UAV에서의 ML을 사용한 목표 탐지 및 충돌방지

박재한, 이신우, 신수용
금오공과대학교

qkrwogks7094@kumoh.ac.kr,nakgongs@naver.com wdragon@kumoh.ac.kr

Object Detection and anti-collision using ML for UAV

Jae Han Park, Shin Woo Lee, Soo Young Shin

Kumoh National Institute of Technology

요 약

최근 UAV(Unmanned Aerial Vehicle)를 활용한 무인 자동화 시스템에 대하여 많은 연구가 이루어지고 있다. 본 논문에서는 UAV와 그에 장착된 일반 카메라를 사용하여 무인 자동화 시스템을 설계하고자 한다. 설계된 시스템은 머신 러닝을 사용하여 목표 객체를 탐지한다. 또한 목표 객체와 UAV 사이의 거리를 측정하며, 측정된 정보를 사용하여 UAV의 속도와 방향을 제어하여, UAV가 목표 객체를 충돌 없이 추적하도록 한다.

Ⅰ. 서 론

최근 UAV(Unmanned Aerial Vehicle)는 다양한 분야에서 사용되고 있다. 특히 기존에 사람이 직접 임무를 수행하기 힘든 공중에서의 영상 수집에 적극적으로 사용되는 추세이다. 이러한 임무를 무인 자동화 시스템으로 수행하기 위해서는 UAV의 자율주행이 필수적이며, 자율주행을 위하여 여러가지 센서가 장착된다. 여러가지 센서가 장착된다면 UAV의 거리측정 및 주변 환경의 정보 습득 등의 여러가지 정보 수집에 유리해지지만, UAV의 무게가 증가하고, 단가 상승의 요인이 된다.

본 시스템에서는 그러한 단가 상승의 문제를 완화하기 위한 방식으로 카메라를 이용하여 센서의 기능을 대체함으로써 주변 시야의 정보를 수집하고 목표 탐지 및 거리 측정하는 시스템을 설계한다.

Ⅱ. 본론

본 시스템은 UAV 시야에 보이는 목표 객체를 일정 거리를 유지하며 추적하는 것을 목표로 하며, 객체를 탐지하는 실시간 객체 탐지 시스템과 실시간으로 목표 객체까지의 거리를 측정하는 거리 측정 시스템으로 나누어진다. 실시간 객체 탐지 시스템의 경우 Yolo-v3-tiny를 사용하였으며, 거리측정은 Yolo-v3-tiny에서 탐지된 객체를 대상으로 실시된다. 또한 얻은 측정 정보를 사용하여 ROS를 사용하여 UAV가 제어된다.

A. 객체 탐지 시스템

객체 탐지 시스템은 Yolo-v3-tiny를 사용하여 진행되었으며, 본 시스템에서는 2가지의 객체를 사용하여 실험이 진행되었다. 객체 탐지를 위하여 약3000개의 데이터세트가 사용되었으며, 적색 화살표와 청색 화살표로 목표 객체임을 표기하였다. 본 실험에서 Yolo-v3-tiny를 사용하기 위하여 GTX 1060 그래픽카드가 사용되었다. 또한 ROS환경에서 실행하기 위하여 darknet_ros가 사용되었다.[1] 그림3은 Yolo-v3-tiny를 사용하여 주어진 데이터세트로 학습시킨 결과이며, 2064개의 train 데이터세트와 1031개의 validation 데이터세트를 사용하였다.

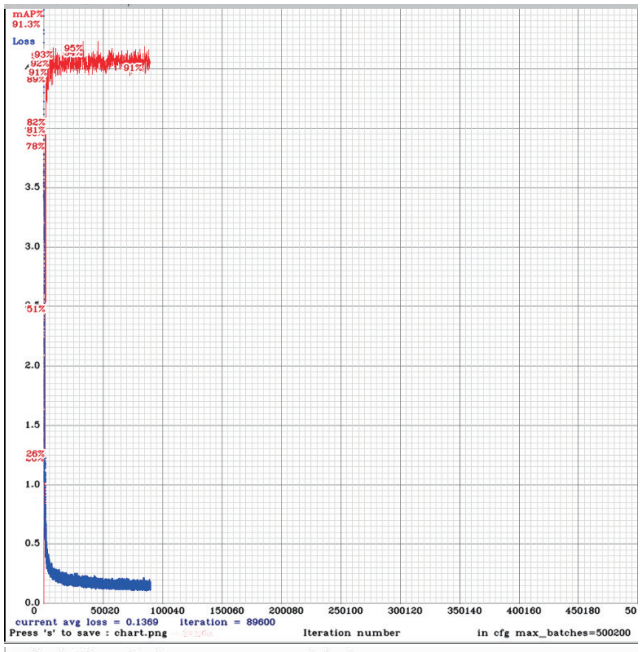
B. 거리 측정 시스템

거리 측정은 목표 객체를 대상으로 진행되었다. 거리 측정은 수식(1)과 같은 방법으로 측정되었으며 W는 탐지된 객체의 바운딩 박스의 가로 길이이며, X는 거리 측정을 위해 얻은 상수 값 이다. D는 객체와 UAV사이의 거리를 나타내며, D의 크기에 따라서 UAV의 속도를 조절한다.

$$\frac{150(mm)}{W(mm)} \times X = D \quad (1)$$

C. UAV 제어 시스템

UAV는 Parrot Bebop 2를 사용하였으며 ROS Kinetic 환경에서 진행되었다. 또한 bebop_driver에서 기본적으로 제공되는 기능을 사용하였다.[2] 그림 2 는 제안된 UAV방향 제어 및 속도 제어 알고리즘이다. 알고리즘은 중 방향 제어의 경우 그림 3 과 같이 UAV가 수집하는 전체 영상을 9등분하여 UAV의 높이와 좌, 우 방향을 조정하게 된다. 1번 영역에 목표 객체가 있다면, UAV가 좌상단으로 이동하고 8번 영역에 목표 객체가 존재한다면 UAV는 하단으로 이동하게 된다. 또한 5번 영역에 목표 객체가 존재한다면 UAV는 방향을 유지한다. UAV의 제어는 bebop_autolanding에서 제공하는 코드를 참고하였다.[3]




```
 calculation mAP (mean average precision)...
1032
 detections_count = 2336, unique_truth_count = 1822
class_id = 0, name = BLUE, ap = 90.88%          (TP = 719, FP = 138)
class_id = 1, name = RED, ap = 98.07%           (TP = 996, FP = 113)

 for thresh = 0.25, precision = 0.87, recall = 0.94, F1-score = 0.91
 for thresh = 0.25, TP = 1715, FP = 251, FN = 107, average IoU = 72.37 %

 IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
 mean average precision (mAP@0.50) = 0.944763, or 94.48 %
Total Detection Time: 13.000000 Seconds
```

그림 1. Yolo-v3-tiny train 결과

Ⅲ. 결론

본 논문에서는 장착된 카메라를 이용하여 센서의 기능을 대체하고 UAV의 움직임을 제어하는 알고리즘을 구현하였다. 카메라 하나로 센서의 기능까지 이용할 수 있기 때문에 추가되는 센서의 비용을 절감하는 효과를 가져올 수 있다. 또한, UAV에 탑재되는 센서의 수를 줄임으로써 로드의 감소로 인한 비행시간 증대를 이룰 수 있다.

하지만, 이 시스템의 경우 ROS로 제어하는 컴퓨터가 주변에 필요하기 때문에 지금의 방법으로는 활동 반경에 제약이 존재한다. 또한 하나의 카메라로 영상을 수집하는 만큼 후방이나 하단 시야에 제한이 있는 것은 분명한 사실이다. 이것을 해결하기 위하여 이후에 360° 카메라를 사용한 연구를 진행할 예정이다.

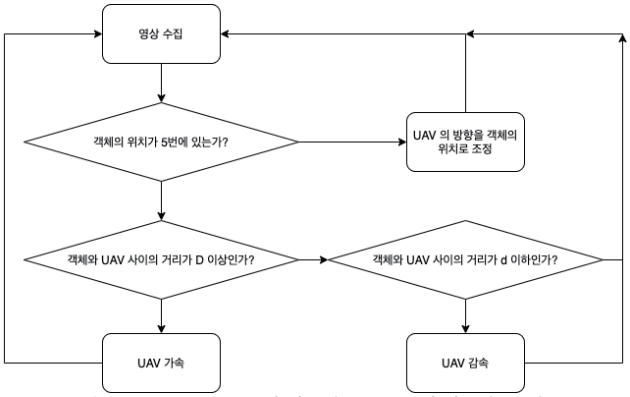


그림 2. UAV 방향 제어 및 속도 제어 알고리즘

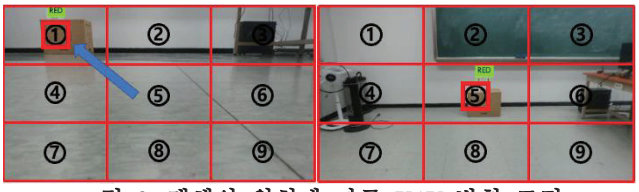


그림 3. 객체의 위치에 따른 UAV 방향 조정

ACKNOWLEDGMENT

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ICAN(ICT Challenge and Advanced Network of HRD) program(IITP-2025-RS-2022-00156394) supervised by the IITP(Institute of Information & Communications Technology Planning & Evaluation, 50%) This research was supported by Basic Science Resarch Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(2018R1A6A1A03024003, 550%)

참 고 문 헌

[1] M. Bjelonic "YOLO ROS: Real-Time Object Detection for ROS",URL: https://github.com/leggedrobotics/darknet_ros, 2018.

[2] thomas-bamford "bebop_autonomy - ROS Driver for Parrot Bebop Drone (quadrocopter) 1.0 & 2.0", URL: https://github.com/AutonomyLab/bebop_autonomy/blob/indigo-devel/docs/index.rst, 2017.

[3] hoa0202 "bebop_autolanding", URL: https://github.com/hoa0202/bebop_autolanding.git, 2020.