J. Korean Soc. Aeronaut. Space Sci. 49(10), 883-892(2021)
DOI:https://doi.org/10.5139/JKSAS.2021.49.10.883
ISSN 1225-1348(print), 2287-6871(online)

온-보드에서의 딥러닝을 활용한 드론의 실시간 객체 인식 연구

이장우[1], 김주영[2], 김재경[3], 권철희[4]

A Study on Realtime Drone Object Detection Using On-board Deep Learning

Jang-Woo Lee[1], Joo-Young Kim[2], Jae-Kyung Kim[3] and Cheol-Hee Kwon[4]
LIG Nex1 Co.

ABSTRACT

This paper provides a process for developing deep learning-based aerial object detection models that can run in realtime on onboard. To improve object detection performance, we pre-process and augment the training data in the training stage. In addition, we perform transfer learning and apply a weighted cross-entropy method to reduce the variations of detection performance for each class. To improve the inference speed, we have generated inference acceleration engines with quantization. Then, we analyze the real-time performance and detection performance on custom aerial image dataset to verify generalization.

초 록

본 논문에서는 드론을 활용한 감시정찰 임무의 효율성을 향상하기 위해 드론 탑재장비에서 실시간으로 구동 가능한 딥러닝 기반의 객체 인식 모델을 개발하는 연구를 수행하였다. 드론 영상 내 객체 인식 성능을 높이는 목적으로 학습 단계에서 학습 데이터 전처리 및 증강, 전이 학습을 수행하였고 각 클래스 별 성능 편차를 줄이기 위해 가중 크로스 엔트로피 방법을 적용하였다. 추론 속도를 개선하기 위해 양자화 기법이 적용된 추론 가속화 엔진을 생성하여 실시간성을 높였다. 마지막으로 모델의 성능을 확인하기 위해 학습에 참여하지 않은 드론 영상 데이터에서 인식 성능 및 실시간성을 분석하였다.

Key Words : Deep Learning(딥러닝), Object Detection(객체 인식), Data Augmentation(데이터 증강), Transfer Learning(전이 학습), Class Imbalance(클래스 불균형), Inference Acceleration (추론 가속화)

Ⅰ. 서 론

컴퓨터 비전(Computer Vision) 기술은 컴퓨터가 디지털 영상 또는 비디오에서 높은 수준의 시각 인지를 달성하여 영상을 이해하고 자동화하는 것을 목표로 한다. 이러한 목표를 달성하기 위해 단일 영상 또는 일련의 영상에서 유용한 정보를 획득하고 분석하기 위한 이론 및 알고리즘 기술이 필요하다. 최근 기계학습과 딥러닝 기술이 다양한 컴퓨터 비전 분야에 적용되어 객체 인식, 추적 및 행동 인식 등의 여러 벤치마크 데이터에서 높은 성능을 달성하였다[1]. 컴퓨터 비전 기술이 발전하면서 자율주행, 로보틱스 및 머신 비전 등의 응용 분야에 적용되고 있다. 특히 높은 기동성을 가진 드론은 농업, 수색, 구조 및 군사 등 다양한 영역에서 활용될 수 있으며, 드론의 고도화된 자율비행을 위해 동적인 환경에서 주변 정보

† Received : May 24, 2021 Revised : September 14, 2021 Accepted : September 27, 2021
[1] Research Engineer, [2,3] Chief Research Engineer, [4] Research Fellow
[1] Corresponding author, E-mail : jangwoo.lee2@lignex1.com, ORCID 0000-0002-4442-340X

를 인지하고 의사결정을 가능하게 하는 지능적인 요소에 대한 연구가 필요하다. 이에 기반이 되는 핵심 기술인 객체 인식(Object Detection)은 영상 내의 관심 객체 위치를 정밀하게 찾아내고 객체의 범주를 분류할 수 있는 기술로 상황 정보 및 의사결정에 필요한 정보를 제공할 수 있다.

특히 감시정찰 분야는 실시간성과 높은 정확도가 보장된 객체 인식 기술에 대한 중요도가 높다. 감시정찰 임무를 수행하기 위해 드론에서 촬영된 영상을 수신하고 드론을 통제하기 위해 지상통제시스템(Ground Control System)을 활용하며, 지상운용자가 수신된 영상을 통해 객체 인식을 원활하게 할 수 있도록 다양한 영상처리 기능을 제공한다. 이와 같은 방법은 지상운용자의 숙련도 및 판단에 의존적이며 장시간 임무로 발생하는 운용자의 피로도 누적으로 인해 임무 수행 효율성이 감소한다. 또한 드론에서 촬영된 영상을 수신 받고 처리하는 과정에서 상황에 대한 인식 및 대응 시간에 지연이 발생하고 이로 인해 임무 달성이 저하된다.

이러한 이유로 촬영된 영상에 대해 드론 탑재 장비에서 직접 객체 인식을 수행할 필요가 있다. 드론 플랫폼에서의 객체 인식은 영상 내 노이즈, 블러 및 소형 객체 등과 같은 다양한 항공 영상 특성을 고려해야 하므로 일반적인 객체 인식에 비해 매우 도전적이고 복잡하다. 또한 드론의 비행성능(비행시간, 배터리 소모 등)에 미치는 영향을 최소화해야 하므로 리소스가 제한된 환경에서 임무 수행의 실시간성을 보장하면서 객체인식 성능을 최대화할 수 있도록 객체 인식 모델의 설계 및 최적화가 필요하다.

온-보드에서 딥러닝을 활용한 드론의 실시간 객체 인식 분야는 이러한 문제점들을 해결하기 위해 다양한 접근 방법으로 연구되고 있다. Vaddi 등은 드론 탑재 플랫폼의 제한된 리소스를 극복하고자 객체 인식 과정을 분리하여 높은 연산량이 요구되는 부분은 클라우드에서 처리하고 나머지 분류 단계와 실시간 탐지는 온-보드에서 수행하는 하이브리드 방식을 제안하였다[2]. Lee 등은 실시간 객체인식을 온-보드에서 구동하기 위해 경량화된 네트워크 아키텍처를 제안하여 Jetson TX2 보드 기준으로 초당 15프레임을 처리할 수 있음을 보였다[3].

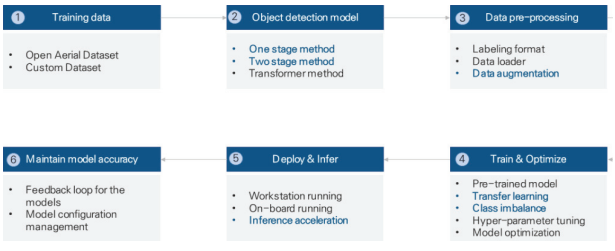

Fig. 1. Deep learning development process for onboard object detection

본 논문에서는 앞서 제기한 다양한 영상 특성, 제한된 리소스, 실시간성 등을 종합적으로 고려하여 Fig. 1과 같이 온-보드에서 구동 가능한 딥러닝 기반 객체 인식 모델의 개발 방법을 제안한다. 본 논문의 구성은 다음과 같다. 2장 1절에서는 딥러닝 기반의 객체 인식 모델을 분석하고 드론 영상에서 인식 성능과 실시간성을 높이기 위해 학습 전략, 클래스 불균형 및 추론 가속화 방법을 제안한다. 2장 2절에서는 제안한 연구 방법이 적용된 각 모델에 대한 성능 평가 도출 및 분석 내용을 설명한다. 3장에서는 본 연구의 결론을 맺는다.

II. 본 론

2.1 딥러닝 기반의 객체 인식 개발 방법

드론에서 촬영된 영상에 대해 온-보드 기반 객체 인식 모델을 개발하기 위해서는 Fig. 1의 프로세스를 따르며 특히, 파란색으로 표기한 사항을 고려해야 한다. 객체 인식 모델 선정 단계에서 모델의 실시간 온-보드 구동을 고려하여 인식 성능과 실시간성을 충족하는 One-stage 방법의 YOLOv3, Two-stage 방법의 Faster R-CNN 구조를 활용한다. Deploy & Infer 단계는 온-보드에서 구동되며 나머지 과정은 워크스테이션에서 개발 및 성능 평가가 이루어진다.

드론의 빠른 기동으로 인해 카메라 이동 및 회전이 발생하고 이는 관심 객체에 모션 블러(Motion Blur)를 동반한 외형(Appearance) 정보의 변형을 야기할 수 있다[4]. 데이터 전처리 단계에서 공개된 드론 영상 학습 데이터인 DOTA[5], VisDrone[6]을 기반으로 부족한 학습 데이터를 보강하고 인식 성능을 높이기 위해 영상 손실 압축 및 모션 블러 현상 등을 모사하는 데이터 증강(Data Augmentation)을 수행한다. 학습 및 최적화 단계에서는 학습 모델을 기반으로 유사 데이터에 대한 전이학습을 수행하여 학습에 참여하지 않은 영상에 대한 모델의 인식 성능을 향상시킬 수 있다.

일반적으로 활용 가능한 PASCAL VOC, COCO 등의 벤치마크 데이터와 달리 높은 고도에서 운용되는 드론 영상의 특징으로 관심 객체의 크기가 매우 소형이며 특정 상황에서 영상 내의 많은 개체 수가 존재한다. 이러한 영상 특성이 반영된 DOTA, VisDrone 등의 벤치마크 데이터가 존재하지만, 클래스 별 학습 샘플 수의 편차가 크고 이로 인해 특정 클래스의 학습 비중이 낮아지고 성능이 저하되는 클래스 불균형 문제가 발생한다[7]. 학습 및 최적화 단계에서 학습 데이터에 존재하는 클래스 불균형 문제를 개선하는 방안으로 학습 단계에서 각 클래스 별 학습 샘플을 기반으로 가중치를 설정하고 크로스 엔트로피 오차 함수에 반영한다.

온-보드에서 고성능 딥러닝 모델을 구동하는 경우 제한된 리소스로 인해 추론 시간이 증가하며 실시간성이 저하된다. 상대적으로 파라미터 수가 적은 딥러닝 모델을 활용하면 구동 리소스를 줄일 수 있지만 고성능 모델 대비 인식 성능이 낮아질 수 있다. Deploy & Infer 단계에서 생성된 모델의 실시간성을 높이는 방안으로 추론 가속화를 수행한다. 양자화, 캘리브레이션 및 그래프 최적화 기능을 제공하는 TensorRT를 활용하여 기존 모델의 성능을 유지하며 추론 속도를 높일 수 있다.

2.1.1 객체 인식 모델 선정

2.1.1.1 객체 인식 모델 아키텍처

객체 인식을 위한 딥러닝 모델은 입력 영상 내 존재하는 객체의 위치를 추출하고 해당 객체가 속하는 카테고리를 예측하는 분류 작업을 수행한다. 성능을 높이는 방안으로 다양한 모델 아키텍처가 연구되고 있으며 이러한 객체 인식을 위한 딥러닝 모델은 공통적으로 backbone, Neck, Head라는 3가지 네트워크 요소로 Fig. 2와 같이 구성된다.

Backbone은 입력 영상에 대하여 특징맵(Feature Map)을 추출하는 기능을 수행한다. 입력 영상 해상도를 상대적으로 낮은 해상도로 down sampling하여 핵심적인 특징을 내포한 특징맵으로 변환한다. 대표적인 backbone 구조로는 VggNet, ResNet50/100, MobileNet

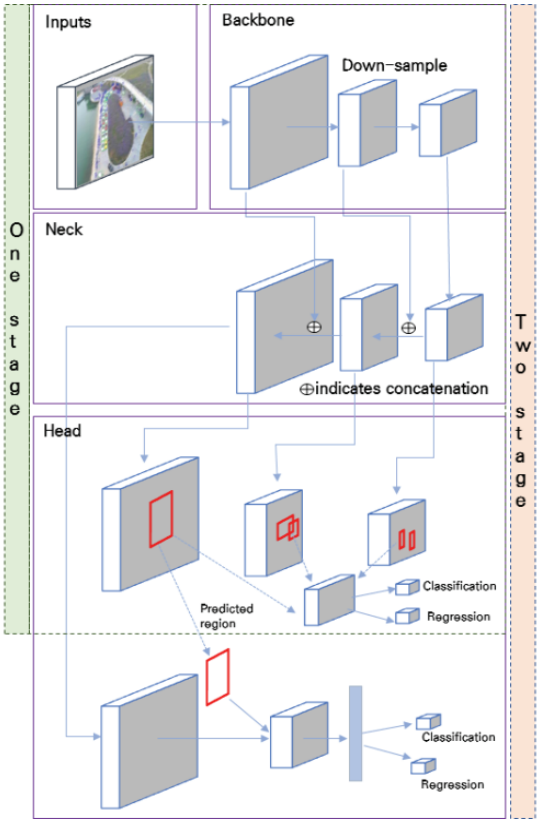

Fig. 2. Object detection model Architecture

등의 네트워크 아키텍처가 존재한다. Neck은 backbone과 head를 연결하는 네트워크 구조로 backbone에 의해 생성된 특징맵을 개선하고 재구성하는 기능을 수행한다. FPN(Feature Pyramid Network) 구조를 통해 backbone을 통과하며 중간 단계에 생성되는 특징맵을 융합하여 다양한 스케일에서도 강인한 특징을 내포한 특징맵으로 재구성된다. Head는 최종적으로 추출된 특징맵을 기반으로 붉은색 상자와 같이 객체가 존재하는 영역을 예측하여 추론 결과에 대한 클래스 분류와 경계 상자 회귀의 기능을 수행한다.

객체 인식 모델은 Fig. 2와 같이 크게 2가지 구조로 구분할 수 있다. YOLO, SSD와 같은 1단계 인식 모델(One Stage Detector)과 2단계 인식 모델(Two Stage Detector)로 대표적인 Faster R-CNN 구조가 있다. 2단계 인식 모델은 1단계 인식 모델과 비교하여 객체 분류 성능과 위치 예측에 대한 정확도가 높다는 장점이 있지만 다소 낮은 추론 속도를 보인다. 2단계 인식 모델은 첫 번째 단계에서 RPN(Region Proposal Network) 구조를 통해 후보가 될 수 있는 객체에 대한 영역인 경계 상자를 후보로 선정한다. 두 번째 단계에서 후보로 선정된 경계 상자를 기준으로 ROI Pooling 과정을 거쳐 클래스 분류와 경계 상자 회귀 과정을 수행하게 된다. 이와 달리 1단계 인식 모델은 RPN과 같은 영역 제안 단계 없이 추출된 특징맵에서 직접 클래스 분류와 경계 상자 회귀 과정을 동시에 수행하기 때문에 높은 추론 속도를 보인다[1]. 이러한 분석을 통해 실시간성이 높은 One-stage 방법의 YOLOv3와 높은 인식 성능을 보이는 Two-stage 방법의 Faster R-CNN 모델을 활용하여 드론 영상을 기준으로 성능을 도출하고 두 모델을 비교한다.

2.1.1.2 YOLOv3

YOLO는 1단계 인식 모델로 영역 제안 단계 없이 경계 상자와 클래스 분류 예측을 동시에 수행하여 추론 속도를 높인 구조이다[8]. YOLOv3는 backbone으로 3x3, 1x1 합성곱 레이어와 일부 커넥션을 결합하여 53개 레이어를 가진 Darknet53을 적용하였다. Fig. 3과 같이 각기 다른 스케일의 3가지 특징맵을 추출하고 합성곱 레이어와 배치 정규화를 거쳐 최종 인식 결과 출력 크기는 13x13xN, 26x26xN, 52x52xN의 텐서로 표현되며 N은 객체가 존재하는 확률, 경계 상자 개수 및 각 클래스의 분류 확률을 표현할 수 있는 텐서 크기로 계산된다. 다중 스케일 특징맵 기반의 예측을 수행하므로 YOLO, YOLOv2와 비교하여 다양한 크기를 가진 객체에 대한 인식 성능이 향상되었다[8].

2.1.1.3 Faster-RCNN

Faster R-CNN은 기존 R-CNN 계열에서 활용되던 RPN을 단일 네트워크로 병합하고 영역 제안 연산으로 인한 병목 현상을 제거하여 2단계 인식 모델의

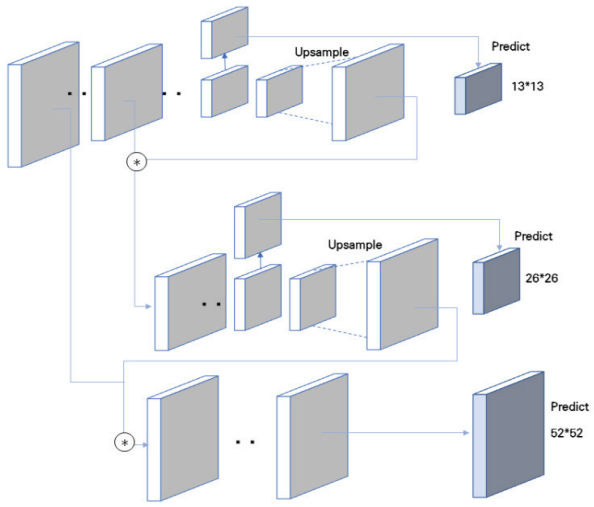

Fig. 3. YOLOv3 Head Architecture

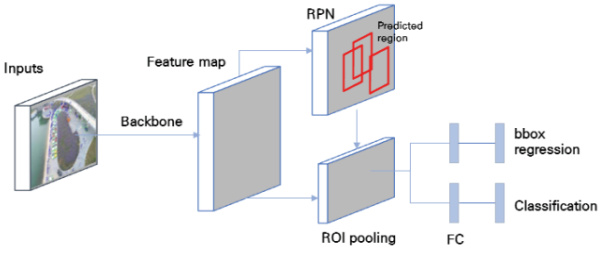

Fig. 4. Faster RCNN Head Architecture

느린 추론 속도를 개선하였다[9]. Fig. 4와 같이 입력 영상으로부터 객체 특징을 추출하기 위해 backbone 구조를 선정해야 한다. 대표적으로 활용되는 구조로 ResNet50과 ResNet100이 있고 뒤에 숫자로 표기된 레이어의 수가 많아질수록 인식 성능은 높아지고 연산량도 증가한다. Backbone을 거쳐 추출된 특징맵에서 RPN을 통해 후보 객체 위치를 제안하고 이를 완전 연결 계층의 고정된 입력으로 변환하기 위해 ROI pooling 연산을 수행한다. Neck의 구성 요소로 FPN을 활용하여 다양한 스케일로 특징맵을 재구성할 수 있다. 고해상도 특징맵은 저해상도 특징맵과 비교하여 의미 정보는 상대적으로 낮지만 객체 위치에 대한 특징 정보의 표현력이 더 높기 때문에 다중 스케일의 특징맵을 융합하여 소형 객체에 대한 인식 성능을 향상시킬 수 있다. 이러한 파이프라인을 가진 2단계 인식 모델은 1단계 인식 모델에 비해 상대적으로 정확한 객체 위치를 예측할 수 있지만 추론 속도가 느린 단점이 있다. Faster RCNN 모델의 전반적인 성능과 추론 속도를 고려하여 ResNet50을 backbone으로 사용한다. 본 연구에서는 YOLOv3 모델과 Faster RCNN 모델을 활용하여 그 성능을 비교한다.

2.1.2 학습 데이터 전처리 및 증강

학습 데이터는 영상 및 각 영상 내 존재하는 객체의 레이블로 구성되어 있다. 학습 단계에서 영상 및

Fig. 5. Data augmentation result in Aerial Dataset

레이블 데이터를 GPU에 로드 가능한 형태로 변환하는 데이터 전처리 과정이 필요하다. 데이터 전처리는 영상과 레이블을 메모리로 로드하고 영상 데이터 사이즈를 인식 모델의 입력 크기에 맞게 변환하고 정규화한다. 학습 데이터와 평가 데이터 간 모델의 성능 차이가 존재하고 학습 데이터가 부족한 경우 과적합이 발생할 수 있다. 데이터 증강의 목적은 영상 인식에서 존재하는 조도 변화, 폐색(occlusion) 및 시점 변화 등의 상황에서도 모델의 인식 성능을 유지할 수 있도록 학습 단계에서 이러한 특성이 반영된 데이터를 생성한다. 데이터 증강을 수행하여 가용 가능한 데이터를 확장하고 모델의 성능을 개선할 수 있다[10]. 증강된 데이터로 충분히 학습된 모델은 외란이 존재하는 실제 영상 데이터에서도 특정 객체의 고유한 특징을 추출할 가능성이 높아진다. 본 연구에서는 Fig. 5와 같이 드론 영상에서 발생 가능한 모션 블러, 조도 변화, 스케일 변화 및 화질 저하 등의 증강 방법을 적용한 데이터를 생성하고 이를 학습 단계에 활용한다.

2.1.3 전이학습

전이학습은 특정 문제에서 학습한 기능을 유사한 문제에서 활용하는 것에 기인한다. 딥러닝에 많이 활용되는 전이학습은 가용 가능한 학습 데이터가 부족한 경우 사전 학습된 모델을 활용하여 효과적으로 목표 성능을 달성하는 것이다[11]. 객체 인식 모델을 Backbone, Neck, Head으로 구분하면 Backbone은 입력 영상 내 객체의 특징을 추출하여 특징맵으로 표현하는 기능을 수행한다. 모델을 특정 학습 데이터로 학습한 경우 이와 유사한 데이터에 대해서도 객체의 일반적인 특징 추출이 가능하기 때문에 사전 학습된 모델의 가중치를 활용할 수 있다. 이와 달리 head 부분은 특정 클래스에 고유하게 학습되므로 분류 성능은 기존 학습 데이터에 크게 의존하게 된다[11].

다량의 데이터로 사전 학습된 모델의 일부 레이어의 가중치를 새로운 학습모델로 전이하여 가중치를 초기화하고 나머지 레이어의 가중치는 무작위로 초기화하여 새로운 데이터에 대한 학습을 진행한다. 또한 학습 단계에서 특정 레이어는 동결(Freezing)하여 가중치를 변하지 않게 하고 낮은 학습률(learning rate)로 새로운 데이터를 재학습한다. 기존 학습 데이터와 새로운 데

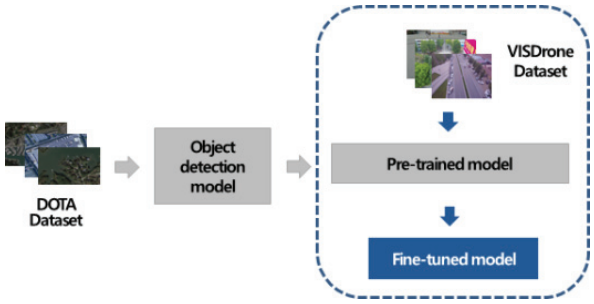

Fig. 6. Transfer learning & Fine tuning for Aerial object detection model

이터의 유사성이 높은 경우 backbone과 neck은 기존 학습 가중치로 초기화하여 동결하고 head만 낮은 학습률로 가중치를 미세 조정(Fine Tuning)한다. 유사성이 상대적으로 낮은 경우 backbone, neck 및 head 모두 동결하지 않고 가중치 미세 조정을 진행한다.

전이학습이 필요한 이유는 드론을 운용하며 확보한 학습 데이터로 모델을 추가 학습하기 위함이다. 모든 학습 데이터에 대해 모델을 전수 재학습하지 않고 추가된 데이터에 대해서만 전체 가중치를 미세조정하거나 head 부분의 가중치만 미세조정 하는 방식을 선택적으로 적용하여 빠른 학습 수렴과 함께 모델의 성능을 높일 수 있다[12].

본 논문에서는 Fig. 6과 같이 DOTA 데이터로 초기 학습을 진행하여 사전 학습모델을 생성한다. 해당 모델을 기반으로 VisDrone 데이터에 대해 전이학습을 진행한다. DOTA와 VisDrone은 영상 해상도와 객체 종류에 차이가 있어 데이터 유사성이 상대적으로 낮으며 학습데이터양이 서로 비슷하다. 이러한 이유로 사전 학습모델의 전체 가중치를 미세 조정하며 낮은 학습률로 전이학습을 진행한다. VisDrone과 유사한 레이블을 가진 자체 데이터를 구축하고 최종적으로 학습된 모델로 그 성능을 평가한다.

2.1.4 클래스 불균형

드론 촬영 영상의 특징으로 운용 고도에 따라 객체의 크기가 변화하며 높은 고도에서는 그 크기가 매우 작고 단일 영상에서 다수의 객체가 존재할 수 있다. 이로 인해 객체 간 출현 빈도에 따라 클래스 불균형이 존재하는 학습 데이터를 획득하게 되고 이러한 데이터로 학습된 객체 인식 모델은 학습 샘플 수가 적은 클래스의 객체에 대하여 낮은 인식 성능을 보일 수 있다[13]. 학습 과정에서 객체 인식 모델의 오차함수는 객체의 존재 유무, 경계 상자의 위치, 클래스 분류 결과 등 여러 오차가 더해진 Joint loss 혹은 Multi-task loss 형태로 이루어져 있다. 특히 클래스 분류에 대한 오차 계산 과정에서 일반적인 크로스 엔트로피 오차함수를 적용하게 되는데 이는 클래스 분포를 고려하지 않기 때문에 샘플 수가 적은 클래스에 대하여 충분한 학습이 이루어지지 않는다. 크로스 엔트로피는 수식 (1)과 같이 표현된다.

$$f(s)_i = \frac{e^{s_i}}{\sum_j^C e^{s_j}}, \quad Loss_{CE} = -t_i \log(f(s)_i) \tag{1}$$

t_i는 Ground truth의 원-핫 인코딩 형태의 벡터 값이다. $f(s)_i$는 소프트맥스 함수로 정규화 되어 확률로 표현되고 주어진 각 클래스에 속하는 i번째 객체의 신뢰도이다. 크로스 엔트로피 오차는 학습 데이터의 각 미니 배치에서 계산된 값을 합산하고 평균을 취한 최종 분류 오차가 역전파를 통해 전달된다. 가중 크로스 엔트로피(Weighted Cross Entropy)는 아래 수식 (2)와 같이 기존 크로스 엔트로피에서 각 클래스에 적합한 가중치를 계산한다.

$$Loss_{WCE} = -w t_i \log(f(s)_i) \tag{2}$$

가중치를 선정하는 방법은 Balanced Cross Entropy, Inverse Class Frequency 등이 있다[7]. Balanced Cross Entropy는 사용자가 임의로 선정하는 방식으로 학습에 중점을 두고 싶은 클래스의 가중치는 $w>1$로 설정하고 나머지 클래스는 $w=1$로 지정한다. Inverse Class Frequency는 각 클래스 샘플 수의 역수를 취하여 가중치를 도출하는 방식이다. 이 방식은 미니 배치에서 출현 빈도가 낮은 클래스의 객체가 존재할 때 오차가 크게 증가하므로 학습이 불안정해질 수 있다. 가중치의 편차를 줄이고 선형적으로 변환하기 위해 매개변수를 곱해주거나 로그함수를 통해 변환된 가중치 값을 사용한다. 수식 (3)에서 q_i는 각 클래스의 샘플 개수이며 k는 임의의 상수이다. k는 실험적으로 정해질 수 있으며 본 논문에서는 k값을 최대 클래스 샘플 개수로 선정하고 q_i는 각 클래스의 샘플 개수와 자연 상수 e를 곱하여 로그함수를 취한 형태로 가중치를 도출한다.

$$Loss_{\mathrm{logarithmic}WCE} = -\log(\frac{k}{q_i}) t_i \log(f(s)_i) \tag{3}$$

2.1.5 추론 가속화

딥러닝 개발 단계는 크게 학습 단계와 추론 단계로 구분할 수 있다. 학습 단계는 다량의 학습 데이터를 기반으로 사용자가 설계한 딥러닝 모델의 가중치를 갱신하여 최적화된 모델을 도출하는 과정이다. 추론 단계는 목적에 맞게 학습된 모델의 추론 기능을 활용하여 응용 프로그램을 설계 및 구현하는 단계이다. 학습 단계는 순전파(Forward Propagation)와 역전파(Back Propagation)에 필요한 중간 변수, 파라미터 및 gradient를 계산하고 저장하는 과정이 필요하고 이는 추론 단계보다 훨씬 큰 메모리와 저장 공간이 요구된다. 텐서플로우, 파이토치 등의 범용적인 딥러닝 프레임워크는 학습 및 추론을 모두 지원하지

만 학습 메커니즘에 필요한 연산 기능에 중점을 두고 있다. 보다 메모리 효율적이며 추론 지연 시간을 단축시킬 수 있는 임베디드 플랫폼과 이에 최적화된 추론용 프레임워크가 필요하다.

현재 자율주행 및 로봇 등의 분야에서 많이 활용되는 엔비디아 GPU 기반의 Jetson 계열 보드와 이를 지원하는 추론 가속 프레임워크인 TensorRT가 있다. TensorRT는 프레임워크의 오버헤드 없이 딥러닝 네트워크의 압축, 최적화가 가능하며 다양한 GPU에서 연산이 가능한 Runtime 엔진을 포함한다[14]. 양자화 및 정밀도 조정으로 딥러닝 프레임워크에서 사용되는 FP32 연산을 FP16 또는 INT8의 형태로 정밀도를 낮추어 추론 속도를 증가시킬 수 있다. 단순히 정밀도를 낮추게 되면 모델 성능이 저하될 수 있기 때문에 양자화된 값과 기존 정밀도 값의 차이가 최소가 되도록 하는 캘리브레이션 방법을 적용하여 양자화에 따른 텐서 정보의 손실을 줄일 수 있다. 동적 텐서 메모리 기능은 각 텐서에 대해 사용 기간 동안에만 메모리를 할당하는 관리 기능으로 메모리 사용량을 줄이고 메모리 재사용량을 개선할 수 있다. 본 연구에서는 TensorRT를 활용하여 드론 영상 데이터로 기 학습된 모델의 추론 가속화를 진행하고 워크스테이션과 Jetson TX2i에서 각각의 성능을 도출한다.

2.2 실험 결과

2.2.1 학습 데이터

본 논문에서는 총 3가지의 데이터를 기반으로 모델을 학습하고 평가 지표를 기반으로 성능을 도출한다. 첫 번째, 항공 위성 영상 기반의 데이터셋인 DOTA(Dataset for Object detection in Aerial images)는 다양한 스케일의 2,806장 영상과 총 188,282개의 주석이 제공되며 객체는 16개의 범주로 구성된다. DOTA를 기반으로 사전 학습 모델을 생성한다. 두 번째는 다양한 장소에서 고도를 변화하여 드론으로 촬영된 총 8,599장의 영상과 전체 540k개 이상의 객체에 대한 경계 상자와 12개의 범주로 구성된 VisDrone-DET2019 챌린지의 데이터셋이다. Fig. 7과 같이 DOTA, VisDrone 데이터셋 내에 각 카테고리에 해당하는 객체의 수를 도시하였다. 마지막으로 자체 데이터셋을 구축하기 위해 640x480 해상도를 가진 스트리밍 영상을 캡처하여 609장의 영상과 각 영상에 대한 주석 데이터를 생성하였다. 해당 주석 데이터는 VisDrone 데이터 내의 일부 카테고리와 동일하게 하여 기존 학습된 모델로 성능 평가가 가능하도록 구성한다. Fig. 8과 같이 타겟으로 하는 데이터의 영상은 객체가 소형이고 모션 블러 및 회전 등의 특성이 포함되어 있다. 이러한 특성이 적용된 DOTA, VisDrone으로 학습된 모델을 통해 실시간 및 인식 성능을 도출하기 위해 자체 데이터셋은 학습 데이터로 사용되지 않고 테스트 목적으로 활용된다.

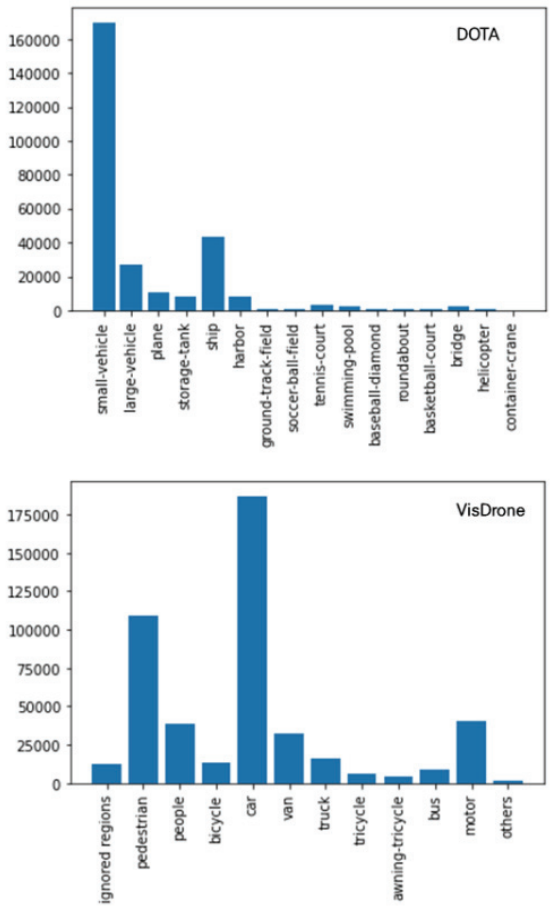

Fig. 7. The number of objects in Aerial object detection dataset

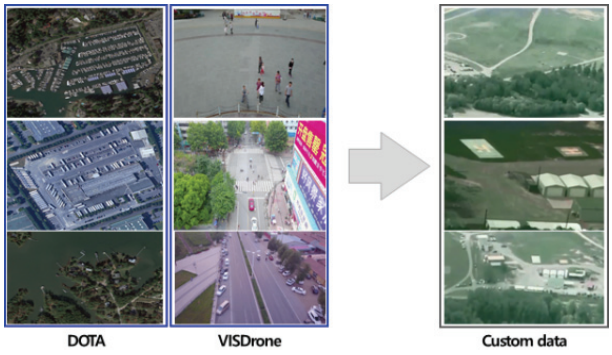

Fig. 8. Aerial Object detection dataset

2.2.2 평가 지표

다중 객체 인식 알고리즘의 성능을 측정하는 평가 지표로는 일반적으로 mAP(Mean Average Precision)와 mAR(Mean Average Recall)이 COCO, PASCAL VOC 등의 챌린지 대회에서 많이 사용된다[15]. 본 논문에서도 제안한 객체 인식 알고리즘의 성능을 분석하기 위해 mAP와 mAR을 사용하여 결과를 제시하였고, COCO 객체 인식 챌린지에서 평가 지표를 측정한 방안을 벤치마크 하여 Table 1과 같이 정의하였다[16].

Table 1. Performance metrics for object detection model

Metric		Description
Average Precision	AP	• AP is evaluated with different IoUs. • AP can be calculated for 10 IoUs varying in a range of 50% to 95% with steps of 5% (reported as AP@50:5:95)
	AP_{50}	• AP also can be evaluated with single values of IoU • AP at IoU=0.50
AP Across Scales	AP_s	• AP for small objects with area < 32^2 pixels
	AP_m	• AP for medium objects with 32^2 < area < 96^2 pixels
	AP_l	• AP for large objects with area > 96^2 pixels
mAP		• Mean AP over all object classes
mAP_{50}		• mAP at IoU=0.50
mAR		• Average of AR
FLOPs		• The number of FLOPs (Floating point Operations per Second) for a single forward-pass
Inference time		• Inference time of model

AP는 참과 거짓을 분류하는 인식 모델에서 참이라고 분류한 샘플 중 실제 정답이 참인 샘플의 비율을 뜻하며, 실제 정답이 참인 것을 판단하는 기준으로 IoU(Intersection over Union)를 사용한다. 객체 인식 분야에서 IoU는 수식 (4)와 같이 인식 모델이 예측한 바운딩 박스 영역(*BBOX*)과 실제 정답의 바운딩 박스 영역(*Groundtruth*)간의 합친 영역과 겹치는 영역의 비율로 계산한다.

$$IoU = \frac{BBOX \cap Groundtruth}{BBOX \cup Groundtruth} \quad (4)$$

이와 같이 계산된 IoU 값이 주어진 임계치 값보다 크면 참으로 판단한다. 임계치 값이 클수록 참인지를 판단하는 기준치가 높아지게 된다. AP는 IoU가 50%부터 95%까지 5%씩 증가하면서 총 10개의 IoU에서 측정된 AP를 평균 내어 계산하며, 특정 IoU에 한해서만도 측정 가능하며 보통 50%, 75%를 많이 사용한다. 본 논문에서는 드론 영상 특성상 객체의 크기가 작기 때문에 예측되는 영역의 크기가 작다는 점을 고려하여 IoU가 50%인 AP에 대해 측정하였다. 다중 객체 인식 알고리즘의 경우 모든 클래스의 AP에 대한 평균값을 계산한 mAP를 활용하여 알고리즘의 성능을 평가한다.

또한 AP는 객체 크기에 따라서 AP_s, AP_m, AP_l로 측정지표를 구분할 수 있으며, AP_s는 32x32 픽셀 이하의 크기를 가지는 객체, AP_m은 32x32 픽셀보다는 크고 96x96 픽셀보다는 작은 크기를 가지는 객체, AP_l은 96x96 픽셀 이상의 크기를 가지는 객체들에 대해 계산한다.

AR은 최대 N개의 객체가 존재하는 영상별 평균 재현율을 의미하며, 이 값들의 평균을 계산한 mAR을 통해 알고리즘에서 실제 정답의 얼마나 검출됐는지를 분석하기 위해 활용한다. 또한 실시간성을 비교하는 지표로 전체 파라미터 개수와 모델의 단일 영상 추론 시간을 활용한다.

2.2.3 실험 환경

실험에서 사용하는 하드웨어 사양은 Table 2와 같다. 워크스테이션에서 객체 인식 모델의 학습을 진행하고 학습 완료된 모델의 성능 평가를 수행한다. 온-보드 하드웨어는 드론 탑재를 고려하여 저전력, Industrial급 사양인 TX2i를 활용한다. TX2i에서 기학습된 모델을 가속화된 추론 엔진용 모델로 변환한 후 인식 성능 및 실시간 성능을 확인한다.

2.2.4 실험 결과 및 성능 비교

Figure 1에서 제안한 온-보드에서 구동 가능한 딥러닝 기반 객체 인식 모델 개발 방법에 따른 실험 결과를 아래와 같이 제시한다. Table 3-5는 학습모델에 대한 성능 평가로서 워크스테이션에서 수행한 결과이며, 실험에 사용된 워크스테이션 사양은 Table 2와 같다.

Table 3은 드론 영상을 기준으로 객체 인식 모델의 성능을 비교 및 선정하기 위해 1/2단계 인식 모델에서 각각 대표적인 알고리즘인 YOLOv3와 FRCNN (Faster R-CNN)을 활용하여 DOTA 및 VisDrone 데이터를 통해 학습한 결과를 도출하였다. 이 과정에서는 데이터 증강, 전이 학습 및 클래스 불균형 등에 대한 기술은 적용하지 않고 드론 영상 데이터에서 기존 방식으로 구현된 모델 간의 성능 비교를 목적으로 한다.

인식 성능의 척도가 되는 mAP에 대해 모델 간 비교를 수행하였고 FRCNN이 DOTA와 VisDrone에서 각각 mAP 35.1%, 28.9%를 보이고 YOLOv3와 비교하여 DOTA는 약 2.95배 이상, VisDrone에서 약 1.36배

Table 2. Specification of hardware

Spec	Hardware	
	Workstation	Jetson TX2i
OS	Linux	Linux
CPU	i7-9700k (4.9GHz)	ARM Quard (1.95GHz)
GPU	RTX 2080	256 CUDA cores
RAM	64GB	8GB
HDD	1TB	32GB

Table 3. Object detection results on the Aerial Dataset

DOTA						
Method	mAP_{50}[%]	mAP_s[%]	mAP_m[%]	mAP_l[%]	mAR[%]	FLOPs
FRCNN	35.1	11.5	35.0	52.2	58.8	215.49G
YOLOv3	11.9	1.1	10.8	18.2	28.1	32.81G
VisDrone						
Method	mAP_{50}[%]	mAP_s[%]	mAP_m[%]	mAP_l[%]	mAR[%]	FLOPs
FRCNN	28.9	18.6	41.9	49.6	63.0	215.47G
YOLOv3	21.2	11.2	35.3	44.0	59.3	32.80G

Table 4. Object Detection results on the VisDrone dataset

Method	mAP_{50}[%]	AP_{50}[%]									
		ped.	peo.	bic.	car	van	truck	tri.	awn.	bus	mot.
YOLOv3	25.24	**15.9**	9.5	7.0	**63.7**	**29.8**	**32.6**	13.2	12.0	51.5	**17.2**
Our method	**25.28**	13.1	**11.8**	**8.7**	60.9	27.7	31.3	**16.5**	**15.1**	**52.8**	14.9

(The names of the VisDrone 10 object classes are abbreviated as follows: ped. – pedestrian, peo – people, bic – bicycle, car – car, van – van, truck – truck, tri – tricycle, awn. – awning tricycle, bus – bus, mot – motor.)

Table 5. Object detection results on the Custom dataset

Method	mAP_{50}[%]	AP_{50}[%]		
		ped.	car	truck
YOLOv3	22.5	12.5	43.2	11.7
Our method	**24.2**	**13.6**	**45.0**	**13.9**

이상 높다. 영상 내 존재하는 전체 객체에 대한 인식률을 나타내는 mAR은 FRCNN이 DOTA 58.8%, VisDrone 63.0%를 보이며 테스트셋에 존재하는 개별 영상 내에서 평균적으로 절반 이상의 객체를 탐지할 수 있음을 알 수 있다.

FRCNN은 소형 객체 인식률인 mAPs가 DOTA에서 11.5%, VisDrone에서 18.6%로 YOLOv3에 비해 모두 높은 것은 알 수 있다. 2단계 인식 모델인 FRCNN은 영역 추천 네트워크가 별도로 존재하기 때문에 1단계 인식 모델보다 정밀한 객체 위치추론이 가능하며 결과도 동일함을 알 수 있다.

전반적인 인식 성능은 FRCNN이 높지만 모델 구동에 필요한 부동소수점 연산을 나타내는 FLOPs는 약 215G로 YOLOv3의 약 32G보다 6.72배 이상의 연산이 필요하다. YOLOv3가 FRCNN보다 모델 추론 시간이 더 빠른 것을 알 수 있다. 1단계 인식 모델은 별도의 영역 추천 네트워크 없이 하나의 파이프라인에서 객체 위치 추정과 분류를 동시에 수행하기 때문에 2단계 인식 모델에 비해 모델의 크기가 작아 모델 구동에 필요한 연산량이 적은 것을 알 수 있다.

Table 3의 결과를 통해 객체 인식 모델을 선정하는 기준으로 실시간성과 인식 성능 간의 Trade-off가 존재하기 때문에 드론의 운용 개념과 임무 목적을 고려해야 한다. 드론이 상대적으로 높은 고도에서 운용되며 특정 객체를 정밀하게 인식해야 하는 목적으로는 실시간성은 낮지만 인식 성능이 높은 2단계 인식 모델을 활용해야 하며 중, 저고도에서 운용되며 실시간으로 다수의 객체를 탐지해야 하는 경우는 1단계 인식 모델이 적합하다. 본 논문에서는 감시정찰 분야에서 활용 가능한 드론의 실시간 객체 인식 연구에 중점을 두고 있기 때문에 인식율보다는 실시간성에 좀 더 높은 가중치를 두어 YOLOv3를 기반으로 다음 실험을 진행하였다.

Table 4, 5에서는 기존 YOLOv3 모델로 VisDrone 데이터셋에 대한 학습을 수행하였고 Our method는 DOTA로 사전 학습된 모델을 활용하여 VisDrone 데이터셋으로 전이 학습을 수행하고 학습 과정에서 데이터 증강 및 가중 크로스 엔트로피를 적용하였다.

Table 4에서는 클래스 불균형이 존재하는 VisDrone 데이터셋 기준으로 기존 방법과 제안하는 방법의 성능을 도출한 실험 결과이다. 기존 YOLOv3 모델과 2.1.4절에서 제안한 가중 학습을 진행한 모델의 결과를 보면 샘플 수가 적은 people, bicycle, tricycle, awning-tricycle, bus 클래스에 대해 AP가 평균 약 2%정도 증가하였고, pedestrian, car, van, truck, motor 클래스의 AP는 다소 저하되는 결과를 보였지만, 전체 클래스에 대한 mAP는 기존 YOLOv3 모델에 비해 제안한 모델이 0.04% 증가하였다. 제안하는 방식은 기존 YOLOv3 모델과 동등 이상의 성능을 보이면서 클래스 별 인식 성능의 편차는 상대적으로 줄어드는 것을 알 수 있다.

Table 5는 학습에 참여하지 않은 자체 구축 데이터를 평가 데이터로만 사용하여 기존 YOLOv3 모델과 제안 방법의 성능 평가를 수행한 결과이다. 자체 구축 데이터는 기존 학습 데이터와 비교하여 해상도, 화질 등이 다르기 때문에 모델의 학습 과정에서 활용된 입력 영상과는 차이가 존재한다. 드론을 실제로 운용하며 기상상황의 변화나 통신상태에 따라 영상의 품질이 저하될 수 있기 때문에 정제된 벤치마크 데이터의 테스트셋이 아닌 새로운 영상에 대한 테스트셋을 기준으로 모델의 인식 성능을 비교하고자 하였다. 자체 구축 데이터는 VisDrone 데이터에 존재하는 클래스명과 동일하게 설정하여 모델 평가가 가능하게 하였다. 기존 방식으로 학습을 진행한 YOLOv3 모델의 mAP는 22.5%이다. 제안하는 방법으로 학습된 YOLOv3 모델은 mAP가 24.2%로 기존 대비 1.7% 증가한 것을 알 수 있다. Fig. 8과 같이 평가에 사용되는 자체 데이터는 학습 데이터에 비해 상대적으로 낮은 해상도와 노이즈가 포함된 영상으로 인식에 어려움을 주는 요소가 포함되어 있지만 개선된 mAP를 통해 모델의 인식 성능을 확인할 수 있었다.

추가로 Table 4와 Table 5를 결과를 연계하여 실험 결과를 분석하였다. Table 4에서는 제안 방법이 mAP 0.4%로 성능 향상 정도가 크지 않지만 학습에 참여하지 않은 일반적인 영상 데이터에서 동일 모델로 mAP 1.7%의 성능 향상이 이루어진 것을 알 수 있다. 또한, Table 4에서 다수 클래스인 Pedestrian, Car, Truck에 대하여 제안 방식의 mAP가 기존 YOLOv3보다 모두 낮았지만 Table 5에서 제안 방식의 해당 클래스들에 대한 mAP는 기존 YOLOv3와 비교하여 높아진 것을 알 수 있다. Table 4에서 VisDrone 데이터는 학습과 평가 데이터가 구분되어 있지만 두 데이터는 동일한 촬영 조건(해상도, 드론 고도 등)으로 획득되었기 때문에 기존 방식과 제안 방식으로 충분히 학습된 모델은 동일 데이터셋에서 유사한 성능을 보였다. 그러나 학습에 참여하지 않고 학습 영상과 다른 촬영 조건으로 구축된 자체 데이터로 평가한 Table 5에서는 데이터 증강 방법과 전이 학습 적용으로 그 성능 차이가 더 큰 것을 알 수 있고 본 실험 결과로 제안하는 방식으로 개발된 모델이 새로운 영상에 대하여 더 강건한 성능을 보이는 것을 알 수 있다.

Table 6은 제안 방법으로 학습을 진행하여 학습 완료된 모델에 TensorRT를 기반으로 양자화 및 캘리브레이션을 수행하여 추론 엔진을 생성하고 모델의 인식 성능 및 추론 시간을 VisDrone 데이터셋 기준으로 도출하였다. 또한 Table 2와 같은 사양의 워크스테이션과 온-보드 상에서 가속화된 모델을 구동하여 각각의 성능을 도출하였다.

기학습 모델은 FP32로 연산을 처리하며 인식 성능은 21.2%, 한 프레임 당 평균적인 추론 시간은 워크스테이션에서 약 0.031초, TX2i에서 약 0.332초이다.

Table 6. Inference acceleration results in Our method based on YOLOv3 on the VisDrone dataset

Method	mAP_{50} [%]	Inference time(s)	
		Workstation	Jetson TX2i
Our Method (FP32)	25.3	0.031	0.332
Our Method +Quantization (FP16)	24.4	0.016	0.082
Our Method +Quantization (INT8)	18.4	0.013	0.074

FP16으로 양자화를 수행한 모델의 경우 인식 성능은 20.3%로 기존 모델보다 1.2% 하락했지만 추론 시간은 TX2i에서 약 0.082초로 약 4배 증가하였다. INT8의 경우 인식 성능은 14.3%로 FP32 대비 약 7% 감소하였다. 속도는 FP16과 비교하여 약 0.008초 가속되었다. INT8의 급격한 성능 하락은 캘리브레이션 수행 시 타겟 데이터를 고려해서 기존 학습 데이터의 분포가 충분히 캘리브레이션 과정에 반영되어야 하지만 학습 데이터양이 부족하여 성능 저하가 발생한 것으로 보인다. 결과적으로 FP32 대비 FP16, INT8로 변환된 추론 모델은 기존 모델과 비교하여 mAP가 낮아지는 결과를 보이지만 온-보드 디바이스에서 실시간성이 크게 향상됨을 알 수 있다.

Ⅲ. 결 론

본 논문에서는 온-보드에서 구동 가능한 딥러닝 기반의 객체 인식 모델을 개발하는 방법을 제안하였다. 드론은 다양한 고도에서 운용되며 이에 따라 촬영된 영상 내의 객체의 크기는 매우 작고 드론의 기동으로 인한 모션 블러가 존재한다. 또한 학습 데이터에 존재하는 클래스 불균형으로 인해 인식 모델의 전반적인 성능이 낮아지는 문제가 있다.

학습 데이터 전처리 단계에서 특정 객체의 외형적인 특징을 보존하기 위해 영상의 종횡비를 유지하며 모델의 입력 영상 크기로 조정하였다. 새로운 영상에 대한 성능을 높이고 부족한 학습 데이터를 보완하기 위해 회전, 반전, 채널 시프트, 압축 및 블러 등을 적용하여 데이터를 증강하고 이를 학습 데이터로 사용하였다.

객체 인식 모델의 구조를 분석하였고 드론의 운용 개념이나 임무 목적에 따라 정확도와 실시간성이 중요해지기 때문에 이를 모두 고려하여 인식 모델 두 가지를 선정하였다. 객체 인식 모델의 구조를 backbone, neck, head로 분할하고 학습 데이터 특성에 따라 전이

학습방법을 선정하여 모델의 가중치를 미세 조정하는 방법으로 추가 데이터를 학습하였다. 그 결과, 학습에 참여하지 않은 자체 구축 데이터에서 전이학습과 증강이 적용된 모델은 기존 모델과 비교하여 높은 인식 성능을 보였다. 학습 데이터의 클래스 간 샘플 수 차이로 발생하는 클래스 불균형을 줄이는 방안으로 오차함수에 클래스 가중치를 반영한 크로스 엔트로피 오차를 활용하여 소수 클래스에 대한 성능 향상으로 모델의 전반적인 mAP를 높일 수 있었다. 제한된 리소스에서 개발된 객체 인식 모델을 활용하기 위해 추론 가속화를 수행하였다. 온-보드 하드웨어인 TX2i에서 기존 모델에 양자화 기법을 적용하여 기존 대비 추론 속도가 향상됨을 보였다.

드론 운용자를 보조하기 위해 지상 통제소에서 직접 고성능의 딥러닝 모델을 구동하는 것은 전파 지연, 통신 대역폭 제한 등의 제약사항이 존재한다. 온-보드 기반 방법은 이러한 문제로부터 자유롭다는 장점이 있지만 딥러닝 모델 구동에 필요한 리소스가 제한적인 단점이 있다. 본 연구에서는 온-보드에서 구동 가능한 객체 인식 모델의 개발 방법을 제안하였다. 드론에 탑재 가능한 하드웨어를 기반으로 실험 결과를 도출하였고 활용 가능성을 보였다. 향후 연구로는 클래스 불균형 문제에 있어 소수 클래스의 성능을 개선하면서 다수 클래스의 성능을 유지할 수 있는 방안을 연구할 계획이다. 또한, 경량화 및 가속화 과정에서 모델의 인식 성능이 저하되기 때문에 이를 개선하기 위해 향후 다양한 가지치기(Pruning) 기법, 모델 증류(Model Distillation) 및 경량화 모델 재학습 방법 등의 추가 연구를 통해 모델의 구조적인 최적화 및 경량화에 따른 성능 저하를 최소화하는 방안을 연구할 계획이다.

References

1) Jiao, L., Zhang, F., Liu, F., Yang, S., Li, L., Feng, Z. and Qu, R., “A survey of deep learning-based object detection,” *IEEE Access*, Vol. 7, 2019, pp. 128837~128868.

2) Vaddi, S., “Efficient object detection model for real-time UAV applications,” *Doctoral dissertation Iowa State University*, 2019.

3) Lee, J., Wang, J., Crandall, D., Šabanović, S. and Fox, G., “Real-time, cloud-based object detection for unmanned aerial vehicles,” *International Conference on Robotic Computing (IRC) IEEE*, April 2017, pp. 36~43.

4) Du, D., *et al*, “The unmanned aerial vehicle benchmark: Object detection and tracking,” *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 370~386.

5) Xia, G. S., *et al*, “DOTA: A large-scale dataset for object detection in aerial images,“ *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3974~3983.

6) Du, D., et al, “VisDrone-DET2019: The vision meets drone object detection in image challenge results,“ *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019.

7) Panchapagesan, S., Sun, M., Khare, A., Matsoukas, S., Mandal, A., Hoffmeister, B. and Vitaladevuni, S., “Multi-task learning and weighted cross-entropy for DNN-based keyword spotting,“ *Interspeech*, Vol. 9, September 2016, pp. 760~764.

8) Redmon, J. and Farhadi, A., “Yolov3: An incremental improvement,“ *Computer Vision and Pattern Recognition*, April 2018.

9) Ren, S., He, K., Girshick, R. and Sun, J., “Faster r-cnn: Towards real-time object detection with region proposal networks,“ *Advances in Neural Information Processing Systems*, Vol. 28, 2015, pp. 91~99.

10) Shorten, C. and Khoshgoftaar, T. M., “A survey on image data augmentation for deep learning,“ *Journal of Big Data*, Vol. 6, No. 1, 2019, pp. 1~48.

11) Yosinski, J., Clune, J., Bengio, Y. and Lipson, H., “How transferable are features in deep neural networks?,” *arXiv preprint arXiv:1411.1792*, 2014.

12) Salman, H., Ilyas, A., Engstrom, L., Kapoor, A. and Madry, A., “Do adversarially robust imagenet models transfer better?,” *arXiv preprint arXiv:2007.08489*, 2020.

13) Hanif, A. and Azhar, N., “Resolving class imbalance and feature selection in customer churn dataset,” *International Conference on Frontiers of Information Technology (FIT)*, 2017, pp. 82~86.

14) Yoo, S. M., Lee, K. H., Park, J., Yoon, S. J., Cho, C., Jung, Y. J. and Cho, I. Y., “Trends in Deep Learning Inference Engines for Embedded Systems,” *Electronics and Telecommunications Trends*, Vol. 34, No. 4, 2019, pp. 23~31.

15) Padilla, R., Netto, S. L. and da Silva, E. A., “A survey on performance metrics for object-detection algorithms.,” *International Conference on Systems, Signals and Image Processing (IWSSIP)*, 2020, pp. 237~242.

16) Lin, T. Y., et al, “Microsoft coco: Common objects in context,” *European Conference on Computer Vision*, 2014, pp. 740~755.