

딥러닝 기반 특정 객체 추적 드론 시스템

임창훈, 권대현, 신수용*

금오공과대학교

gotogoin@kumoh.ac.kr, lakms123456@naver.com, wdragon@kumoh.ac.kr*

Deep Learning based Specific Object Tracking System using Drone

Chang Hun Lim, Dae Hyeon Kwon, Soo Young Shin*

Kumoh National Institute of Technology

요약

본 논문은 고정된 감시 카메라의 한계점을 보완하고 실시간으로 감시와 동시 특정 개체 발견 시 이를 추적하는 드론 시스템을 구현한다. 딥러닝 영상처리의 기법 중 객체 탐지를 목적으로 하는 YOLO(You Only Look Once)를 사용하여 추적대상으로 예상되는 이미지를 학습시킨다. 드론에서 촬영하는 실시간 영상을 기준으로 객체를 식별하고 이 추적하는 시스템 ROS(Robot Operating System)를 통해 개발한다.

I. 서론

최근 산업환경의 스마트화로 인하여 인적 자원 대신 전자 장비를 활용하여 다양한 영역에 감시 카메라를 설치하여 실시간으로 관리하고 있다. 이러한 추세에 힘입어 감시 시스템 또한 발전하고 있다.[1] 기존의 방법은 IoT 장비와 함께 카메라를 특정 지역에 설치하여 이를 기준으로 감시 반경을 지정한다. 그러나 이러한 방식은 신원불상자를 감지하거나 범죄자를 추적하는데 한계가 있다. 따라서 본 논문에서는 기존의 감시 카메라의 한계점을 보완하기 위하여 드론을 이용한다. 지정된 영역 외에도 다양한 구간에 드론이 순환 감시 체계를 ROS(Robot Operating System)를 통하여 구현한다. 추가적으로 딥러닝 영상처리 기법 중 하나 YOLO(You Only Look Once)를 사용하여 객체를 탐지한다.[2,3] 이 객체를 사람으로 한정하여 이미지를 학습시킨다. 드론을 이용한 실시간 감시 및 추적 시스템 설계하여 구현한다.

darknet의 makefile에서 GPU, CUDNN, OPENCV를 사용한다.

2.1.2 학습 데이터 생성

객체를 동영상으로 촬영하고 프레임 단위로 잘라 이미지들을 만든 뒤, data augmentation 파이썬 코드를 이용하여 이미지들의 각도를 변화시켜 이미지를 추가로 총 6,000장의 이미지를 얻는다. 각 이미지들을 라벨링할 때마다 이미지의 bounding box 좌표가 적힌 txt파일이 자동으로 생성된다. 이미지의 80%는 데이터 학습용, 나머지 20%는 테스트용으로 사용한다.

2.1.3 학습 진행 및 결과

데이터 학습은 darknet53의 모델인 darknet53.conv75 [2]를 사용하여 학습을 진행한다. 그림 1의 표를 통해 학습 중에 실시간으로 average loss값과 mAP값을 확인할 수 있다. average loss는 iteration 210 구간에서 급격하게 감소하고 1260쯤에서 0.2로 수렴한다. 가중치 파일로 객체가 있는 이미지를 테스트해 본 결과, 그림 2와 같이 이미지의 객체를 99%의 정확성으로 검출한다.

II. 본론

2.1 YOLO 데이터 학습

본 논문에서는 R-CNN(Regions with Convolutional Neuron Networks features), Faster R-CNN과 비교하여 객체 검출이 상대적으로 뛰어난 것을 보여주는 YOLOv3를 사용하여 단일 신경망을 구성한다. 드론의 객체 감지 속도를 높이고 가중치 파일로 인해 카메라의 프레임 저하를 피하기 위해 YOLOv3 중 tiny 버전을 사용한다. 사용하는 드론은 bebop으로 해당 드라이버의 호환성을 위하여 ubuntu 16.04 운영체제로 GCS(Ground Control Station)를 구성한다.

2.1.1 학습설정

본 논문에서는 데이터 학습을 설정하기 위해 cfg파일의 값을 변경한다. classes는 검출할 객체의 수와 동일하게 1로 설정하고 filters는 $18((classes + 5) * 3)$ 로 지정한다. batch_size(한 번에 학습할 이미지 양) 값은 64, subdivisions(batch_size를 나눠서 학습할 양) 값은 16, iteration(총 학습 횟수) 값은 4200으로 설정한다. 데이터 학습 속도를 높이기 위해

2.2 ROS를 활용한 객체 추적 시스템

본 논문에서 드론 제어를 위해 ROS를 활용하고 객체 감지를 위해 YOLO를 사용하고 실시간 이미지 처리를 GCS에서 수행한다. 그림 3은 전체 시스템의 순서도이다. 시작 시 GPS 고도 정보를 받아와 지표면의 고도를 측정한다. 키보드 입력을 통해 제어가 이루어지며 수동 조작과 추적 상태로 나누어진다. 추적 상태에서는 GPS 고도 정보를 받아와 기존 지표면 고도와와의 차이를 통해 고도 h보다 클 때까지 상승한다. h = 5 (약 2m)로 설정하고 이후 객체 감지가 완료되면 Bounding boxes를 이용해 객체 추적을 수행한다.

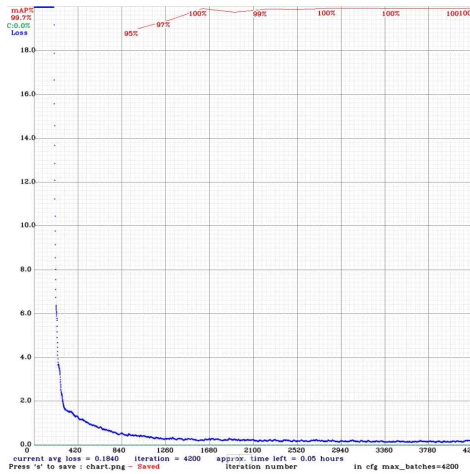


그림 1 Average loss and mAP chart

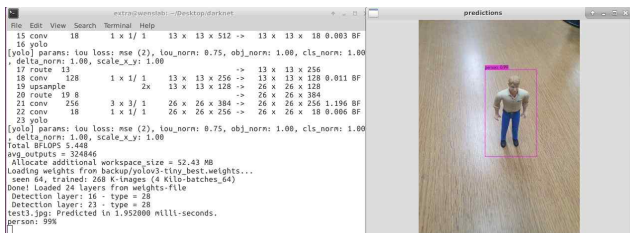


그림 2 Result of training data

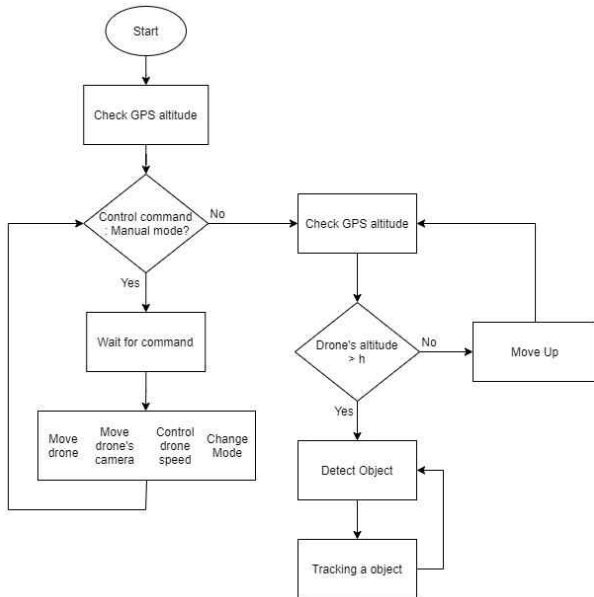


그림 3 System flowchart

2.2.1 Bounding boxes를 이용한 객체 추적

Bounding boxes는 객체 검출 박스를 말하는 것으로 박스의 x 좌표의 최솟값, 최댓값, y좌표의 최솟값, 최댓값, 검출의 정확도, 검출된 사물의 이름이 포함한다. 박스 검출 시 박스의 확률 p 보다 높을 경우에만 객체 추적이 이루어져 검출 오류에 인한 오작동을 방지한다. 이 때 p는 65%로 구성한다. 이후 박스의 너비, 높이, 박스의 x축 중점, y축 중점을 계산한다. 먼저 박스의 높이를 통해 객체와의 거리를 판단한다. 이후 박스의 너비를 통해 객체와 드론의 거리가 가까운지, 멀리 있는지를 판단하고 이에 따라 드론의 x축 움직임을 제어한다. 만약 박스의 위치가 오차 범위 m 이내일 경우 박스의 위치에 대한 제어를 실행한다. 본 논문에서 m은 ± 15 로 설정하였다. 박스의 x, y

축 중점과 카메라 프레임의 중점을 비교하여 박스가 카메라 중심에 오도록 드론을 제어한다. 카메라 프레임의 중심점과 박스의 x축 중점과의 거리를 계산해 드론의 방향을 제어하고 카메라 프레임의 중심점과 y축과의 거리를 비교해 드론의 x축 움직임을 제어한다.

III. 결론

본 논문에서는 드론을 이용하여 특정 객체를 찾아내고 이를 추적하는 시스템을 제안하고 구현한다. ROS를 사용하여 드론과 GCS 사이의 통신을 구현하고 YOLO를 사용하여 실시간 객체 감지 및 추적 시스템 구현한다. 객체와의 거리 및 빛의 세기 등과 같은 환경 변수에 따라 성능저하가 발생할 수 있다. 이러한 문제를 해결하기 위하여 향후에는 다수의 드론을 동시 동작하여 활용하는 기술을 연구할 계획이다.

ACKNOWLEDGMENT

이 논문은 2021년도 정부(교육과학기술부)의 재원으로 한국연구재단의 대학중점연구소 지원사업으로 수행된 연구임(2018R1A6A1A03024003) 본 연구는 과학기술정보통신부 및 정보통신기획평가원의 지역지능화혁신 인재양성(Grand ICT연구센터) 사업의 연구결과로 수행되었음 (IITP-2021-2020-0-01612)

참 고 문 헌

- [1] 강수경, 박명혜, 김영현, 강동훈, "IoT 복합센서 모듈 기반 전력설비 원격감시시스템 개발". 대한전기학회 학술대회 논문집, 2020
- [2] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," 2018.
- [3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.