TUNNEL & UNDERGROUND SPACE Vol.33, No.4, 2023, pp.281-298
https://doi.org/10.7474/TUS.2023.33.4.281 ISSN: 1225-1275(Print) ISSN: 2287-1748(Online)

ORIGINAL ARTICLE

딥러닝 기반 달 표면 모사 환경 실시간 객체 인식 및 매칭 시스템 개발

나종호[1], 공준호[2], 이수득[3], 신휴성[4*]

[1]한국건설기술연구원 미래스마트건설연구본부 학생연구원, [2]한국건설기술연구원 미래스마트건설연구본부 박사후연구원,
[3]SK에코플랜트 Eco Lab센터 프로, [4]한국건설기술연구원 미래스마트건설연구본부 선임연구위원

Development of System for Real-Time Object Recognition and Matching using Deep Learning at Simulated Lunar Surface Environment

Jong-Ho Na[1], Jun-Ho Gong[2], Su-Deuk Lee[3], and Hyu-Soung Shin[4*]

[1]Student Researcher, Department of Future & Smart Construction Research, Korea Institute of Civil Engineering and Building Technology
[2]Postdoctoral Researcher, Department of Future & Smart Construction Research, Korea Institute of Civil Engineering and Building Technology
[3]Pro, Eco Lab Center, SK Ecoplant
[4]Senior Research Fellow, Department of Future & Smart Construction Research, Korea Institute of Civil Engineering and Building Technology

*Corresponding author: hyushin@kict.re.kr

Received: August 16, 2023
Revised: August 21, 2023
Accepted: August 22, 2023

ABSTRACT

Continuous research efforts are being devoted to unmanned mobile platforms for lunar exploration. There is an ongoing demand for real-time information processing to accurately determine the positioning and mapping of areas of interest on the lunar surface. To apply deep learning processing and analysis techniques to practical rovers, research on software integration and optimization is imperative. In this study, a foundational investigation has been conducted on real-time analysis of virtual lunar base construction site images, aimed at automatically quantifying spatial information of key objects. This study involved transitioning from an existing region-based object recognition algorithm to a boundary box-based algorithm, thus enhancing object recognition accuracy and inference speed. To facilitate extensive data-based object matching training, the Batch Hard Triplet Mining technique was introduced, and research was conducted to optimize both training and inference processes. Furthermore, an improved software system for object recognition and identical object matching was integrated, accompanied by the development of visualization software for the automatic matching of identical objects within input images. Leveraging satellite simulative captured video data for training objects and moving object-captured video data for inference, training and inference for identical object matching were successfully executed. The outcomes of this research suggest the feasibility of implementing 3D spatial information based on continuous-capture video data of mobile platforms and utilizing it for positioning objects within regions of interest. As a result, these findings are expected to contribute to the integration of an automated on-site system for video-based construction monitoring and control of significant target objects within future lunar base construction sites.

Keywords: Artificial intelligent, Object recognition, Object matching

초록

달 현지 탐사를 위해 무인 이동체에 대한 연구가 지속적으로 이루어져 있으며 달 지상 관심 지역의 정확한 위치 및 맵핑을 위한 실시간 정보화 작업이 요구되고 있다. 딥러닝 영상 처리 분석 기술을 실제 로버에 적용하기 위해 소프트웨어의 통합과 최적화에 대한 연구가 필요하며 본 연구에서는 가상의 달 기지 건설 현장의 영상을 실시간 분석하여 핵심 객체의 공간 정보를 자동으로 수치화하는 방안에 대한 기초 연구가 진행되었다. 본 연구를 통해 이미 구축된 영역 분할 기반 객체 인식 알고리즘을 경계 상자 기반 객체 인식 알고리즘으로 변경하여 객체 인식 정확도 및 추론 속도를 개선하는 작업이 이루어졌으며, 대용량 데이터 기반 객체 매칭 학습을 위해 Batch Hard Triplet Mining 기법을 도입하고, 학습 및 추론에 대한 최적화 연구가 수행되었다. 또한 개선된 객체 인식 및 동일 객체 매칭 소프트웨어를 통합하고, 입력 이미지 내 동일 객체 자동 매칭을 시각화하는 소프트웨어를 개발하였으며, 위성 모사 촬영 영상 내 객체를 학습 데이터로, 이동체 촬영 영상 내 객체를 추론 데이터로 사용하여 동일 객체 매칭의 학습 및 추론이 이루어졌다. 본 연구의 결과는 이동체의 연속 촬영 영상을 기반 3차원 공간 정보를 구현 및 관심 공간 내 객체 위치 설정에 활용할 수 있을 것으로 사료되며, 향후 달 기지 건설 현장에서의 영상 기반 시공 모니터링 및 제어를 위한 자동 현장 및 주요 대상물 공간 정보 구축 시스템과의 연계에 기여할 것으로 기대된다.

핵심어: 인공지능, 객체 인식, 객체 매칭

1. 서 론

딥러닝 영상 처리 분석 기술을 실제 달 현장 내 이동체에 적용하기 위해서는 객체 인식 및 동일 객체 매칭 소프트웨어의 개선 작업과 이를 통합하여 일련의 프로세스로 구축하는 과정이 요구된다. 또한, 통합된 소프트웨어가 핵심 객체에 대한 공간정보를 자동으로 수치화하기 위해서 실시간 분석이 필요하며, 이를 위한 기초연구를 선행해야한다. 통합된 프로세스 내 각 파트(객체 인식, 동일 객체 매칭)별 성능 향상을 위해서는 추가 딥러닝 학습용 데이터셋을 수집 및 구축하는 작업이 필요하다. 본 연구에서는 객체 인식 및 동일 객체 매칭의 성능 향상을 위한 딥러닝 학습용 빅데이터를 확보 및 구축하였고, 로버 측면 영상과 드론 측면 간 동일 객체에 대한 매칭하는 연구를 Triplet 네트워크를 적용하여 실험적으로 분석하였다.

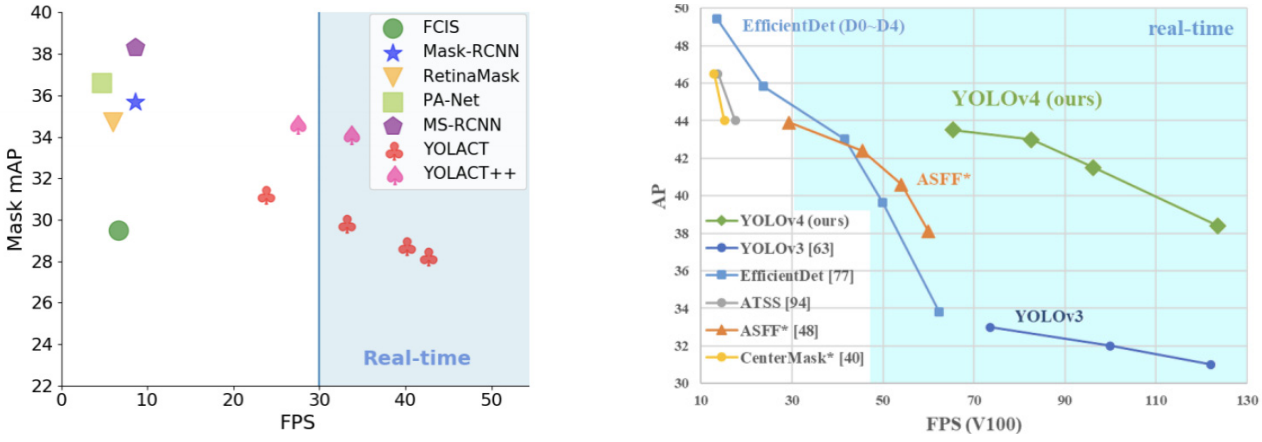
컴퓨터 비전(Computer Vision)분야에서 이미지 내 객체를 인식하는 방식은 Fig. 1처럼 크게 두 가지 형태로 나뉜다. 구체적으로는 이미지 내 객체 주변을 사각형으로 표현하는 경계 상자(bounding box)기반의 방식과 객체 형태를 따라 표현하는 영역 분할(segmentation) 방식이 있다.

Fig. 1. Example of object labeling : Bounding Box (left), Segmentation (right) (Wu et al., 2020)

경계 상자 방식은 데이터 레이블링(labeling)을 사각형(rectangular)형태로 표현하는 방식을 말하며, 대표적인 레이블 표현 방식(Annotation) 두 가지가 존재한다. Pascal VOC의 경우, 이미지 내 객체 경계 상자를 좌측상단 꼭짓점(x1, y1)과 우측하단 꼭짓점(x2, y2)으로 표현하고, 이를 XML (eXtensible Markup Language)로 기록한다. 반면, COCO 데이터셋은 객체의 레이블(Label)을 좌측상단 꼭짓점(x1, y1)과 객체의 너비/높이(w, h)로 표현하고, 이를 JSON (JavaScript Object Notation)파일로 기록한다.

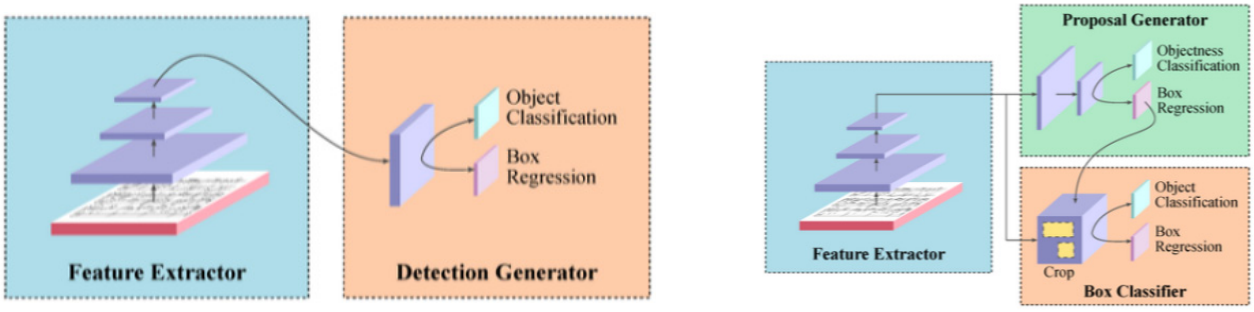
영역 분할 방식은 경계 상자 방식과는 다르게 레이블 정보를 객체의 형태에 맞춰 표현한다. 다시 말해, 객체 형태를 따라서 다수의 지점(point)을 선택하여 여백의 공간(객체 정보가 없는 공간) 없이 레이블이 JSON파일로 기록된다. 정확한 객체 정보를 취득한다는 측면에서는 경계 상자보다는 영역 분할 방식에 강점이 있지만, 이를 위해 많은 연산이 요구되기 때문에 적용 여부를 신중하게 고려할 필요가 있다. 레이블링 방식에서의 차이는 딥러닝 기반 객체 인식 모델의 학습 및 추론 속도에도 영향을 미친다. 영역 분할 기반 객체 인식 모델(Instance Segmentation)은 초당 30장의 이미지를 처리할 수 있는 수준까지 도달하였으나, 추론속도가 증가함에 따라 반대로 정확도가 손실(mAP 30 이하)되는 Trade-off 관계에 있다.

상대적으로 경계 상자 기반 객체 인식 모델(Object Detection)은 2020년 현재 발표된 YOLOv4 기준 초당 최대 100장 이상의 이미지(fps) 처리까지도 가능하며 정확도도 뛰어나다는 강점(mAP 40 이상)이 있다(Fig. 2). 이로 인해 안정적인 성능을 기반으로 한 실시간 처리가 필요한 분야(도시, 건설 안전 분야 등)에서 경계상자 기반 객체 인식 모델이 활발하게 사용되고 있으며, 크게 Two-stage 기법과 One-stage 기법으로 나뉜다(Fig. 3).



(a) Segmentation (b) Object Detection Model

Fig. 2. Performance comparison of object detection by (a) segmentation and (b) object detection model (Bolya et al., 2020)



(a) One-Stage (b) Two-Stage

Fig. 3. Boundary box based object detection method (Pacha et al., 2018)

Two-stage 기반 객체 인식 모델은 객체 위치를 추출하는 부분(Region Proposal)과 어떤 객체인지 분류(Classification)하는 부분으로 구성되어 있다. 대표적인 two-stage 기반 객체 인식 모델은 2015년에 발표된 Faster R-CNN (Ren et al., 2016)으로 sliding window방식을 활용하여 이미지 내 객체가 있을법한 영역을 빠르게 찾아내는 RPN (Region Proposal Network)을 가진 것이 특징이다. 기존 머신러닝 기법(예를 들어, Selective Search)들은 객체로 추정되는 영역들(proposals)을 합성곱 신경망(Convolutional Neural Network)에 입력하기 전에 생성하였다. 이미지 1장에 대해 약 2,000개의 proposals를 만들어 신경망의 입력으로 투입하였다. 이로 인해 이미지 1장의 처리시간이 대폭 증가하는 결과를 초래하였다. 반면, Faster R-CNN은 이미지를 신경망의 입력으로 투입한 이후에 RPN을 통해 proposals를 약 300개 추출하여 처리시간 및 비효율성을 대폭 개선하였다(Tan et al., 2019).

One-stage 기법은 객체 위치를 추출하는 부분(RPN)과 어떤 객체인지 분류(Classification)하는 부분이 나뉘어져있지 않고, 하나의 네트워크로 구성되어 있는 것이 특징이다. 대표적인 One-stage 객체 인식 모델로는 YOLO (You Only Look Once)가 있으며 시간에 따라 다양한 버전으로 개선되어왔다. YOLO계열의 객체 인식 모델은 기본적으로 그리드(Grid)를 기반으로 하여 이미지 내 객체의 위치를 감지한다. 객체의 위치를 추정하는 작업(RPN)과 분류(Classification)을 동시에 진행하기 때문에 Two-stage 기법보다 추론속도(fps) 측면에서 뛰어난 성능을 보여주지만 객체 인식 정확도(mAP) 측면에서 Two-stage 모델과 비교했을 때, YOLO 모델은 낮은 성능을 기록하였다. 하지만, 개선된 버전의 YOLO모델(YOLOv3)이 등장하면서 객체 인식 정확도는 Two-stage 수준으로 높이고, 추론속도는 기존 YOLO모델 이상의 성능을 보여주고 있다(Redmon et al., 2016, 2017, 2018). 최근 YOLOv3에 가우시안 분포(Gaussian Distribution)를 적용한 객체 인식 모델이 등장하면서 One-stage 기법 이상의 안정적인 성능(높은 정확도, 낮은 오작동 발생률)을 낼 수 있게 되었다(Choi et al., 2019).

2. 객체 인식 프로세스 및 데이터셋 구성

2.1 객체 인식 모델 - YOLOv3

본 연구에서는 통합 소프트웨어 개발하고 이를 효율적으로 활용하기 위해 우선적으로 객체 인식 모듈의 개선작업을 수행하였다. 영상/이미지 내 객체 영역을 분할하기 위해 개발된 기존 모델(Mask R-CNN)의 이미지 처리속도는 3 fps에 불과하다. 따라서, 문제를 단순화하고, 이미지 처리속도를 높이기 위해 YOLO 계열의 객체 인식 모델(YOLOv3)을 채택하였다. 또한 기본 YOLOv3 모델이 가지고 있는 문제점인 오탐지(False Positive) 발생률을 감소시키기 위해 객체 인식 모델의 예측값에 가우시안 모델링(Gaussian Modeling)을 적용하였다.

YOLOv3의 전체 네트워크 구조는 객체 특징 추출부(Backbone Network), 객체 위치 추정부(Detection Layer)로 구성되어있다 (Fig. 4). YOLOv3의 객체 특징 추출부(Backbone Network)는 DarkNet53으로 합성곱(Convolution)과 잔차 블록(Residual Block)의 나열로 이루어져있다. 이는 2020년 현재까지도 많이 사용되는 ResNet 구조와 유사한 형태를 띄고 있어, 객체의 다양한 특징을 추출하는데 높은 성능을 내고 있다.

객체 위치 추정부(Detection Layer)는 세 가지 크기의 예측값을 출력한다. 객체 특징 추출부의 최종 출력값을 기반으로 이미지 내 큰 객체를 예측하고, 객체 특징 추출부 중간에서 특징값(features)을 추출하여 중간, 작은 크기의 객체를 예측한다. 이러한 피라미드 형식의 특징 추출(feature pyramid) 방식은 적은 연산으로 성능 향상을 극대화 할 수 있다.

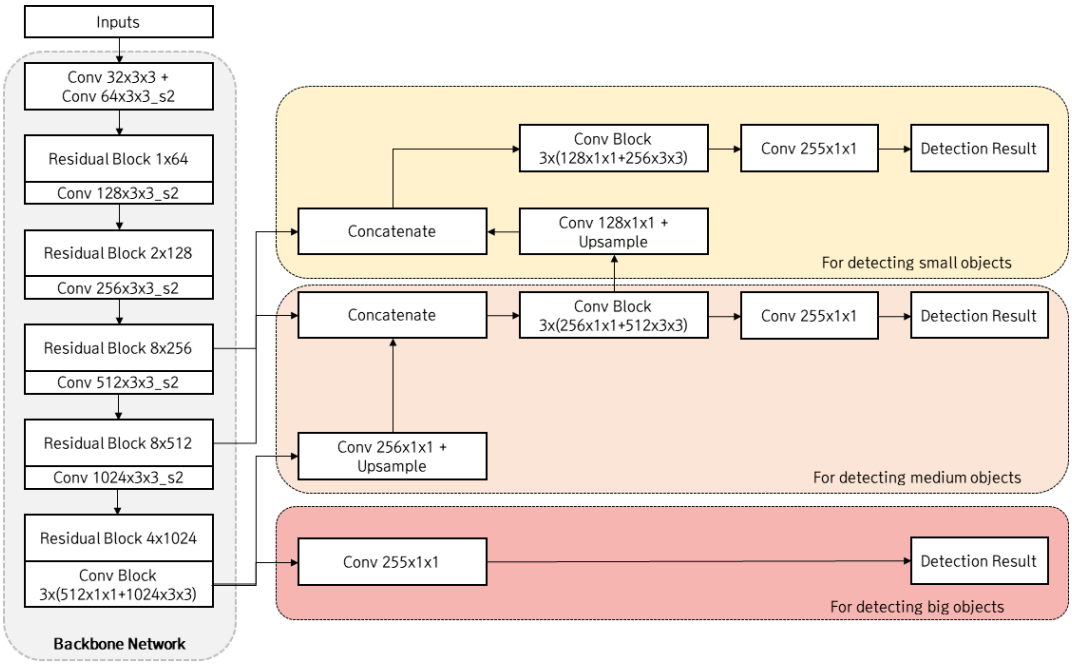


Fig. 4. Conceptual flow of YOLOv3 network

본 연구에서는 YOLOv3의 성능 개선을 위해서 가우시안 분포를 손실함수(loss function)에 반영하였다. Object Detection 알고리즘의 output은 경계상자 좌표(bounding box coordinate), class probability인데, class에 대한 정보는 확률 값으로 나오지만 경계상자 좌표는 deterministic한 값이 출력되기 때문에 bounding box 예측 결과에 대한 불확실성을 알 수 없다. 이에 경계상자 좌표에 가우시안 모델링을 적용하고 손실함수를 재설계하여 모델의 정확도를 높이고 위치 정보의 불확실성(localization uncertainty)을 예측하는 방법을 적용하였다.

YOLOv3의 예측 결과는 각 Grid마다 4개의 경계상자 좌표, 객체 유무 확률(objectness score), 클래스 확률(class score)가 한 묶음이 되어 하나의 상자를 예측하고 나타내게 된다. 경계상자 좌표를 구성하는 t 파라미터들은 예측된 box의 중심좌표, size를 나타내는 값으로 하나의 정해진 값이다. 즉, 객체 유무 확률, 클래스 확률은 확률 값을 나타내어 역치값(threshold) 등을 통해 낮은 확률을 갖는 값들을 필터링 할 수 있지만 경계상자 좌표는 확률값이 아니므로 예측한 상자의 좌표가 얼마나 정확한지 알 수 없다. 이를 해결하기 위해 좌표를 구성하는 파라메터 t 에 Gaussian Modeling을 적용하고, 이에 맞게 손실함수를 수정하였다.

2.2 학습 데이터 구성 및 분석

학습을 위한 데이터셋으로 한국건설기술연구원의 연천 인공 달 모사 시험장에서 Stereo Camera를 이용하여 취득한 로버 주행 영상을 사용하였으며 조도 신뢰성 확보를 위해 주, 야간 영상을 모두 활용하여 Frame단위로 분할하여 사용하였다(Fig. 5). 영상 취득에 사용된 카메라 모델은 FLIR사의 Blackfly S 모델을 2대 활용하였으며, 카메라 캘리브레이션을 통해 스테레오 카메라로 구성하였다. 영상 프레임별로 이미지 자체의 변화가 크지 않아 이전 프레임과 현재 프레임간의 SSIM (Structural Similarity Index Measure)이 0.8 미만인 이미지만 1차로 선별하고 잘못 레이블링이 되었거나 경계 상자가 부정확하게 표기되는 등의 수정작업이 필요한 데이터들은 수작업 선별하여 수정하거나 데이터셋에서 제외하였다.

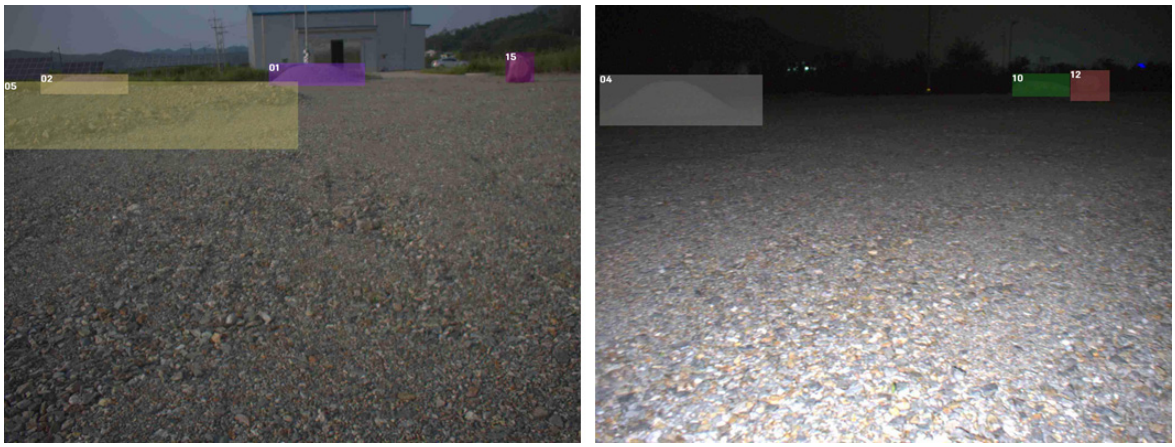


Fig. 5. Data labeling results: day image (left), night image (right)

필터링 및 가공이 완료된 이미지는 모두 51,257장이며(Table 1) 이미지 내 객체들은 크레이터 24,080개, 암석 89,915개, 사면 34,571개로 검출되었다. 객체 인식 모델의 과적합(Over-fitting)을 방지하고, 안정적인 성능을 산출하기 위해 구축된 데이터셋을, 학습(train)/검증(validation) 데이터셋으로 나누었다. 두 데이터셋은 8:2의 비율로 나누었으며(Table 2), 목표 객체의 주/야간 분포를 고려하여 구성하였다. 또한, 효율적인 데이터 관리를 위해, 데이터(이미지/레이블)를 직렬화(Serialization)된 형태의 데이터(.tfrecord)로 변환하였다.

Table 1. Quantities of the datasets in terms of condition

	Day image	Night image	Sum.
No.	34,601	16,656	51,257

Table 2. Train and validation split of dataset

	Training dataset	Validation dataset
Day	27,680	6,921
Night	13,324	3,332
Sum	41,004	10,253

2.3 하이퍼 파라미터(Hyper-Parameter) 설정 및 개발환경 구성

객체 인식 모델의 학습에 필요한 하이퍼 파라미터를 Table 3과 같이 설정하였다. 학습 환경을 고려하여 배치 사이즈(batch size) 및 입력값의 해상도(resolution) 등을 조절하였으며, 이를 위한 파일(.yaml)을 별도로 구성하여 학습 프로세스를 설계하였다. 객체 인식 모델 학습 시 대규모 배치(batch)연산 및 안정적인 학습을 위해 본 연구에서는 Ubuntu 16.04 LTS, Python 3.6.10, Tensorflow 2.1, CUDA 10.1, cuDNN 7.6.5의 개발환경 및 Intel Xeon Gold 5120 CPU, DDR4 256GB 메모리, 3개의 NVIDIA Tesla V100 32GB 하드웨어를 사용하였다.

Table 3. Hyper parameter settings of object recognition model

Hyper parameter	Value
Epoch	50
Batch Size	16
Buffer Size	1,024
Learning Rate	1e-4
Margin	0.5
Height	416
Width	416

총 학습 시간은 약 18시간이 소요되었으며, 매 epoch별로 21분가량이 소요되었다. 반복(iteration)별로 학습 손실값(train loss)와 검증 손실값(validation loss)를 산출하여 기록하였다(Fig. 6). 학습 손실값의 추이를 보면, 40 epochs 이후까지 지속적으로 감소하는 것을 볼 수 있다. 검증 손실값 또한 안정적인 감소 추세를 보여 학습에서 발생할 수 있는 과적합(Overfitting)문제는 해소되었다고 판단하였다.

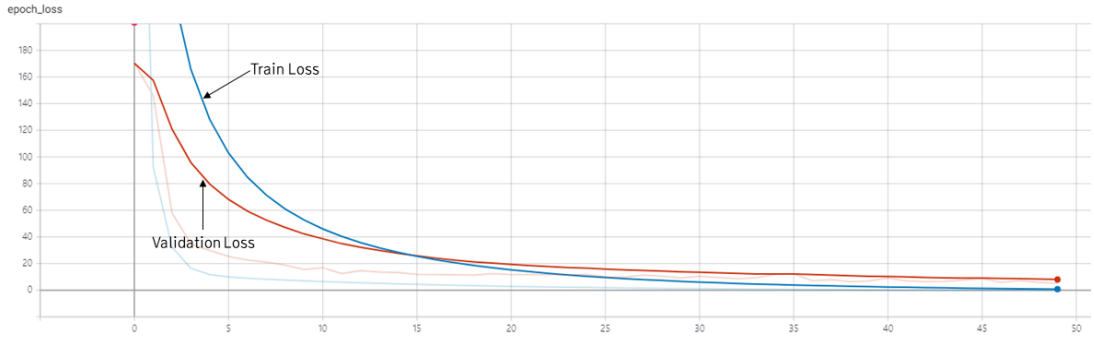


Fig. 6. Train/Validation loss curve

완료된 객체 인식 모델의 가중치를 기반으로 객체 인식 성능을 측정하였다. 성능 평가 척도로는 mAP를 사용하였으며, 검증 데이터에 대해 수행하였다. 객체 인식 성능은 0.87을 기록하였고, 객체별 인식률(AP) 또한 실제 적용 가능한 수준까지 학습된 것을 알 수 있다(Fig. 7).

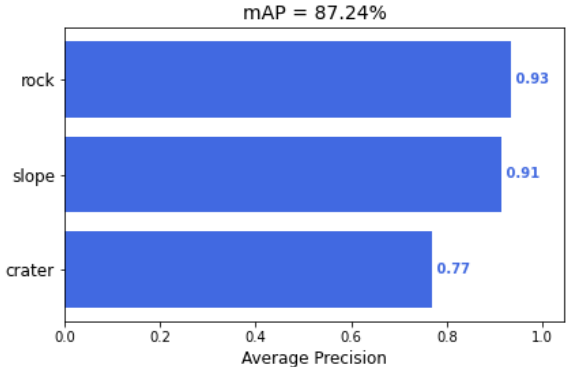


Fig. 7. mAP(mean Average Precision) by classes

3. 객체 매칭 프로세스 및 데이터셋 구성

3.1 Batch-Hard/Online Triplet Mining 기법

이미지셋 사이에 동일 객체를 매칭하기 위해 Triplet Loss를 활용하였고, 그 중에서도 가장 단순한 방식의 손실함수(Loss Function) 형태를 도입하였다. 특정 객체(anchor)와 동일한 클래스의 객체(positive) 사이의 거리, 다른 클래스의 객체(negative)와의 거리를 계산하여 손실값(Loss)을 산출 및 객체 매칭 네트워크 역전파(Backpropagation)을 진행하였다.

Triplet (anchor, positive, negative) 조합을 만들어 네트워크의 입력으로 설정하기 때문에 데이터 수량이 증가함에 따라 조합도 기하급수적으로 증가하게 된다는 한계가 존재한다. 또한, 정해진 조합들의 평균 손실값을 산출하여 학습하다 보니 Fig. 8에서처럼 매 반복(iteration)마다 감소하는 추세는 보이지만 불안정한 상태인 것을 확인하였다.

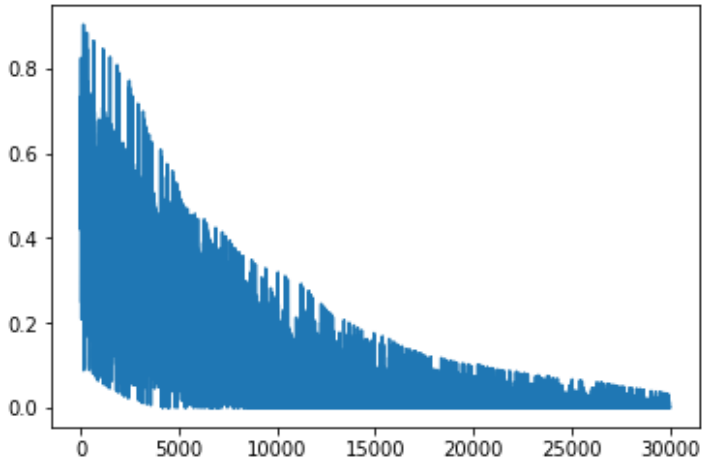


Fig. 8. Initial triplet loss curve

이를 극복하기위해 두 가지 기법을 도입하였다. 먼저, 기하급수적으로 Triplet 조합이 늘어나는 문제를 극복하기 위해 Online Triplet Mining을 추가하였다. Online Triplet Mining은 학습 전에 조합을 미리 만들지 않고, 학습 시 입력으로 들어오는 배치(batch) 내에서 Triplet 조합을 만들어내는 것이 핵심이다(Schroff et al., 2015). 이를 통해 네트워크의 입력(input)은 이미지 분류(Classification) 문제와 동일한 형태의 단일 이미지가 된다. 두 번째로, 효율적인 학습을 위해 Online Triplet Mining을 통해 만들어진 조합 중에 Hard Triplets를 추출하였다. 동일 클래스이지만 anchor와의 거리가 가장 먼 positive (Hard Positive)와 다른 클래스이지만 anchor와의 거리가 가장 가까운 negative (Hard Negative)를 입력 데이터(example)별로 선정하여 평균 손실값을 산출하였다.

추가적으로 딥러닝 네트워크의 학습 시, 모델의 동일 객체 매칭 정확도를 측정하기위해 Batch Hard Accuracy를 지표로 활용하였다. 배치 내에서 동일 객체간 거리가 가장 멀리 떨어져있는 positive (Hard Positive)와 다른 클래스의 객체 중 가장 가까운 negative (Hard Negative)를 비교한다. Hard Positive가 Hard Negative보다 anchor에 가까운 경우에만 정답으로 인정하기 때문에, 기존 정확도 측정방식보다는 가혹한 지표라 할 수 있다. Batch Hard Accuracy는 손실값과 같이 매 반복마다 측정되어 epoch이 종료되면 평균값이 산출된다.

3.2 로버-위성 매칭 학습 데이터셋 구성

로버 객체 이미지와 위성(드론) 객체 이미지 사이의 동일 객체 매칭 학습을 위해 앞서 구축한 로버 영상간 객체 데이터셋과 드론 객체 이미지를 종합하여 학습하였다. 드론 객체 이미지는 고도 30 m 및 50 m 드론 촬영영상을 활용하여 프레임(이미지) 단위로 구분하고, 가공 작업을 통해 학습에 필요한 객체를 추출(cropping)하였다(Fig. 9). 총 30,771장의 객체 이미지를 확보하였으며, 149,627개의 객체가 검출되어 18가지 객체 종류별로 분류하였다(Table 4). 다음 Fig. 10은 18종류에 따른 객체 분포도 결과다. 최종적으로 확보한 데이터셋을 바탕으로 로버와 위성(드론) 각각의 객체 이미지 분포를 고려하여 7:3의 비율로 학습/검증 데이터셋을 분리하였다.

Fig. 9. Sample of drone image

Table 4. Rover-Satellite image object matching trainning dataset

	Day image object	Drone image object	Total
No.	188,856	30,771	149,627

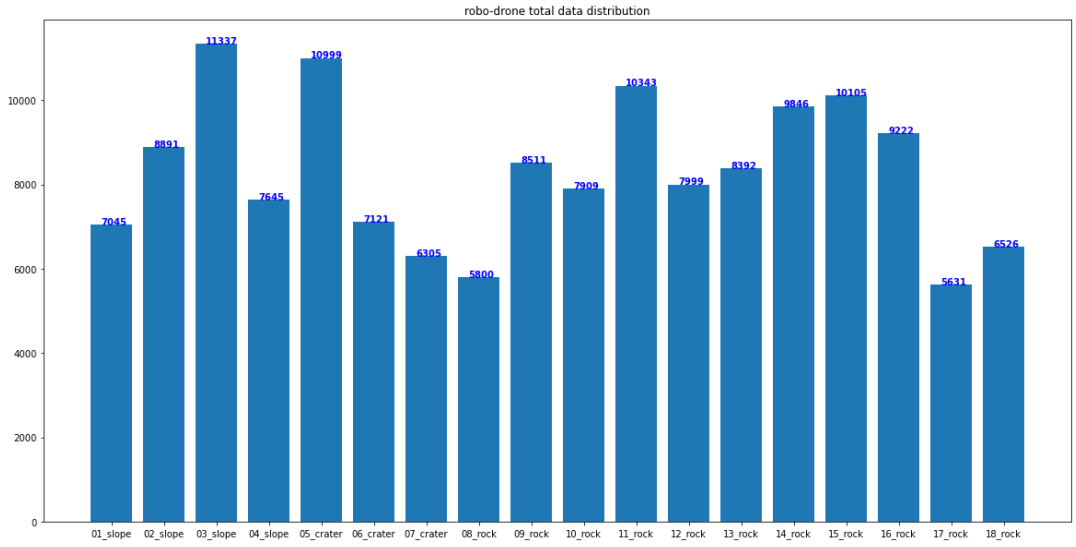


Fig. 10. Object distribution histogram

3.3 로버-위성 매칭 학습 환경 구성 및 결과

로버-위성 객체 매칭 학습 환경은 로버 영상 간 객체 매칭 학습과 동일하게 진행하였다 다만 하이퍼 파라미터 설정은 Table 5 와 같이 변경하였으며 입력값의 크기, 마진 등은 로버 영상 간 객체 매칭 학습과 동일하게 설정하고 원활한 학습을 위해 배치 사이즈(batch size)를 128로 설정하였다.

Table 5. Hyper parameter settings of rover-satellite object matching model

Hyper parameter	Value
Epoch	250
Batch Size	128
Buffer Size	4,000
Learning Rate	1e-6
Margin	0.5
Height	128
Width	128

로버-위성 객체 매칭 학습은 전반적으로 안정적인 학습패턴(수렴속도, 정확도 향상 추이)을 보였다. 손실값은 학습 초반 정체된 모습을 보였으나, 약 80 epoch이후부터 감소하는 추세를 보였고, 약 250 epoch지점에서 수렴되는 것을 볼 수 있다(Fig. 11).

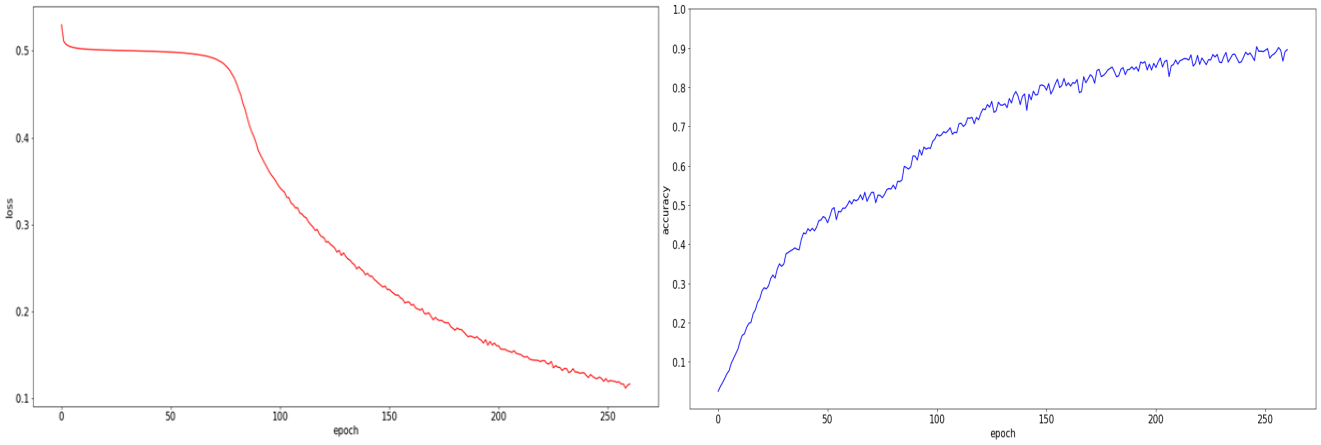


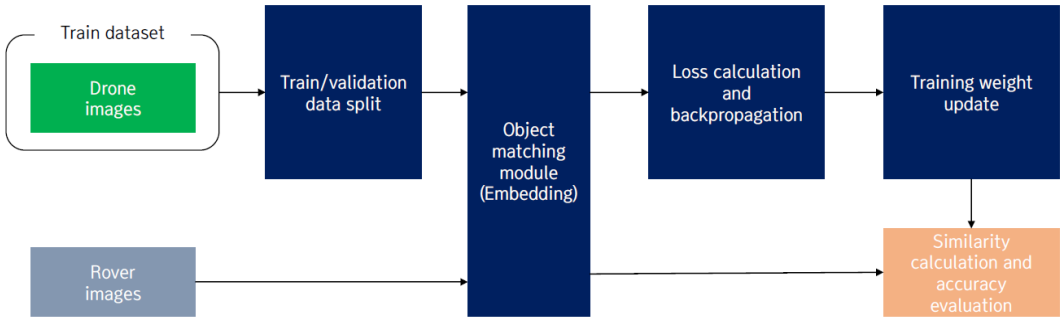
Fig. 11. Rover-Drone object matching training result : Loss trend (left), Accuracy trend (right)

정확도(Batch hard accuracy) 또한 로버 영상 간 매칭 학습 때와 마찬가지로 지속적으로 향상되는 추이를 보였고, 수렴지점인 250 epoch에서 90%이상의 정확도를 산출하였다. 객체 특징값을 3차원 공간에 임베딩(Embedding)시킨 결과, 각 특징값들이 동일 클래스별로 군집화(Clustering)되어 적절하게 학습된 것을 확인할 수 있다.

4. 드론-로버 객체 매칭 학습 프로세스

4.1 학습 프로세스 개요

변형된 로버-위성 이미지간 객체 매칭(이하 드론-로버 객체 매칭)은 실제 현장(달 표면)에서의 적용가능여부를 확인하게 위한 목적으로 실험이 진행되었다. 다음 Fig. 12는 드론-로버 객체 매칭 프로세스를 도식화한 결과다. 드론-로버 객체 매칭은 앞선 로버-위성 이미지간 학습 프로세스와 다르게 드론 객체 데이터만 학습에 사용하고 로버 객체 데이터는 사용하지 않는다. 로버-위성 이미지간 객체 매칭은 로버 객체 데이터셋과 드론(30 m) 객체 데이터셋을 통합하여 하나의 데이터셋으로 구성하고, 이를 입력으로 하여 객체 매칭 학습이 이루어졌다. 반면, 드론-로버 객체 매칭 학습은 드론 객체 데이터만 입력으로 하여 학습하고, 로버 객체 데이터는 추론에 활용하여 성능 평가를 진행하였다.

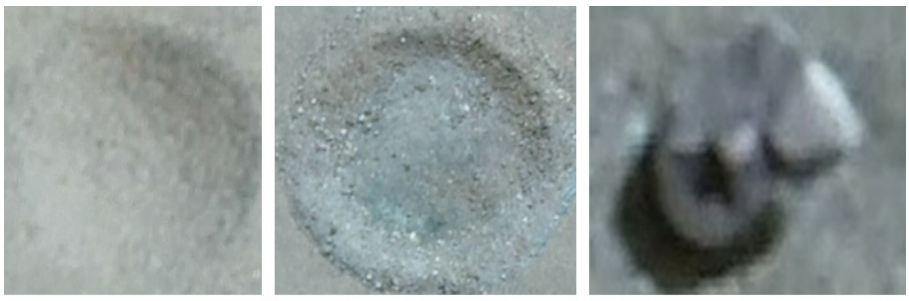


〈Drone-Rover training process〉

Fig. 12. Drone-Rover object matching training process

4.2 데이터셋 가공

먼저, 전체 영상을 프레임(이미지)단위로 나누고 앞뒤 프레임 간 변화량을 통해 유의미한 데이터만 추출하였다. 유사도 평가 모듈(SSIM)을 기반으로 변화량을 측정하여 유사도가 0.9 이하인 이미지만 선별하였다. 결과적으로 총 1,917장의 50 m 드론 이미지를 추출하였으며, 예시는 Fig. 13과 같다. 다음으로, 추출된 이미지 내에서 목표 객체를 레이블링하였다. 총 12,449장의 객체 이미지를 확보하였다(Table 6).



01_slope 06_crater 09_rock

Fig. 13. Example of object in drone image (50 m)

Table 6. The number of object detected from 50 m drone image

Object name	Data num.	Object name	Data num.
01_slope	748	10_rock	866
02_slope	739	11_rock	636
03_slope	748	12_rock	782
04_slope	547	13_rock	562
05_crater	819	14_rock	794
06_crater	820	15_rock	722
07_crater	585	16_rock	702
08_rock	772	17_rock	567
09_rock	530	18_rock	510

30 m 영상은 50 m 촬영 영상과 동일하게 유사도 평가 모듈을 활용하여 이미지 선별을 진행하였으나, 목표 객체의 데이터 불균형(class imbalance)문제가 심하였다. 데이터 불균형 문제를 해소하기 위해 프레임 간 유사도 평가 없이 전체 프레임에 대해 이미지 변환을 진행하였고, 이어서 데이터 가공 작업을 수행하였다. 이를 통해, 총 30,771장의 객체 이미지를 확보하였고, 데이터 불균형 문제가 유사도 평가를 통한 선별작업 이후보다 개선된 것을 다음 Fig. 14를 통해 확인가능하다. 30 m 영상에서 취득한 데이터 현황은 Table 7에서 확인 가능하며, 이에 대한 샘플 영상은 다음 Fig. 15와 같다.

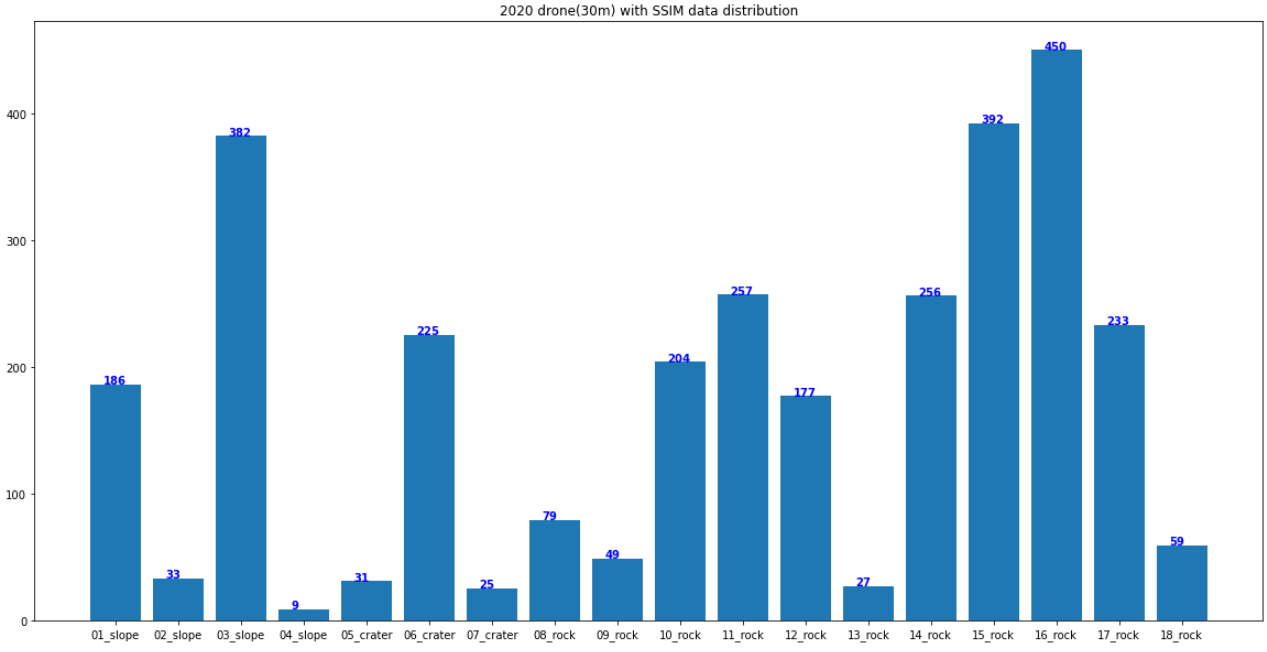


Fig. 14. Object distribution histogram after SSIM filtering process (30 m drone image)

Table 7. The number of object detected from 30 m drone image

Object name	Data num.	Object name	Data num.
01_slope	1,587	10_rock	1,559
02_slope	1,982	11_rock	1,901
03_slope	1,798	12_rock	1,478
04_slope	1,358	13_rock	1,482
05_crater	1,090	14_rock	1,866
06_crater	1,651	15_rock	2,123
07_crater	1,494	16_rock	1,840
08_rock	1,704	17_rock	1,733
09_rock	1,977	18_rock	2,148

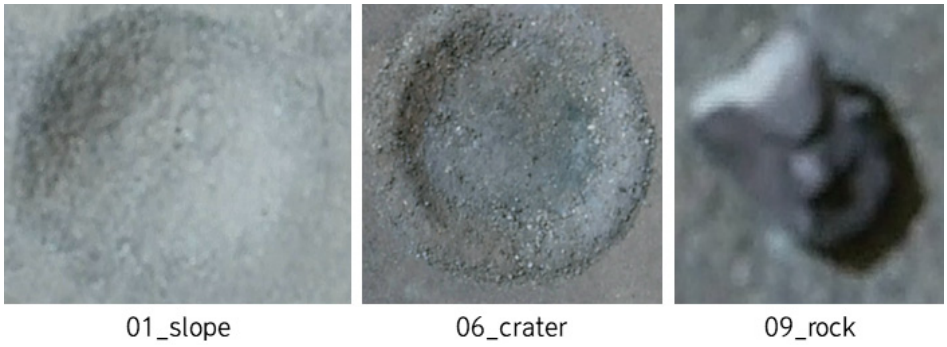
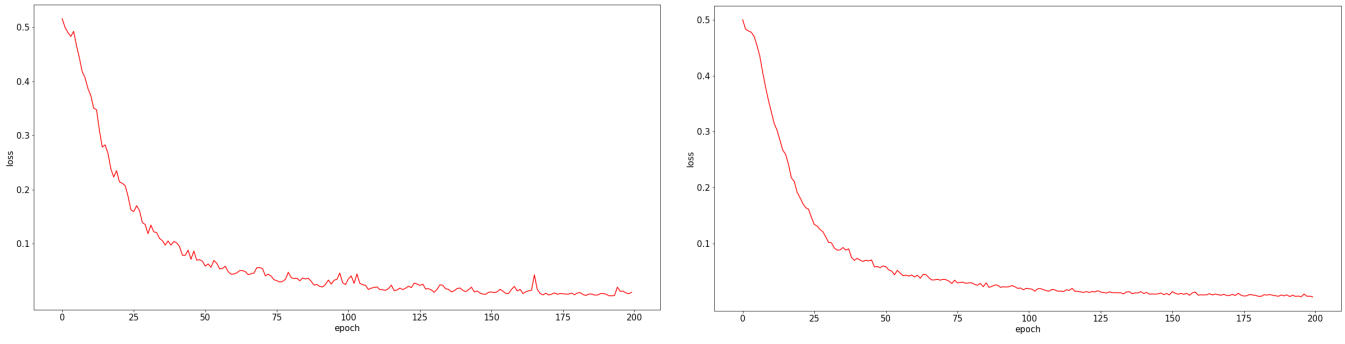


Fig. 15. Example of object in drone image (30 m)

4.3 드론-로버 객체 매칭 학습 결과

50 m 드론 영상의 객체 매칭 학습결과, 100 epoch이후부터 어느 정도 수렴이 이루어진 것을 볼 수 있다. 로버 영상 간, 로버-위성 이미지간 객체 매칭 학습과 달리 학습 초반부터 손실값이 지속적으로 감소하는 추세를 보였다. 30 m 드론 영상의 객체 매칭 학습은 50 m 영상의 학습과 하이퍼 파라미터 설정을 동일하게 하였으나 상대적으로 안정적인 손실값 감소 추이를 보였다(Fig. 16).



(a) (b)

Fig. 16. Trend of loss values in drone image training results : (a) 50 m and (b) 30 m drone image

5. 드론-로버 매칭 추론 프로세스

5.1 추론 프로세스 및 데이터셋 구성

학습된 드론 객체 매칭 모델을 기반으로 로버 객체 데이터를 추론하는 실험은 크게 클래스 간 비교와 범주별 비교 두 가지 프로세스로 진행하였다. 매칭을 위한 드론 데이터는 효율적인 연산을 위해 미리 DB로 구성해두었다. 로버 데이터셋은 전체 가공된 데이터를 주/야간 포함 데이터셋, 주간 데이터셋으로 구성하였다.

5.1.1 클래스간 비교 프로세스

클래스 간 단순 비교 프로세스는 다음 Fig. 17과 같다. 기존 머신러닝 및 딥러닝의 분류모델이 수행하던 정확도 측정방식과 동일하게 진행하였다. 로버 이미지로부터 추출된 객체를 매칭 모듈을 활용하여 128차원의 벡터(vector)로 임베딩시키고, 드론 객체 임베딩 벡터와의 비교(Euclidean Distance 기반 유사도 측정)를 통해 정확도를 측정하였다.

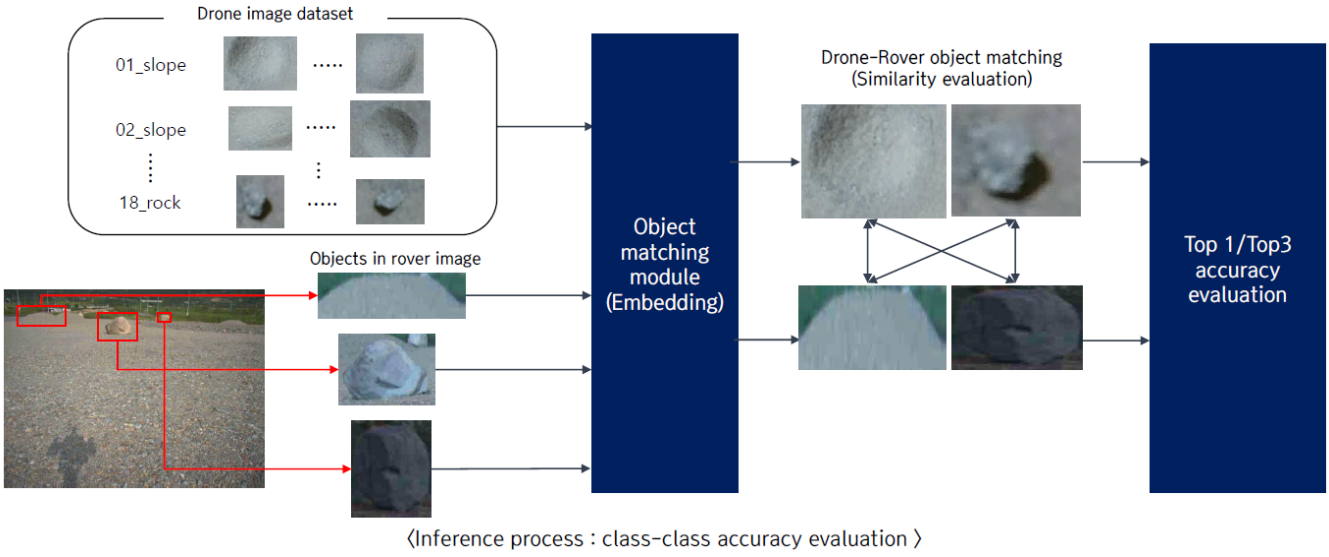


Fig. 17. Conceptual diagram of inter-class detection accuracy measurement process

5.1.2 범주별 비교 프로세스

범주 간 단순 비교 프로세스는 다음 Fig. 18과 같다. 사전에 정의된 목표 객체의 클래스는 세 가지 범주(slope, crater, rock)에서 파생된 것이며, 이 중에서도 slope와 crater는 드론 이미지 상에서 상당히 유사한 형태로 나타난다. 동일 범주끼리만 유사도를 측정하여 정확도를 산출한다면, 보다 높은 정확도를 산출할 수 있기 때문에 범주별 정확도 측정을 수행하였다.

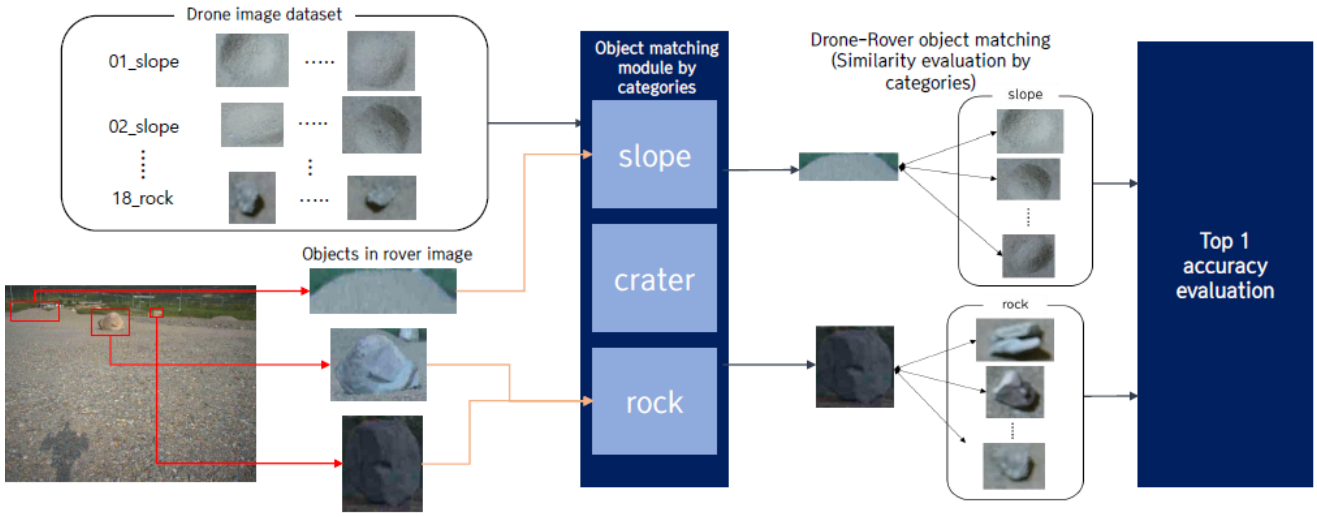


Fig. 18. Conceptual diagram of categorical detection accuracy measurement process

객체 인식으로부터 이미지 내 객체 추출 객체의 위치정보뿐만 아니라 범주 정보까지 산출해내게 된다. 이때 산출된 범주 정보를 활용하여 정확도를 측정하는 것이 바로 범주간 정확도 측정 프로세스인 것이다. 추출된 로버 객체는 범주 정보를 통해 해당 범주의 드론 객체의 임베딩 벡터와 정확도 측정이 이루어지게 된다.

5.2 추론 데이터셋 구성

드론-로버간의 매칭 추론을 위해 드론 객체 데이터를 기반으로 샘플 데이터셋(DB)을 구축하였다. 각 클래스당 드론 객체 이미지 50장을 선별하였으며, 50 m, 30 m, 고도에서 촬영된 드론 이미지 각각에 대해 독립적으로 구축하였다. 객체 매칭 추론 및 정확도 측정을 위한 로버 객체 데이터는 기 구축한 데이터 전체를 활용하였으며, 주간과 야간 객체가 종합된 데이터셋(148,566장의 객체)과 야간 객체를 제외한 데이터셋(118,856장의 객체) 두 가지 형태로 구성하였다.

5.3 추론 결과

5.3.1 클래스간 비교 결과

클래스간 단순 비교한 결과, Top-1(유사도 상위 첫 번째) 정확도는 두 로버 데이터셋에 대해 모두 균등확률(Uniform Probability, 1/N)과 유사한 수준(7%)으로 나타났다.

유사도 상위 3번째 내 정확도(Top 3)로 범위를 넓혔을 때, 정확도가 20%대까지 향상되었다(Table 8). 주간 로버 객체와 주/야간 통합 로버 객체의 정확도는 근소한 차이로 유사한 수준으로 평가되었다. 50 m 드론 영상에 비해 30 m 드론 영상의 정확도가 높은 경향을 보이며, 객체의 특징 정보가 더 많이 포함되므로 이러한 결과가 도출되었다고 판단된다.

Table 8. Drone image object inference accuracy rate result between class

	50 m drone image		30 m drone image	
	Rover (Day + Night)	Rover (Day)	Rover (Day + Night)	Rover (Day)
Top 1	7.96 %	7.43 %	14.1 %	13.3 %
Top 3	23.23 %	22.71 %	35.45 %	35.56 %

5.3.2 범주간 비교 결과

종합 데이터셋(주간+야간)과 주간 데이터셋에 대해 범주별 정확도를 측정한 결과, 약 16% 수준으로 나타났다. 주간과 야간 데이터 모두에 대해 추론한 결과가 근소하게 측정되었다(Table 9).

Table 9. Drone image object inference accuracy rate result between category

	50 m drone image		30 m drone image	
	Rover (Day + Night)	Rover (Day)	Rover (Day + Night)	Rover (Day)
Top 1	16.86 %	16.42 %	21.88%	21.05%

30 m, 50 m 고도의 범주간 비교 실험 결과는 Fig. 19와 Fig. 20과 같으며 Top 1 정확도 측정만 진행하였다. crater 객체의 경우, 범주 내에 속해 있는 목표 객체가 3종류뿐이기 때문에 Top 3 정확도 산출시 실험결과가 왜곡될 수 있기 때문이다. 세부적으로 목표 객체별 정확도를 보면, 범주에 포함된 객체가 적은 slope(4개), crater(3개)가 rock(11개)보다 상대적으로 높은 정확성을 보였다. 각 데이터셋별로 비교하였을 때, 종합 데이터셋에 대한 정확도가 주간 데이터셋보다 대체로 비슷한 경향을 보였다.

30 m 드론 객체에 대한 클래스간 단순비교결과는 50 m에 비해 높은 정확도를 보였다. Top 1 정확도의 경우, 두 데이터셋에 대해 13~14%대의 정확도를 보여 50 m에 비해 6~7%가량 정확도가 향상되었다. 또한, Top 3 정확도는 10% 이상 개선된 것을 확인하였다. 범주간 정확도 또한 50 m 대비 5% 가량 개선된 약 21% 수준으로 나타났고 목표 객체별 정확도는 50 m 결과와 유사하게 slope, crater의 정확도가 상대적으로 높게 측정되었다.

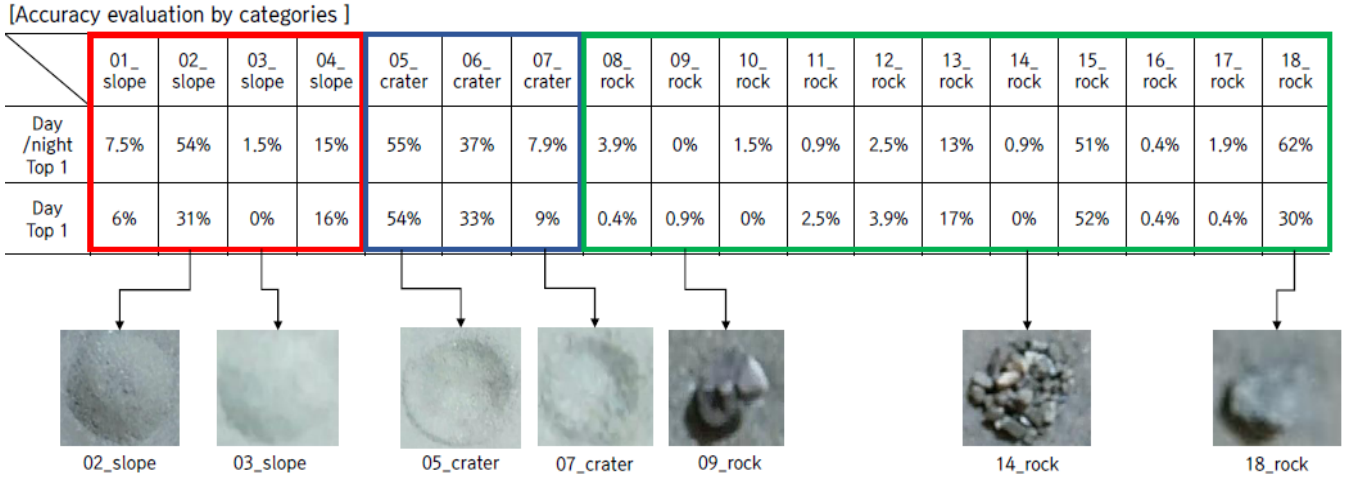


Fig. 19. Object inference result (50 m drone image)

[Accuracy evaluation by categories]

	01_ slope	02_ slope	03_ slope	04_ slope	05_ crater	06_ crater	07_ crater	08_ rock	09_ rock	10_ rock	11_ rock	12_ rock	13_ rock	14_ rock	15_ rock	16_ rock	17_ rock	18_ rock
Day /night Top 1	0.9%	81%	5.4%	7.5%	35%	46%	50%	0%	0.9%	0.4%	0.4%	15%	54%	7.5%	62%	15%	1.9%	11%
Day Top 1	1.9%	68%	3.9%	7.5%	36%	33%	56%	0%	0.4%	0%	1.4%	20%	41%	7.9%	62%	10%	0.9%	20%

Fig. 20. Object inference result (30 m drone image)

6. 소 결

객체 매칭 실험 결과 주간 데이터셋과 주/야간 데이터셋의 클래스간 정확도는 대부분 유사하게 측정되었다. 일부 실험 결과에서 주간 데이터셋에 비해 주/야간 데이터셋의 정확도가 높은 결과를 보였는데, 주간 데이터셋에 반해 주/야간 데이터셋은 데이터 수가 많고 다양한 영상 정보를 포함하고 있어 이러한 결과를 나온 것으로 예상된다. 또한, 드론의 고도에 따라 정확도에 큰 영향을 미친다는 것을 확인하였다. 50 m 영상에 비해 30 m 영상이 검출해야하는 객체의 윤곽 정보, 선명도와 같은 특징 정보를 더 많이 포함되어 있어 매칭 정확도가 상대적으로 높게 형성되었다고 판단된다.

추가적으로 본 실험을 통해 객체 매칭 기술의 보완점을 확인 할 수 있었다. 객체 매칭 기술은 두 영상 간의 유사도를 기반으로 정확도를 산출한다. 그러나 드론 영상과 로버 영상은 측정하는 객체의 방향이 다르므로 동일한 객체여도 정보 중첩이 상이하여 정확도가 낮게 측정된 것으로 판단된다. 따라서 드론 영상 취득 시 카메라를 수직 방향이 아닌 대각으로 설치하여 객체의 정보가 최대한 중첩되도록 영상을 취득한다면 객체 매칭의 정확도를 향상시킬 수 있을 것이라 사료된다.

7. 결 론

본 연구에서는 딥러닝 기반 객체 인식 및 매칭 소프트웨어를 개발하고 실제 로버에 적용하기 위해 소프트웨어 통합 및 최적화를 수행하였다. 구체적으로는 목표 객체의 영역을 픽셀단위로 계산하여 정밀하지만 연산 속도가 느린 기존의 영역 분할 기법 대신 목표 객체를 경계 상자 단위로 연산하는 객체 인식 기술로 객체 인식 알고리즘을 교체하였다. 객체 인식 알고리즘은 속도와 성능, 모두 범용적으로 뛰어나다고 평가 받은 YOLOv3를 채용하였고, YOLOv3의 성능을 개선하기 위해 경계 상자의 손실함수에 Gaussian modeling을 접목시킨 Gaussian YOLOv3를 최종적으로 구현하여 적용하였다.

실제 달 현장에서는 위성이미지로 구성된 데이터만 존재하며 로버 영상과 매칭되는 데이터 확보가 어렵기 때문에 위성 내 객체의 정보만 활용하여 로버 영상 내 객체 매칭이 이루어져야한다. 이를 위해, 기 구축한 로버영상과 위성 영상간 객체 매칭을 별도로 학습

시켜 매칭하는 실험을 진행하였다. 구체적으로는 추가로 확보한 위성(드론) 영상만을 객체 매칭 학습에 사용하고, 로버 영상을 추론에 사용하였다. 결과적으로, 50 m, 30 m 영상 중 인식 및 매칭 성능은 30 m에서 상대적으로 좋은 결과를 확인하였다. 그 원인으로 원시 데이터의 수와 근거리 이미지의 영향이 결과에 영향을 미쳤다고 볼 수 있다. 본 실험에서는 객체 매칭 성능의 한계점을 확인할 수 있었다. 객체 매칭은 영상 간의 유사도를 통해 정확도를 측정하지만 위성 측면 영상과 로버 측면의 영상은 대상 객체의 이미지 정보 중첩이 낮기 때문에 이러한 결과를 도출됐다고 판단된다. 따라서 위성(드론) 영상 취득 시 카메라를 사선으로 설치하여 대상 객체의 대각 정보가 최대한 취득되도록 영상을 수집하여 데이터를 보완한다면 객체 매칭의 정확도를 획기적으로 향상시킬 수 있을 것이라 사료된다.

본 연구의 결과는 이동체의 연속촬영 영상 기반 3차원 공간정보 구현 및 관심 공간 내 객체 위치 설정에 활용 가능할 것이다. 특히, 위성에서 촬영된 객체의 모습이 다양하게 반영될수록 로버와의 정보 연계를 통해 더욱 정확한 위치 설정이 가능할 것으로 보인다. 이를 토대로, 향후 달기지 건설 현장의 영상기반 시공 모니터링/시공제어를 위한 자동 현장 및 주요대상물 공간정보 구축 시스템 연계에 기여 할 수 있을 것으로 판단된다.

REFERENCES

Bolya, D., Zhou, C., Xiao F., and Lee, Y.J., 2020, YOLACT++ Better Real-Time Instance Segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(2), 1108-1121.

Choi, J., Chun, D., Kim, H., and Lee, H.J., 2019, Gaussian yolov3: An accurate and fast object detector using localization uncertainty for autonomous driving, Proceedings of the IEEE International Conference on Computer Vision, 502-511.

Pacha, A., Jan, H.J., and Jorge, C.Z., 2018, A Baseline for General Music Object Detection with Deep Learning, Applied Sciences, 8(9), 1488.

Redmon, J. and Farhadi, A., 2017, YOLO9000: better, faster, stronger, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 7263-7271.

Redmon, J. and Farhadi, A., 2018, Yolov3: An incremental improvement, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1804, 1-6.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A., 2016, You only look once: Unified, real-time object detection, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 779-788.

Ren, S., He, K., Girshick, R., and Sun, J., 2016, Faster r-cnn: Towards real-time object detection with region proposal networks, IEEE Transactions on Pattern Analysis and Machine Intelligence, 39, 1137-1149.

Schroff, F., Dmitry, K., and Philbin, J., 2015, Facenet: A unified embedding for face recognition and clustering, Proceedings of the IEEE Conference On Computer Vision And Pattern Recognition, 815-823.

Tan, M. and Quoc, V.L., 2019, Efficientnet: Rethinking model scaling for convolutional neural networks, International Conference on Machine Learning, 6105-6114.

Wu, X., Sahoo, D., and Steven Hoi, C.H., 2020, Recent advances in deep learning for object detection, Neurocomputing, 396, 39-64.