Journal of the KIMST, Vol. 25, No. 3, pp. 300-310, 2022
ISSN 1598-9127(Print) · 2636-0640(Online)
DOI https://doi.org/10.9766/KIMST.2022.25.3.300

Research Paper 정보·통신 부문

유사물체 치환증강을 통한 기동장비 물체 인식 성능 향상

허지성[1] · 박지훈[*,1]

[1] 국방과학연구소 국방인공지능기술센터

Object Detection Accuracy Improvements of Mobility Equipments through Substitution Augmentation of Similar Objects

Jiseong Heo[1] · Jihun Park[*,1]

[1] *Defense AI Technology Center, Agency for Defense Development, Korea*

(Received 16 December 2021 / Revised 7 March 2022 / Accepted 29 April 2022)

Abstract

A vast amount of labeled data is required for deep neural network training. A typical strategy to improve the performance of a neural network given a training data set is to use data augmentation technique. The goal of this work is to offer a novel image augmentation method for improving object detection accuracy. An object in an image is removed, and a similar object from the training data set is placed in its area. An in-painting algorithm fills the space that is eliminated but not filled by a similar object. Our technique shows at most 2.32 percent improvements on mAP in our testing on a military vehicle dataset using the YOLOv4 object detector.

Key Words : Object Detection(물체 인식), Image Augmentation(영상 증강), Military Vehicle(군용 기동장비), Artificial Intelligence(인공지능)

기 호 설 명

mAP : mean Average Precision
IoU : Intersection over Union

1. 서 론

콘볼루션 인공신경망 기반 컴퓨터 비전 기술의 발전으로 인해 물체 인식 기술의 정확도는 매우 높아졌다. 주어진 영상 내에서 물체를 인식하는 활동은 국방 분야의 다양한 운용 환경에서의 경계 감시 활동과 매우 유사하며, 따라서 물체 인식 기술의 발전이 경계 감시 체계에 인간을 보조할 수 있는 방안으로 활용될 가능성이 매우 크다. 현재 운용되고 있는 지상 기동무기체계의 경우 외부 상황을 육안으로 식별하거나, EO/IR등의 센서를 활용해 기동장비 내부에서 외부의

* Corresponding author, E-mail: jhpark_a@add.re.kr

환경과 위협요소 등을 식별하고 있고, 인공지능 기반 물체 인식 기술은 이러한 식별 과정에서 인간을 보조하는 수단으로 활용될 수 있다. 또한 최근 드론 기술의 발전으로 경계감시를 비롯하여 다양한 분야에서 드론의 활용성이 대두되고 있으며, 드론의 자율 운용을 위해 인식 자동화 기술이 활용될 가능성이 높다.

최신의 인공신경망 기반 인공지능 모델을 학습하기 위해서는 대량의 데이터가 필요하다. 많게는 수백만 장의 레이블 된 영상 데이터를 통해 인공지능 모델을 학습하게 되는데, 이러한 학습용 빅데이터를 수집하고 레이블 하는 노력과 비용이 굉장히 크다. 의료나 국방과 같은 일부 분야는 데이터 획득 및 레이블이 더욱 어려울 수 있는데, 데이터 레이블에 전문적이 지식이 필요하여 레이블에 소요되는 비용이 매우 크거나, 때로는 학습에 필요한 데이터 수가 매우 적은 경우도 있다. 학습용 데이터가 적을 경우에 인공지능 모델을 개발하고 탑재하여 운용하는데 큰 어려움이 있을 수 있다. 학습용 데이터가 적다면 실제 운용 환경에서의 추론용 입력 데이터가 학습용 데이터의 분포를 크게 벗어날 수 있으며, 이는 해당 입력 데이터에 대한 잘못된 인식 결과로 이어질 수 있다.

학습용 데이터의 분포를 최대한 크게 수집하여 실사용 시 가능한 입력 데이터의 범위를 모두 포함하는 것이 가장 좋은 상황이라고 할 수 있으나, 많은 경우에 실제 사용 환경을 모두 포함하는 학습 데이터를 만드는 것은 불가능에 가깝고, 따라서 일부 사용 환경에서는 인공지능 인식 기술의 정확도가 보장되기 어려운 게 현실이다. 인공지능 인식 기술은 다양한 접근 방법을 통해 학습 데이터에서의 정보를 바탕으로 일반화시켜 학습 데이터의 분포 외에 있는 추론 데이터에 대해서도 올바르게 인식하는 것을 목표로 발전하고 있다.

인공지능 인식 기술의 정확도를 높이기 위해 많은 접근방법이 있고, 크게 추론 시의 속도에 영향이 없는 방법(bag of freebies[1]) 과 추론 시의 속도에 영향이 있는 방법(bag of specials[1]) 으로 나누어 볼 수 있다. 데이터 증강, 학습 데이터 선택 알고리즘 개선, 손실함수의 개선 등이 추론 시의 속도에 영향이 없는 방법으로 분류될 수 있는데, 이러한 방법들은 인공지능 모델의 학습 시에는 학습에 소요되는 시간이나 컴퓨팅 파워가 더 소요될 수 있지만, 추론 시의 속도에는 영향을 주지 않는다. 일반적으로 인공지능 모델이 탑재되기 전에 모델 학습은 실제 사용 환경이 아닌 별도의 학습 환경에서 학습되기 때문에 시간이나 컴퓨팅 파워의 제약조건에서 비교적 자유로운 편이다.

본 연구에서는 추론 시의 영향이 없는 데이터 증강 방법의 하나로, 새로운 데이터 증강 방법을 제시하여 지상 기동장비에 대한 차량과 드론에서의 인식 정확도를 높이는 방안에 대해 소개한다. MSCOCO[2], Pascal VOC[3]와 같은 일반적인 공개된 물체 인식알고리즘에서 다루는 문제는 사람, 개, 책상, 모니터 등 서로 완전히 다른 형상을 갖는 물체가 섞여있는 데이터셋에서의 인식을 다루는 문제인데, 본 연구에서 다루는 데이터셋은 다양한 기동장비(전차, 자주포, 장갑차 등 총 8종)의 형상이 일반인의 시각에서는 구분하기 어려울 정도로 비슷하다는 특징이 있다.

본 연구는 모양이 유사한 물체를 바꾸어 증강할 경우 원본 영상에 가까운 증강 영상이 만들어질 수 있다는 점에 착안하여 증강 영상을 생성한다. 임의의 물체 영역을 잘라 새로운 영상에 붙여 넣을 경우, 물체의 크기, 질감 등이 원본 물체와 달라 이질감이 있고, 이러한 이질감 때문에 물체의 다양한 배경에서의 식별을 학습하는 것이 아니라, 물체와 배경영역사이의 이질감에 대해 학습이 될 수도 있다. 따라서 최대한 원본처럼 보이는 증강영상이 필요하고, 서로 다른 두 물체영역의 마스크가 유사한 경우 물체의 형상, 자세 등이 유사할 것으로 가정할 수 있고, 이러한 물체들을 증강 대상으로 삼았다.

본 연구에서 제안하는 방법은 증강 대상이 되는 영상에 대해 물체를 특정하고, 해당 물체와 유사한 물체를 학습 데이터에서 선별한 뒤, 유사한 물체로 본래의 데이터를 치환하는 증강과정으로 구성되어 있다. 본 연구의 특징으로는 행렬 연산을 통해 물체 간의 유사도 계산을 가속화했다는 점이 있고, 유사물체로 치환한 뒤 본래의 물체 영역 중 새로운 물체로 가려지지 않은 영역에 대해 주변 영역의 패턴을 기반으로 칠하는(in-paint) 알고리즘이 적용되었다는 점이다.

본 연구는 YOLOv4 물체 인식기를 기반으로 8개 클래스의 기동장비 데이터셋에 대해 실험을 수행하였다. 데이터를 학습과 검증 데이터셋으로 나누어 학습데이터만을 활용하여 학습한 모델과, 학습데이터에 치환증강이 적용되어 학습한 모델들의 검증 데이터에서의 정확도를 비교하였고, 최대 2.32 %의 정확도 향상을 확인할 수 있었다.

2. 관련 연구

2.1 데이터 증강방법에 관한 연구

영상 분류, 객체 분할, 물체 인식 등의 컴퓨터 비전 연구에서 인공신경망의 정확도를 향상시키기 위하여 다양한 데이터 증강방법이 연구되었다. 데이터 증강 방법은 기존의 학습 데이터로부터 변형된 새로운 데이터를 증강하여 적은 데이터로도 인공신경망의 일반화 능력을 향상시키는 데에 목적이 있다. 밝기, 색조, 대비 등을 변화시키는 기본적인 이미지 증강에서부터 시작하여 최근에는 강화학습이나 적대적 생성 신경망(Generative Adversarial Networks, GAN) 등을 활용한 데이터 증강 방법이 활발하게 연구되고 있다.

Cutout[4] 방법은 영상의 일부 영역을 삭제 (R, G, B, 값을 0으로 변경) 함으로써 세부 특징 중 일부 영역에만 편중되어 학습되지 않고, 일부 영역의 정보가 없는 경우에도 다른 영역을 통해 인식이 가능하도록 유도하는 방법이다. Cutout[4]은 영상 내 물체를 일부 가리는 증강방법을 사용하는데, 이 중 가운데 가림(center occlusion), 경계 가림(boundary occlusion) 증강 방법이 정확도 향상에 더 높은 효과를 보였다.

Mixup[5]는 두 개의 이미지를 픽셀별로 가중합하여 데이터를 증강시키는 방법이다. 이때 가중치는 베타 분포(beta distribution)로부터 임의로 추출되는 값을 사용한다. 레이블은 두 이미지의 레이블로부터 선형보간된 one-hot vector를 사용하며, 이를 통해 특정 클래스가 아닌 클래스가 합쳐진 확률에 대한 학습을 통해 인공신경망의 일반화 능력을 강화하고자 하였다. Mixup[5]은 CIFAR-10, CIFAR-100[6]과 같은 영상 분류에 대한 정확도를 높였다. 하지만, Mixup[5]은 자연스럽지 않은 증강 영상을 만들어 내기 때문에 지역화(localization)에서 좋지 않은 결과를 보인 것으로 나타났다[7].

CutMix[7]는 원본 이미지에 임의로 선택된 다른 이미지의 일부 직사각형 영역을 붙여 넣는 방식의 데이터 증강 방법이다. CutMix[7]로 생성된 데이터는 붙여넣어진 영역이 차지하는 비율을 고려하여 두 개 클래스에 대한 확률적 레이블을 정의하고, 이에 대한 학습을 수행한다. CutMix[7]방식은 Cutout[4]와 달리 손실되는 픽셀이 없기 때문에 효율적인 학습이 가능하며, 정확도 향상 측면에서도 다른 증강방법들에 비해 우위를 보이는 것으로 나타났다. 하지만, 붙여 넣어지는 직사각형 영역의 내부에 물체와 관련없는 픽셀을 포함하기 때문에 영상 분할 등에 적용하기에 비효율적인 부분이 일부 존재한다.

Lin 등[8] 이 발표한 Patch AutoAugment 방법은 강화학습을 활용하여 영상을 격자무늬(Patch) 형태로 나눈 것 중 일부를 증강하는 방법을 제안하였다. 다개체 강화학습(Multi Agent Reinforcement Learning, MARL)을 활용하여 패치 영역의 일부를 변경하는 방법을 강화학습의 행동(action)으로 정의하여 목표 신경망(target network)의 학습 손실(training loss)이 최소화 되는 방법을 학습하였다.

Ghiasi 등[9]의 연구에서는 객체 분할(instance segmentation) 문제에서 객체 영역을 복사하여 다른 배경에 붙여넣는 간단한 복사-붙여넣기 방법이 효과적임을 보였다. 특히, 해당 연구에서 제시한 방법은 대규모 지터링(large-scale jittering) 방법으로, 영상의 객체 영역이나 영상크기 자체를 소규모(0.8배 ~ 1.25배)로 증강하는 것 보다 큰 규모(0.1배 ~ 2.0배)로 변경하여 붙여넣기 하는 것이 정확도 증가에 효과가 더 크다는 것을 보였다.

Google Brain에서 Cubuk 등이 발표한 AutoAugment[10]는 강화학습을 활용하여 영상에 적용될 증강 방법과 확률, 증강의 크기를 도출하도록 하였다. 영상증강 방법과 해당 방법에 해당하는 크기를 찾는 최적화 문제로 생각하여 탐색 공간(search space)에서의 증강을 찾고, 해당 증강을 적용할 확률 및 적용 순서까지 도출하는 것을 목적으로 하였다. 같은 팀에서 발표한 RandAugment[11]의 경우 임의로 증강방법을 선택하여 적용하는데, 데이터셋의 크기가 크고 인공신경망의 크기가 클수록 증강을 더 크게 적용하는 것이 정확도 향상에 좋다는 결과를 보였다.

Hao 등[27]이 발표한 선택적 개체 변환(selective instance-switching) 방법은 같은 클래스에 대한 두 물체를 크기, 모양을 기준으로 유사한 물체를 찾아서 두 물체의 위치를 바꾸어 물체 인식 데이터셋을 증강하는 방법에 대해 발표했다. 본 연구는 서로 다른 클래스에 대해 IoU를 기준으로 유사 물체를 찾고, IoU계산 연산을 가속화 하였으며, 잔여 영역에 대한 칠하기 알고리즘을 적용한 것에 차이점이 있다.

2.2 물체 인식 인공신경망에 관한 연구

심층 인공신경망의 발전에 힘입어 물체 인식을 위한 다양한 인공신경망 모델이 개발되었다. 물체 인식은 영상 내 물체의 위치와 크기 및 클래스 정보를 예

측하는 작업이다.

Faster R-CNN은 물체가 있을만한 영역을 Region Proposal Network(RPN)라는 인공신경망으로 연산함으로써 기존의 selective search을 통한 방법들에 비해 현저하게 빠른 성능과 정확도를 확보할 수 있었다[12].

Tan 외 2인의 연구[13]는 심층 컨볼루션 인공신경망의 깊이, 너비, 해상도를 모두 고려하여 신경망의 크기를 증가시키는 compound scaling을 사용하였으며, 기존의 feature pyramid network[14]를 개선한 BiFPN과 weighted feature fusion 방법을 통해 서로 다른 해상도의 feature map들이 상호보완적으로 작용되어 높은 정확도를 달성할 수 있도록 하였다.

YOLOv4[1]는 다양한 영상 증강 방법(mosaic)과 인공신경망 모듈(Squeeze-Excitation block[15], ASPP block[16], BiFPN[13] 등) 및 학습 최적화 방법(batch normalization[17], cosine annealing[18], dropout[19] 등) 들을 적극적으로 활용하여 높은 정확도와 더불어 빠른 추론속도를 달성하였다.

3. 유사물체 치환 증강 알고리즘

본 장에서는 유사물체 치환증강 방법에 대해 설명한다. 유사물체 치환증강은 유사물체 선별 과정과 치환증강 과정으로 나눌 수 있다.

3.1 유사물체 선별 방법

데이터셋이 의미적 분할(semantic segmentation) 방법으로 레이블 되었다고 가정했을 때, 각 개체의 경계 상자는 이미지 상 X, Y 좌표계의 최솟값과 최댓값을 취하여 얻을 수 있다. 경계 상자로 잘라내기(Crop) 된 상자 영역에 대해 물체에 해당하는 영역을 1, 물체에 해당하지 않는 영역을 0으로 하는 이진 마스크(binary mask)를 계산할 수 있다. 본 연구에서는 서로 다른 두 물체의 유사도를 계산하기 위해 이진 마스크를 활용하여 계산한다.

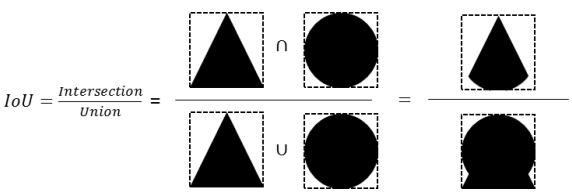

Fig. 1. IoU calculation

서로 다른 두 개의 사각형 모양 이진 마스크의 유사도를 계산하기 위해 Intersection over Union(IoU) 지표를 활용한다. IoU 지표는 두 영역의 합집합 영역(union) 영역대비 교집합 영역(Intersection)을 계산하여 도출할 수 있다. Fig. 1은 IoU의 계산법이다.

본 연구에서는 물체 치환 증강을 위해 학습 데이터 내의 모든 물체들에 대해 IoU를 미리 계산해놓고, 계산값을 바탕으로 치환할 물체를 선별하는 방법을 사용했다. 학습데이터에 총 N 개의 물체가 있다면 해당 물체 사이의 모든 유사도 값을 계산하는 것은 $O(N^2)$ 복잡도의 계산이 필요하다. 학습데이터의 수가 늘어날수록 제곱에 비례하여 계산량이 증가하기 때문에 계산과정의 최적화 없이는 오랜 시간이 걸릴 수 있다.

이러한 IoU 계산과정을 가속화하기 위해 본 연구에서는 행렬기반 GPU 계산을 활용하였다. 우선 데이터셋 내의 모든 영상에 대해 같은 크기(가로크기 w, 세로길이 h)로 리사이즈하여 크기를 균일화 한 뒤, 해당 마스크를 평탄화(flatten)하여 각 물체 영역을 1차원의 벡터 형태로 변경하였다. 물체의 차원(dimension)은 1 * d의 차원을 갖게 되는데, 이 때 d는 리사이즈한 영상의 가로와 세로의 곱과 같다(w * h). Fig. 2는 본 과

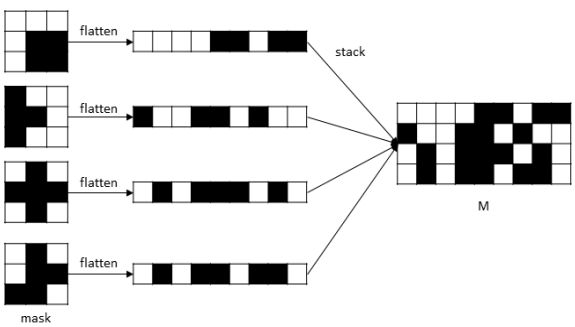

Fig. 2. Matrix M calculation by resize, flatten, and stack

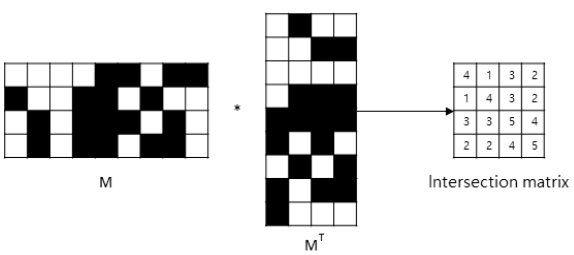

Fig. 3. Intersection matrix calculation

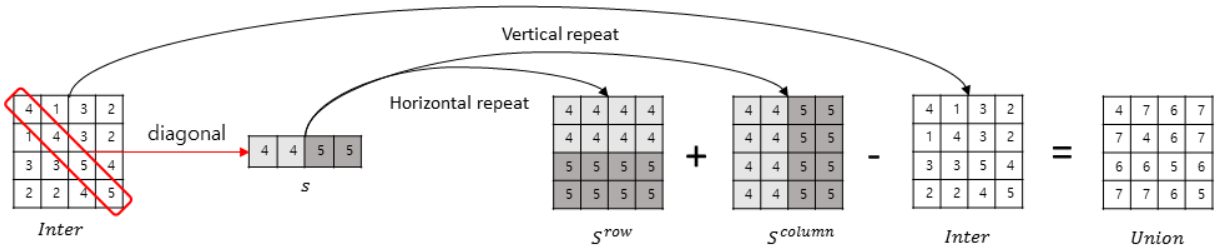

Fig. 4. A process of union matrix calculation based on an intersection matrix

정을 나타내는 그림으로, 3×3 크기의 물체 영역이 있다고 가정하였을 때 4개의 물체 영역에 대해 평탄화한 뒤 세로 방향으로 쌓아서(stack) 2차원 행렬 M을 만드는 과정을 나타낸다. 행렬 M의 세로 길이는 학습 데이터 내의 물체의 개수 N을 갖고, 가로 길이는 각 이미지의 차원수인 d를 갖게 된다.

행렬 M에서 각 행은 하나의 이미지에 해당하는 벡터이다. M과 M의 전치행렬(transpose matrix) M^T의 행렬(matrix multiplication) 곱을 교집합 행렬(intersection matrix, $Inter$)로 정의할 수 있다. 행렬 곱을 수행하면 두 영상의 동일한 위치의 값이 모두 1인 경우에 대해서만 결과 값이 1을 갖고, 모든 위치에 대한 값들을 더한 값이 교집합 행렬의 값이 된다. 따라서 교집합 행렬의 (i, j)의 원소 값은 M에서 i번째 마스크와 j 번째 마스크의 교집합 영역의 크기를 나타내게 된다. Fig. 3은 교집합 행렬을 구하는 과정의 예시를 보여준다.

Fig. 4는 교집합 행렬를 통해 합집합 행렬(union matrix, $Union$)를 구하는 과정을 보여준다. 합집합 행렬 (i, j)의 원소 값은 각 i번째와 j번째 이미지의 마스크 영역의 크기의 합에서 교집합의 크기를 뺀 것과 같다($A \cup B = A + B - (A \cap B)$). 교집합 행렬($Inter$)의 대각선 성분은 각 마스크에서 1에 해당하는 성분의 개수를 나타내므로 해당 마스크의 크기라고 볼 수 있다. 대각선 성분에 대해 크기 벡터(Size vector, s)로 정의한다. 합집합 행렬($Union$)의 성분은 $Inter$와 s를 활용하여 다음과 같이 정의될 수 있다.

$$Union_{i,j} = s_i + s_j - Inter_{i,j} \tag{1}$$

본 계산을 쉽게 하기 위해 크기 벡터를 크기 행렬로 Fig. 4와 같이 S^{row}와 S^{column} 행렬로 만들어 행렬 연산으로 구할 수 있다. S^{row}와 S^{column}는 아래와 같이 정의된다.

$$S^{row}_{i,j} = s_i \qquad (i,j \in 1...N) \tag{2}$$

$$S^{column}_{i,j} = s_j \qquad (i,j \in 1...N) \tag{3}$$

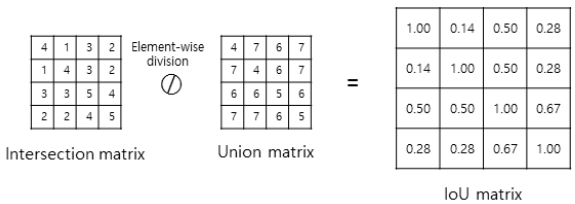

Fig. 5. IoU matrix calculation

마지막으로, 계산된 교집합 행렬과 합집합 행렬의 원소별 나눗셈을 통해 IoU 행렬을 도출할 수 있다. Fig. 5는 이 과정을 도식화한 것이다.

Table 1은 IoU 행렬을 구하는 데 걸리는 시간을 보여준다. GPU를 활용한 행렬 연산의 경우 CPU를 활용한 이중 for문 방법에 비해 수십만 배 이상 빠른 결과를 보였다.

Table 1. Speed comparison for IoU matrix calculation

행렬 기반 계산(GPU)	이중 for문(CPU)
0.00213 초	506.11142 초

마지막으로, IoU 행렬 원소의 값이 특정 수치(IoU 임계치) 이상인 원소들을 필터하여 해당 물체 쌍이 서로 유사하다고 정량적으로 판단할 수 있다.

3.2 치환증강 알고리즘

본 논문에서는 물체 간 유사도를 기반으로 물체 영역을 치환하여 데이터를 증강하는 알고리즘을 제시한다. Fig. 6은 전체적인 알고리즘을 예시 사진을 통해

Fig. 6. Substitution augmentation for Object detection dataset

도식화 한 것이다(보안상의 이유로 실제 데이터셋이 아닌 공개 사진[22,23]를 사용한 예시). 먼저, 원본 영상(①) 으로부터 물체 영역을 삭제하고 (②) 해당 물체 영역과 유사도(IoU)가 특정 임계치 이상인 물체 중 하나를 증강물체로 임의로 선정한다 (③). 선정된 증강 물체의 물체 영역을 분할하고, (④) 원본 영상의 물체 영역 크기와 동일한 크기로 리사이즈하여 붙여 넣는다. (⑤) 이 경우 각 물체의 영역 모양이 다르기 때문에 원본 영상에서 삭제된 영역의 일부가 증강 물체에 의해 가려지지 않을 수 있다. 이 때 해당 부분을 자연스럽게 채우기 위해 Fast marching[20] 또는 Navier-Stokes[21]와 같은 칠하기(in-painting) 기법을 활용한다. (⑥) In-painting 기법은 주변의 픽셀을 참고하여 해당 영역을 채우는 방법으로, 다양한 연구가 진행 중이다. 본 연구에서는 구현의 용이성과 속도를 위해 Navier-Stokes[4]를 활용하였다. 치환 대상 물체와 치환물체의 선정 시, 과도한 크기 확대 및 축소에 의한 효과를 줄이기 위해 일정 크기(가로 크기 224픽셀) 이상의 물체만을 대상으로 하였다.

4. 실험 결과

4.1 실험 환경

본 연구에 사용된 데이터셋은 기동장비를 포함한 총 8종의 클래스에 대한 데이터셋으로, 실제 운용환경과 유사한 시험환경에서 획득되었다. 보안상의 이유로 8종의 정확한 명칭 대신 A, B, C, D, E, F, G, H 라는 이름으로 표기할 예정이다. 대상 장비는 모두 궤도형 기동장비로, 전차, 자주포, 장갑차의 범주에 속하는 기종들이다. 각 기동장비들은 최대한 다양한 환경에서 촬영하였는데, 이는 수풀 등을 이용한 부분적 가림 상황, 건물로 인한 부분적 가림 상황, 발연기 등의 연막을 이용한 가림 상황 등을 포함한다. 데이터는 총 111개의 동영상으로 이루어져있는데, 각자 다른 상황(가림 상황, 기동장비 위치 등)에서 촬영된 것이다. 동영상을 기준으로 2:1로 나누어져 전체 중 2/3개의 동영상은 학습 데이터로, 1/3개의 동영상은 검증 데이터로 사용되었다. 동영상에서 프레임단위로 학습/검증 데이터를 나눈 것이 아닌, 각각 다른 상황에서 촬영된 동영상 단위로 학습/검증 데이터를 나누었기 때문에 학습 데이터와 검증 데이터 사이의 유사도가 상당히 낮

Fig. 7. Dataset examples(left: a training data of class A, right: a validation data of class E). Images are intentionally blurred for security reasons

다고 할 수 있다. 총 이미지 수로 20,266 개의 학습 이미지와 8,779 개의 검증 이미지로 구성되었다. Fig. 7은 학습 및 검증에 사용된 데이터셋 예시를 나타낸다. 왼쪽은 학습 데이터 중 하나로, A 클래스에 해당하는 영상이며, 오른쪽은 검증 데이터 중 E 클래스에 해당하는 사진이다(영상은 보안상의 이유로 흐리게 처리되었음).

본 실험에서 사용된 물체 인식기는 YOLOv4[1]이다. YOLOv4는 최근 물체 인식기 중 높은 정확도를 보이고 있다. 실험에는 총 3개의 실험 환경을 활용하였는데, 각각 608, 512, 416 크기의 영상을 입력으로 받아 8개 클래스에 대해 물체의 클래스와 위치를 인식하는 알고리즘을 학습시켜 그 정확도를 비교하였다.

본 연구에서는 mean average precision(mAP) 정확도를 사용하여 비교하였다. mAP는 Average Precision(AP)의 평균으로, AP는 각 클래스에서 서로 다른 Intersection over Union(IoU) 임계치에 대한 정밀도(Precision)의 평균값을 나타낸다. 본 연구에서는 IoU 임계치는 0.5를 사용하였다. 즉, 각 클래스 별로 IoU가 0.5를 넘는 바운딩 박스에 대한 예측이 얼마나 맞고 얼마나 틀렸는지에 대해 정밀도를 측정 한 뒤에, 모든 클래스에 대해 평균을 낸 값으로 볼 수 있다.

4.2 치환 증강을 위한 IoU 임계치 설정

본 연구에서 제시된 치환 증강은 IoU를 기준으로 유사한 물체를 선별하고, 유사한 물체들에 대해서만 치환을 진행한다. IoU 기준에 따라 치환할 대상 물체가 없는 경우가 존재 할 수 있다. 예를 들어, 특정 객체와 IoU가 0.9 이상인 물체가 하나도 없는 경우에, IoU 임계치 0.9를 사용한 치환증강 알고리즘에서는 해당 물체는 치환되지 않는다. IoU 임계치가 너무 높게 설정되는 경우 유사도가 매우 높은 물체가 존재하는 경우에만 물체가 치환되기 때문에 증강 대상이 되는 영상의 수가 줄어들 수 있다. IoU 임계치가 너무 낮은 경우엔 유사하지 않은 물체도 치환이 되고, 원본 물체가 삭제된 뒤에 임의의 배경으로 칠하기(in-paint) 되는 영역이 커져 실제 영상과의 괴리가 생길 수 있다.

본 연구에서는 적절한 IoU 기준을 찾기 위해 IoU를 0.1 단위로 변화시켜 치환 증강이 적용된 데이터셋을 생성하였고, 해당 데이터셋들을 대상으로 학습된 모델의 정확도를 평가하였다. 상기 설명대로 치환증강 시 IoU 임계치가 낮을수록 더 많은 수의 증강 영상이 생성될 수 있는데, Table 2는 IoU 임계치에 따른 증강영상의 수를 나타낸다. 증강 영상은 IoU 임계치가 0.1일 때 최대 19822개까지 생성되고, 0.9일 경우엔 3955개만 생성된다. IoU임계치가 0.9일 때는 상당수의 영상이 치환 대상 물체가 없어 치환되지 못하는 상황으로, 유사한 물체로 치환하는 것이 증강에 유리할 것인지, 아니면 가능한 많은 증강을 수행하는 것이 좋을지 정확도를 통해 비교가 필요하다.

Table 2. The number of augmented images of different IoU thresholds

IoU 임계치	증강 영상 수
0.1	19822
0.2	19822
0.3	19822
0.4	19821
0.5	19797
0.6	19667
0.7	19111
0.8	17490
0.9	3955

각각의 임계치를 활용한 치환증강 데이터셋을 원래의 학습데이터셋과 합쳐 각 크기 별로 물체 인식기를 학습하였고, 그 정확도를 비교하였다. Fig. 8은 그 결과를 나타낸다.

영상 크기 별 결과를 보면 416 크기에서는 IoU 임계치 0.5에서, 512 크기에서는 IoU 임계치 0.6에서, 608 크기에서는 IoU 임계치 0.5에서 가장 높은 정확도를 갖는 것을 확인할 수 있다. IoU 임계치가 너무 낮거나 너무 높은 경우보다는 0.5~0.6 수준의 임계치가 증강 영상을 다양하게 많이 만들 수 있고, 유사한 물체를 선별하여 정확도를 최대로 높일 수 있는 IoU 임계치라는 것을 확인할 수 있었다. 또한, 아주 낮은 임계치(0.1 혹은 0.2) 보다 중간정도의 임계치를 활용한 증강이 정확도 향상에 더 효과가 크다는 것은, 임의 선정된 증강 물체 선택보다 유사도를 기반으로 한 증강 물체 선택이 치환 증강에 유리함을 나타낸다.

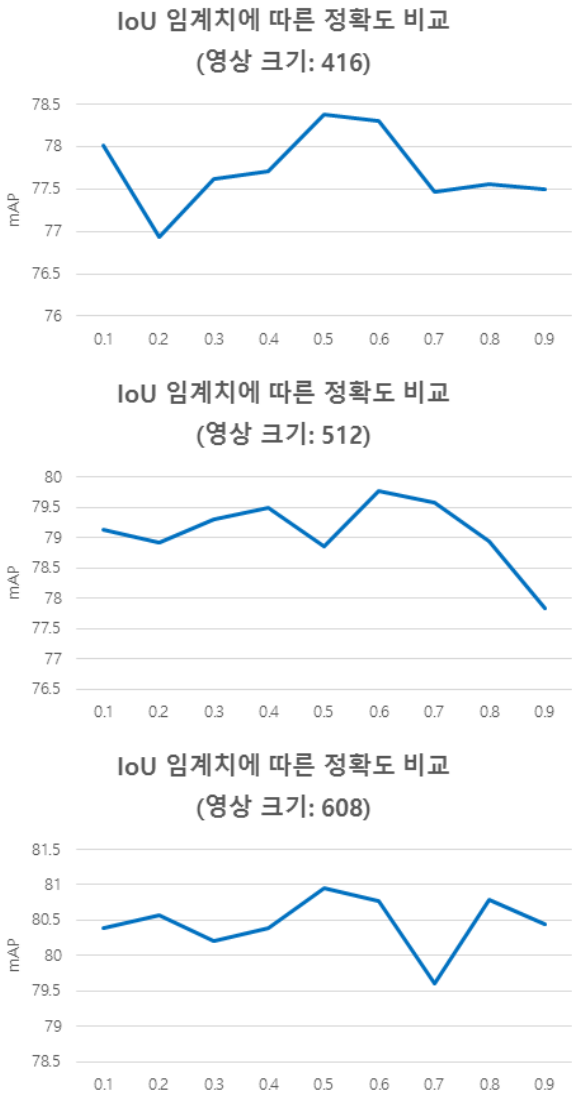

Fig. 8. Accuracy comparison varying IoU thresholds

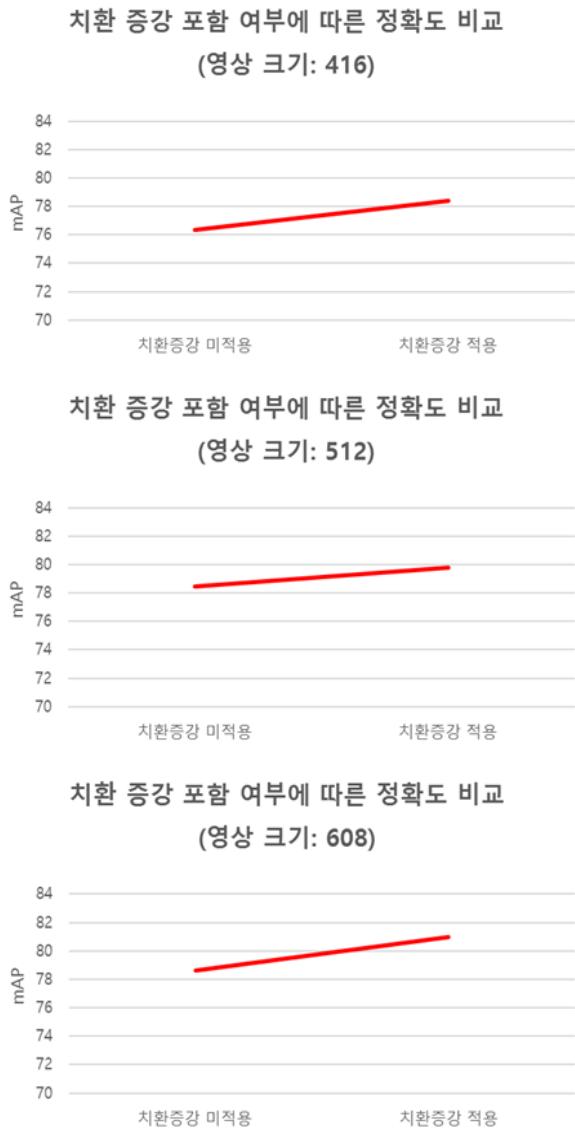

Fig. 9. Accuracy comparison according to whether or not substitution augmentation is included

4.3 치환 증강에 따른 정확도 향상 분석

치환 증강을 적용한 데이터셋에 대한 정확도를 확인하기 위해 동일한 실험 환경에서 치환증강을 적용한 상황과 적용하지 않은 상황에 대해서 정확도를 비교하였다. 기본적인 환경은 YOLOv4[1]의 실험환경과 동일하다. YOLOv4에서 물체 식별 정확도 향상을 위해 포함된 Mosaic 증강이 적용되었으며, 영상 크기 별로 모델을 학습하여 각 모델의 검증 데이터에서의 정확도를 측정하였다.

Fig. 9는 치환증강 포함 여부에 따른 정확도(mAP) 비교를 나타낸다. Fig. 9는 영상 크기 별로 측정된 결과이며, 각각 상단, 중간, 하단의 결과는 416, 512, 608 크기의 영상 크기를 사용한 결과이다. 각 그림의 좌측은 치환증강을 적용하지 않은 결과를 나타내며, 우측은 치환증강을 적용한 결과를 나타낸다. 전반적으로 우측의 결과값이 좌측보다 높은 것은 본 연구를 통해 제안된 치환증강이 모든 크기의 영상 크기에서 정확도가 향상되는 효과를 보였다는 것을 뜻한다.

Table 3은 치환증강 적용시의 정확도 향상을 수치로 나타낸 것이다. 각각 416, 512, 608의 이미지 크기에서 2.01 %, 1.34 %, 2.32 %의 향상을 이루어내 YOLOv4의 결과를 상당히 향상시켰다고 볼 수 있다.

Table 3. Accuracy improvements on substitution augmentation (unit: mAP, %)

영상 크기	치환증강 미적용	치환증강 적용	정확도 향상
416	76.37	78.38	+2.01
512	78.43	79.77	+1.34
608	78.64	80.96	+2.32

Table 4는 클래스 별 치환증강 적용 전후의 정확도를 비교한 것이다. 치환증강을 적용한 경우 416 영상 크기에서는 총 6개의 클래스에서 향상을 보였으며, 향상 폭은 최대 6.86 %로, 클래스 D에서 최대 향상폭을 보였다. 클래스 G와 H에서 일부 정확도가 소폭 감소하였으나, 최대 1.5 % 감소하였고, 결과적으로 mAP는 2.01 % 증가하였다. 512 크기의 영상에서는 4 개의 클래스에서 향상, 4 개의 클래스에서 정확도 감소가 있었다. 정확도는 클래스 F에서 최대 6.23 %가 증가하였으며, 클래스 G에서는 2.1%의 정확도 감소가 있었다. 전체적으로 증가폭이 감소폭보다 크기 때문에 mAP는 1.34 % 증가하였다. 608 영상크기에서도 5 개의 클래스에서는 정확도 증가, 3개의 클래스에서는 정확도가 감소하였다. 최대 7.12 %의 정확도 증가가 클래스 F에서 나타났으며, 클래스 A의 경우엔 정확도가 3.81 % 감소하였고, mAP 는 2.32 % 증가하였다.

클래스 별로 정확도 향상의 차이는 있지만, 전반적으로 증가폭이 감소폭보다 우월하여 mAP가 향상되는 것을 확인할 수 있었다. 클래스 별로 결과에 대해 검토해 보았을 때, 상호 유사도가 높은 클래스인 B, D, E, F 클래스의 경우 유사물체 치환증강에 의한 정확도 향상에 큰 도움이 되는 것을 확인할 수 있었다. 다만, 타 클래스 대비 유사도가 크지 않은 G, H 클래스의 경우 치환증강에 의한 정확도 향상을 기대하기 어려웠다.

치환 증강이 YOLOv4 외의 다른 인공신경망 구조에서도 물체 인식 정확도 향상의 효과가 있음을 보이기 위해, RetinaNet[24], YOLOv3[25], EfficientNet B0[26]의 인공신경망 구조에 치환증강이 적용된 데이터와 적용되지 않은 데이터를 이용해 학습한 모델의 정확도를 비교해보았다. Table 5에 나타난 결과에 따르면, 치환증강이 적용된 경우 세 가지 인공신경망 구조 모두 정확도가 향상되었으며, 향상폭은 RetinaNet에서 1.60 %, YOLOv3에서 0.63 %, EfficientNet B0에서 1.87 % 이다. 이러한 결과는 치환증강의 물체 인식 정확도 향상이 인공신경망에 종속적이지 않음을 나타낸다. 치환 증강은 다양한 종류의 인공신경망과 다양한 크기의 영상 크기에서 모두 정확도 향상을 보였고, 이는 치환증강을 통한 물체인식 정확도 향상이 다양한 조건에서 일반화 될 수 있음을 확인하는 결과이다.

Table 4. Per class results on substitution augmentation

영상 크기	치환증강 적용여부	클래스								
		A	B	C	D	E	F	G	H	mAP
416	×	64.8	89.15	69.43	79.23	81.62	75.74	**75.08**	**75.93**	76.37
	○	**69.31**	**89.55**	**70.46**	**86.09**	**84.06**	**78.32**	73.58	75.7	**78.38**
512	×	**69.12**	89.17	**71.9**	83.25	84.29	77.2	**76.71**	**75.8**	78.43
	○	69.08	**90.06**	71.16	**88.02**	**87.46**	**83.43**	74.61	74.37	**79.77**
608	×	**71.42**	89.52	**74.42**	83.63	82.69	74.29	**77.93**	75.22	78.64
	○	67.61	**90.66**	74.13	**87.63**	**87.92**	**81.41**	77.52	**80.79**	**80.96**

Table 5. mAP results on different neural network architectures

모델	영상 크기	치환증강 적용여부	mAP
RetinaNet	512	×	53.75
RetinaNet	512	○	55.35
YOLOv3	512	×	76.97
YOLOv3	512	○	77.60
EfficientNet B0	416	×	64.02
EfficientNet B0	416	○	65.89

4. 결 론

인공지능 기술의 급격한 발전으로 물체 인식 정확도가 매우 높아졌고, 이러한 기술은 차량 및 드론에서의 인식을 통해 인간의 인식능력을 보조하고, 나아가 자율 운용기술에 활용될 수 있다. 본 연구에서는 물체 인식 학습에 있어 주어진 데이터를 증강하여 물체 인식기의 정확도를 높이는 방안에 대한 연구를 진행하였다. 임의의 물체를 임의의 배경에 치환하는 일반적인 증강방법과는 다르게, 유사한 물체에 대해 선별하여 유사한 물체끼리만 위치를 치환하고, 남은 잔여 부분에 대해 배경과 유사한 내용으로 칠하는 방법을 사용하여 실제 영상과 매우 유사한 증강 영상을 만드는 것을 목표로 하였다. 군용 기동장비 데이터셋에 대한 실험 결과는 본 연구에서 제시한 치환증강이 물체 인식기의 성능 향상에 도움이 된다는 것을 보였다. 향후 본 연구의 결과를 다른 증강기법과 연계하여 물체 영역에 대한 추가적인 증강을 통해 물체 인식 정확도를 더 높일 수 있는 방안에 대해 연구할 예정이다. 또한 물체의 방향, 카메라의 촬영각도 등에 대한 데이터를 포함하여 이를 활용한 더욱 정교한 치환 증강이 가능할 것으로 보이고, 이러한 경우의 정확도 향상에 대한 연구를 수행할 예정이다.

References

[1] A. Bochkovskiy, C. Y. Wang and H. Y. M. Liao, "Yolov4: Optimal Speed and Accuracy of Object Detection," arXiv preprint arXiv:2004.10934, 2020.

[2] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár and C. L. Zitnick, "Microsoft Coco: Common Objects in Context," European Conference on Computer Vision, pp. 740-755, September, 2014.

[3] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn and A. Zisserman, "The Pascal Visual Object Classes(VOC) Challenge," International Journal of Computer Vision, Vol. 88, No. 2, pp. 303-338, 2010.

[4] T. DeVries and G. W. Taylor, "Improved Regularization of Convolutional Neural Networks with Cutout," arXiv preprint arXiv:1708.04552., 2017.

[5] H. Zhang, M. Cisse, Y. N. Dauphin and D. Lopez-Paz, "Mixup: Beyond Empirical Risk Minimization," arXiv preprint arXiv:1710.09412. 2017.

[6] A. Krizhevsky and G. Hinton, "Learning Multiple Layers of Features from Tiny Images," 2009.

[7] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe and Y. Yoo, "Cutmix: Regularization Strategy to Train Strong Classifiers with Localizable Features" In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6023-6032, 2019.

[8] S. Lin, T. Yu, R. Feng, X. Li, X. Jin and Z. Chen, "Local Patch AutoAugment with Multi-Agent Collaboration." arXiv preprint arXiv:2103.11099, 2021.

[9] G. Ghiasi, Y. Cui, A. Srinivas, R. Qian, T. Y. Lin, E. D. Cubuk, Q. V. Le, B. Zoph, "Simple Copy-Paste is a Strong Data Augmentation Method for Instance Segmentation," In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2918-2928, 2021.

[10] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan and Q. V. Le, "Autoaugment: Learning Augmentation Strategies from Data," In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 113-123, 2019.

[11] E. D. Cubuk, B. Zoph, J. Shlens and Q. V. Le,

"Randaugment: Practical Automated Data Augmentation with a Reduced Search Space," In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 702-703, 2020.

[12] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-Cnn: Towards Real-Time Object Detection with Region Proposal Networks," Advances in Neural Information Processing Systems, 28, pp. 91-99, 2015.

[13] M. Tan, R. Pang and Q. V. Le, "Efficientdet: Scalable and Efficient Object Detection," In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10781-10790, 2020.

[14] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan and S. Belongie, "Feature Pyramid Networks for Object Detection," In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2117-2125, 2017.

[15] J. Hu, L. Shen and G. Sun, "Squeeze-and-Excitation Networks," In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7132-7141, 2018.

[16] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy and A. L. Yuille, "Deeplab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected Crfs," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 40, No. 4, pp. 834-848, 2017.

[17] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," In International Conference on Machine Learning, pp. 448-456, June, 2015.

[18] I. Loshchilov and F. Hutter, "Sgdr: Stochastic Gradient Descent with Warm Restarts," arXiv preprint arXiv:1608.03983, 2016.

[19] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," The Journal of Machine Learning Research, Vol. 15, No. 1, pp. 1929-1958, 2014.

[20] A. Telea, "An Image Inpainting Technique based on the Fast Marching Method," Journal of Graphics Tools, Vol. 9, No. 1, pp. 23-34, 2004.

[21] M. Bertalmio, A. L. Bertozzi and Sapiro, G., "Navier-Stokes, Fluid Dynamics, and Image and Video Inpainting," In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE, December, 2001.

[22] C. Park. "Kia Makes Korean High-Performance Military Vehicles," The Kyunghyang Shinmun, November 4, 2012, https://m.khan.co.kr/economy/auto/article/201211042143515. accessed November 19, 2021.

[23] KIA, "Kia Corporation's Special Vehicle Website," KIA Special Vehicle, 19 Nov. 2021, https://special.kia.com/en/kia/subpage/models-km450/Cargo-Truck.do#.Ya8Dc9BByUk. accessed 19 Nov. 2021.

[24] T. Y. Lin, P. Goyal, R. Girshick, K. He and P. Dollar, "Focal Loss for Dense Object Detection," Proceedings of the IEEE International Conference on Computer Vision, 2017.

[25] J. Redmon and F. Ali, "Yolov3: An Incremental Improvement," arXiv preprint arXiv:1804.02767, 2018.

[26] M. Tan and Q. Le, "Efficientnet: Rethinking Model Scaling for Convolutional Neural Networks," International Conference on Machine Learning, PMLR, 2019.

[27] H. Wang, Q. Wang, F. Yang, W. Zhang and W. Zuo, "Data Augmentation for Object Detection via Progressive and Selective Instance-Switching," arXiv preprint arXiv:1906.00358, 2019.