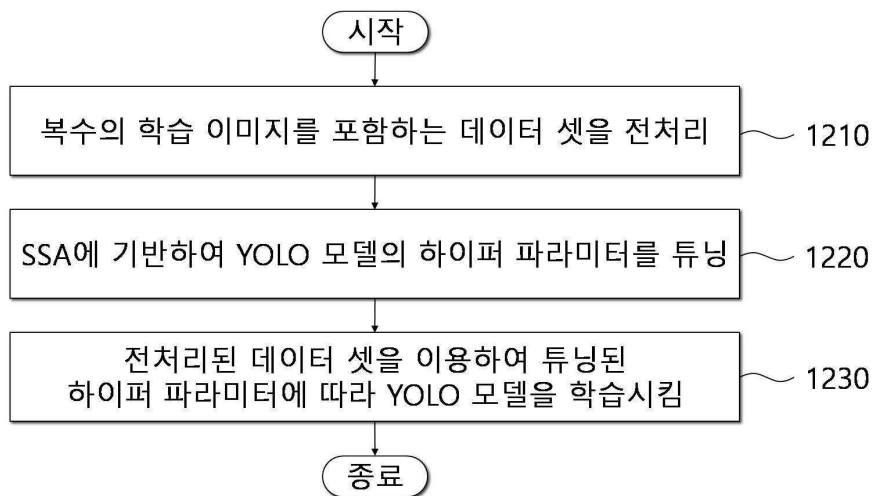
도면10



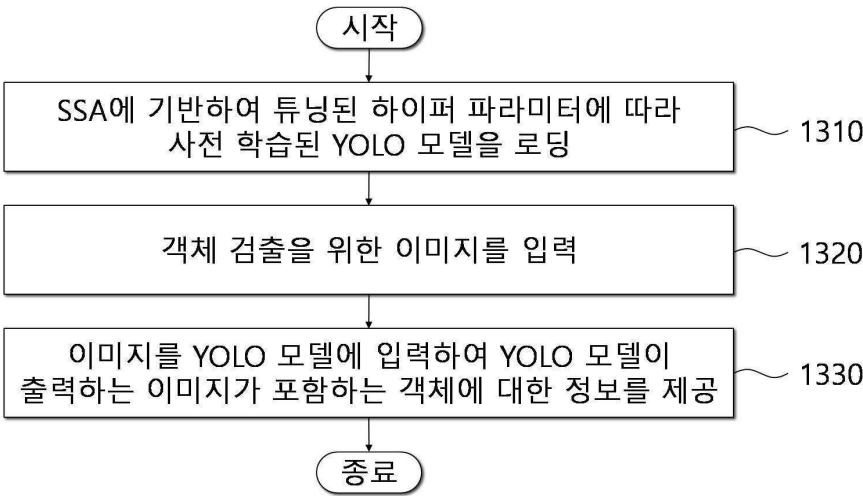
도면11



도면12



도면13



도면14

