

드론 네트워크에서 엣지컴퓨팅 기반 객체 검출 성능 비교에 관한 연구

박재필[§], 박상기[§], 신지혜*, 방인규^{■†}, 김태훈^{■§}

[§]한밭대학교 컴퓨터공학과, *한밭대학교 정보통신공학과, [†]한밭대학교 지능미디어공학과

20181617@edu.hanbat.ac.kr, 20181616@edu.hanbat.ac.kr, 20201847@edu.hanbat.ac.kr,
ikbang@hanbat.ac.kr, thkim@hanbat.ac.kr

Performance Comparison of Object Detection based on Edge Computing in Drone Networks

JaePil Park[§], SangGi Park[§], JiHye Shin*, Inkyu Bang^{■†}, Taehoon Kim^{■§}

[§]Department of Computer Engineering, Hanbat National University

*Department of Information and Communication Engineering, Hanbat National University

[†]Department of Intelligence Media Engineering, Hanbat National University

요약

최근 드론(Drone)은 운송, 농업, 군사 등 여러 산업 분야에 활발히 활용되고 있으며, 인공지능, 자율주행 등과 융합하며 새로운 응용 애플리케이션이 개발되고 있다. 본 연구에서는 자원 제약적(resource limited) 드론의 원활한 임무(mission) 수행을 위해 엣지 컴퓨팅 환경을 고려하고자 한다. 보다 구체적으로, 엣지 컴퓨팅을 위한 서버에서 사용하는 모델별 임무 완수 시간(mission completion time)을 측정하고, 임무 완수 시간과 mAP(mean average precision) 간의 상충관계(trade-off relationship)를 검증한다.

I. 서론

드론은 무인 항공기 중 하나로써 조종사가 탑승하지 않고, 자동/반자동으로 조종이 가능한 항공기로 정의된다. 드론은 운송, 농업, 군사 등 다양한 산업에서 활용되고 있으며, 최근 4차 산업혁명이라는 새로운 물결로 인하여 새로운 응용 애플리케이션이 지속적으로 등장하고 있다. 구글, 아마존과 같은 세계적인 기업들도 드론을 활용하여 배달하는 서비스를 개발 중이며, 산물 진화용 드론 등 다양한 분야에서 활용되고 있다.

드론과 관련된 연구는 역사가 오래되었으며 최근에는 인공지능, 자율주행 등의 기술들의 발전으로 인해 융합 연구들이 진행되고 있다. 대표적으로 드론에서 객체 탐지(object detection)를 수행하는 과정에서 성능 개선을 위한 연구들이 많이 보고되고 있으며 [1], 물체를 추적(tracking)하는 과정에서 성능 개선을 위한 연구들이 보고되고 있다 [2, 3]. 또한, 드론의 성능이 제약적인 상황이 빈번히 발생할 수 있기 때문에, 드론의 연산 능력이 부족한 상황에서 클라우드로 임무를 위탁(offloading)하여 임무 완수 시간을 단축하고자 하는 연구도 보고된 바 있다 [4].

본 연구에서는 자원 제약적 드론을 활용하며, 연산 능력의 부족함을 보완하기 위해 드론 인근 주변의 서버에 연산을 위탁(offloading)하는 엣지 컴퓨팅 환경을 고려하고자 한다. 보다 구체적으로, 드론을 통해 객체를 검출하는 임무를 수행하기 위해 주변 서버(server)의 도움을 받아 임무를 완수하는데까지 소요되는 시간을 측정하고, 서버에서 활용하는 모델에 따른 임무 완수 시간과 mAP 간의 상충관계를 검증한다.

II. 시스템 모델

본 연구에서는 드론이 주변 인근을 정찰하며 객체를 검출하고 분류하

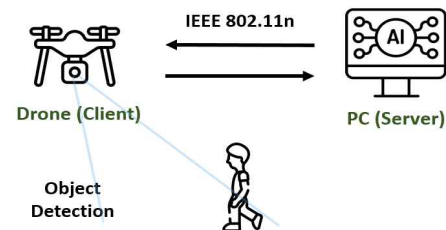


그림 1. 시스템 모델

Fig. 1. System Model

는 시나리오를 고려하고자 한다. 그림 1은 본 연구에서 고려하고 있는 시스템 구성을 나타내고 있으며, 1대의 드론과 엣지컴퓨팅을 수행할 수 있는 1대의 지상 통제 장비(GCS: ground control station)으로 구성되어 있다. 드론의 연산 능력이 충분할 경우 드론에서 로컬(local) 컴퓨팅을 수행하면 되지만, 드론의 연산 능력이 충분하지 못하는 경우 임무를 완수하는데까지 소요되는 시간이 증가하기 때문에, 본 연구에서는 GCS의 연산 능력을 활용하고자 한다. 즉, 본 연구에서는 자원 제약적인 드론을 고려하며 임무를 수행하는 과정에서 엣지 컴퓨팅을 활용한다.

드론은 객체를 검출하고 분류하는 임무를 수행하기 위해 카메라로 촬영한 이미지를 GCS로 전송하며, GCS는 수신한 이미지를 이용해 목표물인 특정 객체를 검출하고 분류하는 작업을 수행하며 목표물의 유무를 판단한다. 객체 검출을 위해 YOLO(You Only Look Once) [5]를 활용하며, 드론과 GCS 사이에는 IEEE 802.11n 기반의 WiFi-Direct를 이용해 무선통신을 수행한다.

III. 성능 평가

본 논문에서는 임무 완수 시간(mission completion time)을 성능 지표로 활용하며, 임무 완수 시간은 드론이 임무를 완수하는데까지 소요되는 총 시간으로 요청 및 응답 시간(request and response time)과 검출 시간(detection time)의 합으로 정의한다. 보다 구체적으로, 요청 및 응답 시간은 GCS에서 드론에게 이미지 촬영을 요청하고 그에 대한 응답을 수신하는 시간이며, 검출 시간은 GCS에서 YOLO(You Only Look Once) 모델을 활용해 객체를 검출하는데까지 소요되는 시간을 의미한다.

성능 평가를 위해 필드 실험을 진행하였다. 드론은 DJI사의 Tello Drone을 활용하였고, 엣지 컴퓨팅 성능 비교를 위해 i5-6200(8GB)급의 노트북과 i5-11500(32GB)급의 노트북을 GCS로 활용하였다. GCS가 드론에 이미지 전송 요청 메시지를 송신하게 되면 드론은 해당 메시지를 성공적으로 수신했을 경우 탑재된 5MP(2592×1936)급의 카메라를 통해 촬영된 이미지를 GCS로 전송하게 된다. GCS는 수신한 이미지를 후처리(post-processing) 한 후 YOLO 모델을 활용해 객체 검출 및 분류를 시도한다. 본 연구에서는 객체 검출에 소요되는 시간 및 mAP 간의 관계를 확인하기 위해 yolov2, yolov3, yolov4, yolov2-tiny, yolov3-tiny, yolov4-tiny 모델을 활용한다.

표 1은 i5-11500(32GB)급 노트북을 활용하여 엣지 컴퓨팅을 수행하는 시나리오에서 모델별 요청 및 응답 시간, 검출 시간, 임무 완수 시간을 보여주고 있다. 요청 및 응답 시간은 통신 프로토콜, 드론의 성능에 결정되는 부분이기 때문에 어떤 모델을 사용하든 유사한 시간이 측정된 것을 확인할 수 있었고, 측정된 시간이 상당히 작기 때문에 엣지 컴퓨팅을 활용하는데도 큰 무리가 없는 것을 확인할 수 있었다. 또한, YOLO의 버전(version)이 개선됨에 따라 객체를 검출하는데 소요되는 시간이 증가하는 것을 확인할 수 있었고, 경량화된 모델을 사용할 경우 검출 시간이 상당히 줄어드는 것을 확인할 수 있었다.

그림 2는 객체 검출 모델별 임무 완수 시간과 mAP 사이의 상충관계(trade-off relationship)를 보여주고 있다. 보다 우수한 mAP를 보장하는 모델을 사용하는 경우에는 임무를 완료하는데까지 소요되는 시간이 증가하는 경향성을 보이는 것을 확인할 수 있었고, 경량화된 모델을 이용할 경우 임무 완료 시간을 상당히 줄일 수 있었지만 mAP 성능 또한 열화되는 것을 확인할 수 있었다. 실험에 활용한 노트북의 성능에 따라 임무 완료까지 소요되는 시간이 달라질 수 있음을 확인하였으며, 이는 엣지 컴퓨팅을 활용하는 상황에서 목표로 하는 성능을 만족시키기 위해 서버로 활용되는 기기가 충분한 연산 능력을 갖추고 있는지 면밀히 검토할 필요가 있을 것으로 보인다.

표 1 i5-11500 (32GB)급 엣지 컴퓨팅 서버 활용 시나리오에서 모델별 임무 완수 소요 시간 비교

Table 1. Comparison of mission completion time for varying detection models in scenario with edge computing server of i5-11500 (32GB)

	yolo2	yolo3	yolo4	yolo2-tiny	yolo3-tiny	yolo4-tiny
Request and Response Time	4.7ms	4.6ms	4.5ms	4.5ms	4.6ms	4.6ms
Detection Time	167.8 ms	423.5 ms	469.5 ms	36.3 ms	41.5 ms	46.6 ms
Mission Completion Time	172.5 ms	428.1 ms	474.1 ms	40.8 ms	46.1 ms	51.2 ms

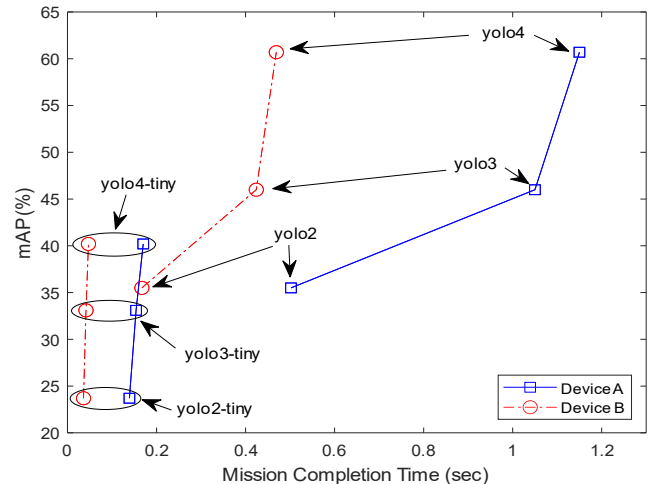


그림 3 객체 검출 모델별 임무 완수 시간과 mAP 사이의 상충 관계

Fig. 2. Trade-off relationship between mission completion time and mAP according to object detection models

III. 결론

본 논문에서는 자원 제약적 드론의 원활한 임무 수행을 위해 엣지 컴퓨팅을 고려한다. 보다 구체적으로, 상용 드론과 노트북을 활용하여 필드 실험을 구성하고, 사용하는 검출 모델에 따라 임무 완료 시간과 mAP 사이에 존재하는 상충관계를 면밀히 관찰했다. 실험을 설계하고 수행하는 과정에서 상용 드론을 활용하였는데, 제공되는 API를 이용해 드론을 제어하는 것은 가능할지라도 그 외의 기능을 직접 프로그래밍하는 것은 불가능했다. 추후 프로그래밍 가능한 드론을 활용해 로컬에서 컴퓨팅하는 경우에 달성할 수 있는 성능들과 비교 분석을 진행한다면 보다 의미있는 결과를 확인하고 도출할 수 있을 것으로 예상된다.

ACKNOWLEDGMENT

본 논문은 한밭대학교 공학교육혁신센터 “창의 융합형 공학 인재 양성 지원 사업” 과 2022년도 해양수산부 재원으로 해양수산과학기술진흥원의 지원을 받아 수행된 연구임 (20210671,수산식품 스마트 가공 기술개발)

참 고 문 헌

- [1] W. Budiharto, A. A. S. Gunawan, J. S. Suroso, A. Chowanda, A. Patrik and G. Utama, "Fast Object Detection for Quadcopter Drone Using Deep Learning," 2018 3rd International Conference on Computer and Communication Systems (ICCCS), Nagoya, Japan, 2018, pp. 192-195, doi: 10.1109/CCOMS.2018.8463284.
- [2] C.-T. Dang, H.-T. Pham, T.-B. Pham and N.-V. Truong, "Vision based ground object tracking using AR.Drone quadrotor," 2013 International Conference on Control, Automation and Information Sciences (ICCAIS), Nha Trang, Vietnam, 2013, pp. 146-151, doi: 10.1109/ICCAIS.2013.6720545.
- [3] 임창훈, 권대현, 신수용, "딥러닝 기반 특정 객체 추적 드론 시스템," 2021년도 한국통신학회 하계종합학술발표회 논문집, pp. 1436-1437, 2021.
- [4] J. Lee, J. Wang, D. Crandall, S. Šabanović and G. Fox, "Real-Time, Cloud-Based Object Detection for Unmanned Aerial Vehicles," 2017 First IEEE International Conference on Robotic Computing (IRC), Taichung, Taiwan, 2017, pp. 36-43, doi: 10.1109/IRC.2017.77.
- [5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2015. [Online]. Available: <https://arxiv.org/abs/1506.02640>