

CNN 모델 평가를 위한 이미지 데이터 증강 도구 개발[†]

(Development of an Image Data Augmentation Apparatus to Evaluate CNN Model)

최영원[‡] 이영우[§] 채흥석[¶]
(Youngwon Choi) (Youngwoo Lee) (Heung-Seok Chae)

요약 CNN 모델이 이미지 분류와 객체 탐지 등 여러 분야에 활용됨에 따라, 자율주행자동차와 같이 안전필수시스템에 사용되는 CNN 모델의 성능은 신뢰할 수 있어야 한다. 이에 CNN 모델이 다양한 환경에서도 성능을 유지하는지 평가하기 위해 배경을 변경한 이미지를 생성하는 이미지 데이터 증강 도구를 개발한다. 이미지 데이터 증강 도구에 객체가 존재하는 이미지를 입력하면, 해당 이미지로부터 객체 이미지를 추출한 후 수집한 배경 이미지 내에 객체 이미지를 합성하여 새로운 이미지를 생성한다. CNN 모델 성능 평가 방법으로 개발한 도구를 사용하여 기존 테스트 이미지로부터 새로운 테스트 이미지를 생성하고, 생성한 새로운 테스트 이미지로 CNN 모델을 평가한다. 사례 연구로 Pascal VOC2007 테스트 데이터로부터 새로운 테스트 이미지를 생성하고, 새로운 테스트 이미지로 YOLOv3 모델을 평가하였다. 그 결과 기존 테스트 이미지의 mAP보다 새로운 테스트 이미지의 mAP가 약 0.11 더 낮아지는 것을 확인하였다.

키워드 : 메타모픽 테스트, 데이터 어그멘테이션, 합성곱 신경망, 이미지 합성

Abstract As CNN model is applied to various domains such as image classification and object detection, the performance of CNN model which is used to safety critical system like autonomous vehicles should be reliable. To evaluate that CNN model can sustain the performance in various environments, we developed an image data augmentation apparatus which generates images that is changed background. If an image which contains object is entered into the apparatus, it extracts an object image from the entered image and generates composed images by synthesizing the object image with collected background images. As a method to evaluate a CNN model, the apparatus generates new test images from original test images, and we evaluate the CNN model by the new test image. As a case study, we generated new test images from Pascal VOC2007 and evaluated a YOLOv3 model with the new images. As a result, it was detected that mAP of new test images is almost 0.11 lower than mAP of the original test images.

Key words : Metamorphic Testing, Data Augmentation, CNN, Image Composition

1. 서론

Convolutional Neural Network(CNN)은 이미지 분류와 객체 탐지에 널리 사용되는 모델이다[1]. CNN을 이용한 이미지 인식 알고리즘은 2012년에 AlexNet을 시작으로 계속 발전하였다. AlexNet이 2012년 ImageNet Large Scale Visual Recognition Competition(ILSVRC)에서 에러율 15.3%를 달성하였으며[2], 이후 2014년에

GoogLeNet이 에러율 6.67%를[3], 2015년에 ResNet이 에러율 3.57%를 달성하였다[4].

객체 탐지는 이미지나 영상 속 객체를 탐지하여 해당 객체를 분류한 Label과 Bounding Box를 출력하는 것이다. 객체 탐지 알고리즘도 이미지 인식 알고리즘처럼 CNN을 활용하여 문제를 해결하고자 하였다[5]. 이후 R-CNN[6]을 비롯한 객체 탐지 모델이 등장하였고 자동차의 자율주행 기능에 활용되고 있다. 대표적으로 Google의 Waymo가 자율주행택시를 상용 서비스하고 있다. 그러나 Waymo의 무인 자율주행택시는 법적으로 애리조나주 피닉스 지역에 한하여 서비스되고 있다. 이는 자율주행자동차에 사용되는 CNN 모델의 학습/테스트를 위한 이미지 데이터가 여러 환경을 반영하지 않는 것으로 추정한다. 인공신경망 모델을 학습하고 성능을 평가함에 있어 학습 데이터와 테스트 데이터가 영향을 미친다. 데이터 셋의 데이터 수가 많더라도 특정 Label 데이터가 높

[†] 이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2019R1F1A1063336).

[‡] 비회원 : 부산대학교 전기전자컴퓨터공학과
fenrir94@pusan.ac.kr

[§] 학생회원 : 부산대학교 전기전자컴퓨터공학과
amorepooh@pusan.ac.kr

[¶] 종신회원 : 부산대학교 전기전자컴퓨터공학부 교수
hschae@pusan.ac.kr

논문접수 : 2020년 02월 28일

심사완료 : 2020년 03월 01일

은 비율을 차지하거나 낮은 비율을 차지하는 데이터 셋으로 모델을 학습시킬 경우에는 Overfitting이 발생할 수 있으며, 해당 데이터 셋을 테스트에 사용하는 경우에는 특정 Label에 대한 테스트만 많이 수행하고 그 이외 Label에 대해서는 테스트를 적게 수행함에 따라 모델을 테스트한 결과를 신뢰할 수 없다. 테스트 데이터의 양이 많더라도 현실에서 발생하는 상황을 반영하지 못한다면, 테스트 정확도가 높게 측정되었다라도 현실의 데이터에 대한 정확도는 낮게 측정된다. 이러한 문제는 자율주행자동차의 CNN 모델에서도 발생할 수 있으므로, 여러 환경에 대하여 CNN 모델을 테스트해야한다.

여러 환경에 대하여 자율주행자동차의 CNN 모델을 테스트하는 방법으로 여러 지역에서 직접 자동차를 운행하면서 테스트하는 방법과 CARLA[7]같은 도구를 사용하여 3D 맵을 제작하고 시뮬레이션 환경을 구축하여 테스트하는 방법이 있다. 그러나 여러 지역을 직접 자동차를 운행하는 방식은 테스트 비용이 크며, 3D 맵을 제작하여 테스트하는 방법은 맵을 제작하기 위한 비용이 필요하다.

본 연구에서는 여러 환경에 대한 CNN 모델의 테스트 비용을 감소시키기 위해 배경을 변경하는 이미지 데이터 증강 도구를 개발한다. 이미지 데이터 증강 도구는 객체가 존재하는 이미지를 입력하면 객체의 형태는 유지하면서 배경을 변경한 이미지를 출력한다. 이미지 데이터 증강 도구에 대한 자세한 설명은 3장에서 다룰 것이며, 개발한 도구를 사용한 사례 연구는 4장에서 설명할 것이다.

2. 배경 지식

2.1. Metamorphic Testing

Test Oracle은 Testing 과정에서 입력에 따른 프로그램 실행 결과가 올바른지 판단하는 절차이다[8]. 복잡한 수식을 계산하는 프로그램이나 컴파일러 프로그램과 같이 입력에 따른 예상 출력을 계산하기 어려워 Test Oracle을 수행할 수 없는 문제를 Oracle Problem이라고 한다[8]. Oracle Problem이 있는 경우 테스트 케이스를 생성하기가 어렵고, 그에 따라 프로그램을 테스트하기 어려운 문제가 있다. 이러한 Oracle Problem을 완화하기 위해 고안된 기법이 Metamorphic Testing이다[9].

Metamorphic Testing은 여러 입력에 따른 프로그램 출력을 통해 예상되는 관계로 테스트 케이스를 생성하는 방법이다. 예를 들어, \sin 함수에 12를 입력한다고 할 때 예상 출력을 결정하기 어렵다. 이런 상황에서 \sin 함수의 특성이 $\sin(x) = \sin(\pi - x)$ 이므로, $\sin(12)$ 의 출력을 정확하게 알지 않아도 $\sin(12) = \sin(\pi - 12)$ 로 테스트할 수 있다. 이러한 프로그램 출력을 통해 추론되는 관계를 Metamorphic Relation이라 한다[9].

본 연구에서는 테스트 이미지 수를 증가시키는 방법으로 합성 이미지 생성과 Data Augmentation 기법을 적용한다. 예를 들어, 테스트 이미지에 개가 들어있을 경우, 개의 이미지를 추출하고, 추출한 개 이미지를 배경 이미

지와 합성하여 합성 이미지를 생성한다. 생성된 합성 이미지에는 개가 존재하므로 이미지 분류 또는 객체 탐지 모델은 합성 이미지의 Label을 테스트 이미지의 Label과 동일하게 개로 판별할 수 있어야 한다. 또한 합성 이미지에 Data Augmentation 기법을 적용해도 합성 이미지 내 객체의 형태는 그대로 유지된다. 이에 CNN 모델의 Metamorphic Relation으로 식 테스트 이미지의 Label과 합성 이미지의 Label이 같음을 적용한다.

2.2. 합성 이미지 생성

합성 이미지 생성은 CNN 모델을 평가하기 위해 테스트 이미지로부터 객체 이미지를 추출하고, 추출한 객체 이미지를 배경 이미지와 합성하여 새로운 테스트 이미지를 생성하는 방법이다. 합성 이미지 생성은 객체 이미지를 추출과 이미지 합성 두 단계로 이루어진다. 이미지 추출은 객체가 존재하는 이미지에서 객체만 객체 이미지로 추출하는 방법이다. 이미지 합성은 객체 이미지와 수집한 배경 이미지를 합성하고 Data Augmentation 기법을 적용하여 새로운 테스트 이미지를 생성하는 방법이다.

2.2.1. 이미지 추출

이미지 추출은 객체가 존재하는 이미지에서 배경이 없는 객체 이미지를 추출하는 방법이다. 이미지 추출 방법으로 Canny Edge Detection과 GrabCut 알고리즘, Object Detection과 Instance Segmentation이 있다.

Canny Edge Detection은 이미지 내 Edge를 탐색하는 알고리즘으로[10], 해상 알고리즘을 사용해서 이미지 내 객체를 추출할 수 있다. 하지만 Canny Edge Detection을 적용할 때 threshold 값에 따라 객체의 Edge가 판별이 안 되는 경우가 있을 수 있으며, 객체의 일부분만 추출될 수 있다는 단점을 가지고 있다.

GrabCut 알고리즘은 이미지에서 전경을 추출하기 위해 개발된 알고리즘으로, 사용자가 객체를 중심으로 직사각형 Bounding Box를 그리면 객체의 이미지를 추출할 수 있다[11]. 또한 객체의 Bounding Box 내 배경 이미지를 완전히 제거하려면 사용자가 배경과 전경을 설정하는 작업이 필요하여 이미지 생성을 자동화하기 위한 이미지 데이터 증강 도구에 적절하지 않다.

Object Detection은 학습된 Label에 따라 이미지 내 객체를 탐지하여 Bounding Box를 출력하는 방법이다. Object Detection 기법을 할 수 있는 인공지능망 모델로는 YOLOv3[12]와 Faster R-CNN[13] 등 여러 모델이 있다. 인공지능망을 사용하기에 학습된 Label에 대해서만 객체를 탐지할 수 있으며, Bounding Box에 따라 이미지 내 객체 이미지를 추출하면 객체 이외 영역에 기존 이미지의 배경 이미지가 남아있다는 단점이 있다.

Instance Segmentation은 이미지 내 객체의 형태에 따른 Mask 정보로 색을 입히는 방법으로, Instance Segmentation 기능을 하는 모델로 Mask R-CNN이 있

다[14]. Mask R-CNN 모델은 Classification, Object Detection, Instance Segmentation 기능을 가지고 있어, 입력된 이미지의 Mask 정보와 Bounding Box 정보를 사용하여 객체의 형태를 유지하면서 배경 이미지를 제거한 객체 이미지를 추출할 수 있다. Object Detection처럼 학습된 데이터 셋의 Label에 대해서만 객체 이미지를 추출 가능하다는 단점이 있다.

본 연구에서는 객체 이미지 추출을 위한 이미지 추출 방법으로 Instance Segmentation을 사용한다. Instance Segmentation은 Mask 정보를 활용해서 자동으로 배경을 제거하면서 객체 이미지를 추출할 수 있다.

2.2.2. 이미지 합성

이미지 합성은 이미지 추출로 추출한 객체 이미지와 배경 이미지를 합성하는 과정이다. 이미지 합성하는 방법으로는 이미지 더하기와 블랜딩(Blending), 이미지 비트 연산을 사용한 합성이 있다.

이미지 더하기는 두 이미지의 각 좌표의 RGB값을 각각 더하는 방법이다. 더하기 연산 방법을 적용하여 RGB 값이 255보다 큰 값을 가지게 되는 경우 해당 RGB 값을 255로 나눈 나머지 값 또는 255로 변경한다. 따라서 객체 이미지와 배경 이미지의 RGB 값을 더하는 경우 객체 이미지의 형태가 유지되지 않을 수 있다. 객체 이미지의 형태가 크게 달라지는 경우 Metamorphic Relation을 적용할 수 없는 이미지가 생성된다.

블랜딩은 두 이미지의 RGB값마다 일정 비율을 곱하여 더하는 방법이다. 객체 이미지의 합성 비율이 0에서 1 사이의 실수 α 라면 배경 이미지의 합성 비율은 $1-\alpha$ 이다. 두 이미지의 합성 비율 합이 1이므로 이미지 합성을 하더라도 RGB값이 255를 초과하지 않는다. 그러나 이 방법도 객체 이미지의 형태가 변형될 수 있다.

이미지 비트 연산은 두 이미지를 and나 xor 같은 비트 연산을 적용하여 합성하는 방법이다. 추출한 객체 이미지를 threshold값에 따라 객체 형태의 Mask 정보를 계산하고, Mask 정보를 사용하여 객체 이미지를 배경 이미지와 합성한다. Mask 정보를 사용하여 배경 이미지와 객체 이미지 간에 섞이지 않고 이미지가 합성이 된다. 하지만 객체 이미지에서 객체와 배경의 구분이 명확하지 않다면 객체의 영역 이외 영역도 Mask 정보에 포함될 수 있다.

본 연구에서는 비트 연산을 사용하여 객체 이미지와 배경 이미지를 합성한다. 비트 연산을 사용하면 객체 이미지의 형태를 유지하면서 배경 이미지와 합성이 가능하다. 또한 이미지 추출에서 Instance Segmentation을 적용하여 객체 이미지의 배경을 일정한 값으로 바꾸어 객체와 배경의 구분이 명확하다면, 비트 연산에서 발생하는 Mask 영역 문제를 해결할 수 있다.

2.3. Data Augmentation

Data Augmentation은 이미지에 변형 기법을 적용하

여 변형된 이미지를 생성하는 방법이다. Data Augmentation은 주로 모델 학습 전에 학습 데이터에 여러 Data Augmentation 기법을 적용하여 학습 데이터의 양을 늘려서 모델을 학습시킨다. Data Augmentation을 적용할 때 주로 사용하는 이미지 변형 기법으로 이미지 대칭, 이미지 크기, 이미지 회전, 이미지 밝기 조절 등이 있으며[15], 그림 1은 각 변형 기법에 따른 Data Augmentation 예시로, 한 이미지로부터 4가지 Data Augmentation 기법을 적용하여 4개의 새로운 변형된 이미지를 생성하였다.

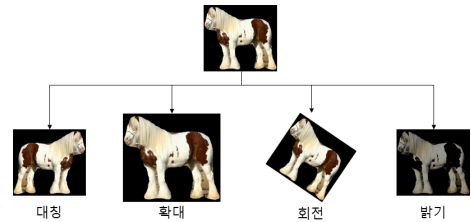


그림 23 Data Augmentation 적용 예시

Data Augmentation을 사용하면 각 테스트 이미지마다 여러 장의 Data Augmentation을 적용한 이미지를 생성할 수 있다. 같은 Data Augmentation 기법을 적용할 때 그 정도에 따라 다른 이미지를 여러 장 생성할 수 있다. 이미지 확대 및 축소의 경우 10%, 20%, 30% 값을 입력하여 크기가 다른 이미지들을 생성할 수 있으며, 회전의 경우 회전 각도를 10° , 20° , 30° 등으로 다르게 적용하여 회전한 정도가 다른 이미지를 생성할 수 있다.

본 연구에서는 테스트 데이터의 양을 증가시키기 위해 Data Augmentation을 적용한다. 평가 대상 CNN 모델은 Metamorphic Relation에 따라 Data Augmentation을 적용한 합성 이미지의 Label과 Data Augmentation을 적용하지 않은 합성 이미지의 Label, 테스트 이미지의 Label은 모두 동일하게 판별해야 한다.

(테스트 이미지의 Label = 합성 이미지의 Label = Data Augmentation을 적용한 합성 이미지 Label)

본 연구에서 적용하는 Data Augmentation 기법들은 아래와 같다. 해당 Data Augmentation 기법들은 imgaug와 scikit-image같은 파이썬 라이브러리에서 지원하는 기법들이다.

- 밝기(Brightness): 이미지의 밝기를 증가 또는 감소함
- 노이즈(Noise): 이미지에 4가지 노이즈 Gaussian Noise, Salt and Pepper Noise, Speckle Noise, Poisson Noise를 적용함
- 날씨(Weather): 비, 안개, 눈 효과를 적용함

3. 이미지 데이터 증강 도구

본 절에서는 기존 테스트 이미지로부터 합성 이미지를 생성하는 도구에 대해서 설명한다. Composed Image Generator는 합성 이미지를 생성하기 위한 도구로, Object Extraction과 Image Composition 2가지 과정으로 이루어져 있다. 그림 2는 Composed Image Generator의 구조와 동작 과정을 나타낸다.

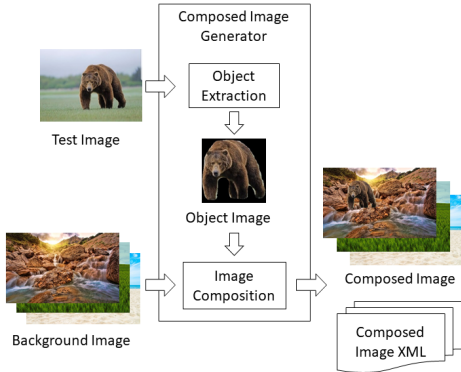


그림 24 Composed Image Generator 동작 과정

Object Extraction 단계에서는 테스트 이미지가 입력 되면 테스트 이미지 내에서 객체를 탐지하여 객체 이미지를 추출한다. Image Composition 단계에서는 추출한 객체 이미지와 배경 이미지를 합성하고, Data Augmentation을 적용한다. Composed Image Generator가 동작 완료하면 합성 이미지와 합성 이미지 내 객체 정보를 저장한 XML 파일이 생성된다.

3.1. Object Extraction

Object Extraction은 Instance Segmentation을 적용하여 테스트 이미지를 입력하면 객체를 탐지하여 객체 이미지를 추출한다. 그림 3은 Object Extraction 단계의 동작 과정을 표현한 것이며, Object Extraction 단계는 그림 3의 아래와 같이 3 단계로 동작한다.

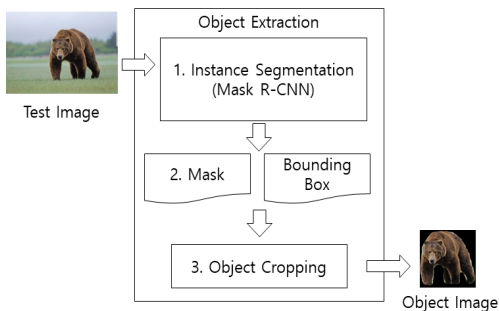


그림 25 Object Extraction 동작 과정

1. 테스트 이미지를 학습된 Mask R-CNN 모델에 입력하여 테스트 이미지 내 객체에 대한 Bounding Box와 Mask 정보를 출력한다.
2. 출력된 Mask 정보를 사용하여 테스트 이미지에서 객체 이외의 영역을 배경으로 가정하고 배경 영역을 검은색으로 변경한다.
3. 배경이 검은색으로 변경된 테스트 이미지에서 객체의 Bounding Box 영역을 객체 이미지로 저장한다.

Object Extraction에서는 Instance Segmentation을 적용하기 위해서 MS COCO Data Set[15]을 학습한 Mask R-CNN 모델을 사용한다. MS COCO Data Set의 객체 Label을 80 종류로, 해당 Mask R-CNN 모델은 이미지가 입력되면 해당 이미지 내 학습된 Label의 객체를 탐지하여 해당 객체의 Bounding Box와 Mask 정보를 출력한다. 그림 4는 Object Extraction 단계에서 Instance Segmentation 기능을 하는 Mask R-CNN을 통해 객체 이미지를 추출하는 예시이다. 그림 4-(a)는 Mask R-CNN에 입력한 이미지이며, 그림 4-(b)는 Mask R-CNN 모델이 그림 4-(a) 내 객체를 탐지하여 출력한 해당 객체의 Label, Bounding Box와 Mask를 그림 4-(a)에 적용한 결과이다.



그림 26 Bounding Box와 Mask 적용 예시

그림 5는 그림 4-(a)에서 추출된 객체 이미지이다. 그림 4-(b)에서 Mask 이외의 영역의 RGB값을 각각 0으로 변경하면 객체 이외의 영역은 검은색이 된다. 그 후 Bounding Box에 따라 이미지를 잘라서 객체 이미지로 저장하면 그림 5가 생성된다.



그림 27 객체 이미지 예시

3.2. Image Composition

Image Composition은 객체 이미지와 배경 이미지를 입력받아 합성 이미지를 생성한다. 그림 6은 Image Composition 단계의 동작 과정을 표현한 것이며, 세부적인 동작 과정은 그림 6 아래 2 단계로 진행된다.

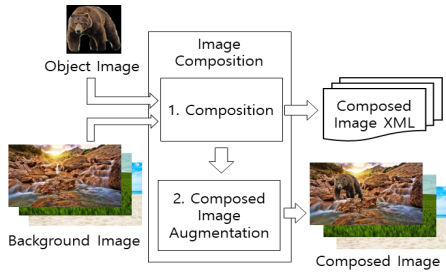


그림 28 Image Composition 동작 과정

1. 객체 이미지를 비트 연산으로 배경 이미지 내 임의의 좌표에 합성하여 저장한다.
2. 합성된 이미지에 설정한 Data Augmentation을 적용한 후 합성 이미지를 저장한다.

Composition 과정에서는 객체 이미지와 배경 이미지를 합성하여 합성 이미지와 합성 이미지 xml을 생성한다. 객체 이미지와 배경 이미지를 합성하기 위해서는 객체 이미지에서 객체의 형태만을 배경 이미지에 합성해야 한다. 이를 위해 객체 이미지를 Grey Scale로 변환한 후 객체의 Mask를 계산하여 해당 객체를 배경 이미지의 임의의 좌표에 합성한다. 객체 이미지의 너비 또는 높이가 배경 이미지보다 큰 경우 배경 이미지와 합성할 수 없으므로 객체 이미지의 너비와 높이를 배경 이미지의 1/3 비율로 축소시킨 후 합성한다.

객체 이미지와 배경 이미지가 합성할 때 객체 이미지의 Label, 너비, 높이, 합성 좌표 정보를 Pascal VOC 포맷에 따라서 XML 파일을 저장한다. 저장한 XML 파일의 정보는 합성 이미지로 모델을 평가하여 출력된 결과와 비교하여 정확도를 측정하기 위해 사용된다.

Composed Image Augmentation 과정에서는 합성된 합성 이미지에 Data Augmentation을 적용한다. 본 연구에서는 합성 이미지에 Data Augmentation 기법으로 Brightness, Noise, Weather를 적용한다. 해당 기법들은 Y. Tian et al.[16]과 M. Koziarski et al.[17]에서처럼 모델 학습 또는 테스트 시 이미지 데이터의 Data Augmentation으로 적용하는 기법들이다. Brightness는 설정한 밝기 값에 따라 이미지의 밝기 값을 변경한다. Noise는 Gaussian Noise와 Salt & Pepper Noise, Poisson Noise, Speckle Noise 총 4가지 노이즈를 이미지에 각각 적용한다. Poisson Noise와 Speckle Noise는 난수 값을 사용한다. Weather는 이미지 합성 방법 중 블렌딩을 사용하여 합성 이미지와 비, 눈, 안개 3가지 날씨별 이미지들을 비는 0.6, 안개는 0.3, 눈은 0.6 비율을 적용하여 합성한다.

표 1 Pascal VOC2007 데이터 셋 Label 별 이미지 수

Label	Bicycle	Bus	Car	Cat	Cow	Dog	Horse	Motorbike	Person	Train
이미지 수	239	178	721	322	127	418	274	222	2007	259

4. 사례 연구

본 절에서는 사례 연구로 CNN 모델을 평가하는 과정과 평가 지표, 테스트 데이터, 평가 대상 모델, 모델 평가 결과에 대해 설명한다. 그림 7은 CNN 모델 평가 과정을 나타내며, 아래 3 단계로 진행된다.

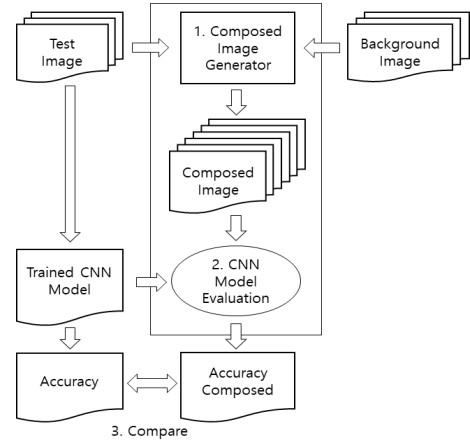


그림 29 CNN 모델 평가 과정

1. Composed Image Generator에 테스트 이미지와 수집한 배경 이미지를 입력하여 합성 이미지를 생성한다.
2. 합성 이미지로 평가 대상 모델의 성능을 측정한다.
3. 합성 이미지로 측정된 모델의 성능과 기존 테스트 이미지로 측정된 모델의 성능을 비교한다.

4.1. 평가 환경

4.1.1 테스트 데이터

테스트 데이터는 Composed Image Generator에서 MS COCO 데이터 셋을 학습한 Mask R-CNN을 사용함에 따라 MS COCO 데이터 셋의 Label을 가지는 데이터 셋을 사용해야 한다. 이에 테스트 이미지로 이미지 인식 대회 Pascal VOC Challenge에서 학습 및 테스트 데이터로 사용된 Pascal VOC 2007[18]의 테스트 데이터 중 Label 10개를 사용한다. 테스트 이미지의 전체 데이터 수는 3707장이며, 이미지 내 객체 크기가 다양하게 분포되어 있다. 이미지의 최대 높이와 너비는 각 500 pixel이다. 표 1은 Pascal VOC2007 데이터 셋의 Label과 Label 마다 Test Image 수를 나타내며, 일부 이미지는 Label 객체가 2개 이상 존재한다.

4.1.2 평가 대상 모델

평가 대상 모델은 YOLOv3 모델을 사용한다. YOLOv3는 Darknet-53으로 구성된 Object Detection 모델이며, Darknet-53은 Convolutional Layer 53개와 Residual Layer 23개 등으로 이루어져 있다. Darknet-53으로 구성된 YOLOv3는 ResNet-101과 ResNet-152과 비교하여 유사한 성능을 내면서 더 빠른 처리 속도를 보인다[12]. 다른 객체 탐지 모델들보다 빠른 처리 속도로 인해 YOLOv3는 자율주행자동차의 객체 탐지 모델로 연구되고 있다[19, 20]. 평가 대상 모델로 사용하는 YOLOv3 모델은 IoU(Intersection over Union) Threshold를 0.5로 설정하여 Pascal VOC2007과 Pascal VOC2012로 학습된 모델이다.

4.1.3 배경 이미지

배경 이미지는 무료 이미지 사이트 Pixabay에서 Crawling한 후 이미지 분류 Tag에 따라 이미지를 선택한다. 배경 이미지를 선택할 때는 배경 이미지 내 객체가 합성될 객체 이미지의 객체를 탐지하는 것에 장애가 되지 않아야 하므로 객체가 존재하지 않거나 탐지하기 어려운 객체만 존재하는 이미지를 선택한다. 배경 이미지의 Tag와 그 수는 표 2와 같으며, 총 24장을 선택하였다. 그림 8은 선택한 배경 이미지의 예시이다. 배경 이미지의 높이는 340 pixel로 동일하며, 너비는 배경 이미지에 따라 다르며 최소 256 pixel, 최대 1141 pixel이다.



그림 30 Background Image 예시

4.1.4 합성 이미지

합성 이미지 생성에 사용할 테스트 이미지는 테스트 이미지의 10개 Label마다 각각 이미지 20장을 선택하여, 총 200장을 선택한다. 선택한 테스트 이미지 20장과 배경 이미지 24장을 Composed Image Generator에 입력하여 합성 이미지 4,800장을 생성한다.

Composed Image Generator는 생성한 합성 이미지에 Data Augmentation 기법으로 Brightness와 Noise, Weather를 적용한다. Brightness는 밝기를 -100, -50, +50, +100 4가지 밝기 값으로 조절하여 합성 이미지 Brightness를 생성한다. Noise는 Gaussian Noise와 Salt

& Pepper Noise, Poisson Noise, Speckle Noise 총 4가지 노이즈를 적용하여 합성 이미지 Noise를 생성한다. Weather 기법은 비, 안개, 눈에 대해 각각 2가지 효과를 적용하여 합성 이미지 Weather를 생성한다. 각 기법을 적용한 합성 이미지의 수는 아래와 같다.

- 합성 이미지 Brightness:
4,800 images * 4 brightness = 19,600 images
- 합성 이미지 Noise:
4,800 images * 4 noise = 19,600 images
- 합성 이미지 Weather:
4,800 images * 3 weather * 2 effects/weather = 28,800 images

그림 9는 합성 이미지와 합성 이미지에 Data Augmentation 기법들을 적용한 이미지의 예시이다. 그림 9-(a)는 배경만 변경된 합성 이미지이고, 그림 9-(b)는 그림 9-(a)에 Brightness -100을 적용한 합성 이미지 Brightness이다. 그림 9-(c)는 그림 9-(a)에 Noise 기법 중 Speckle Noise를 적용한 합성 이미지 Noise이고, 그림 9-(d)는 Weather의 비 효과를 그림 9-(a)에 적용한 합성 이미지 Weather이다.

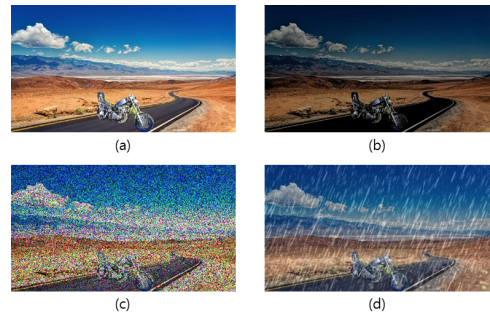


그림 31 합성 이미지 예시

4.1.4 평가 지표

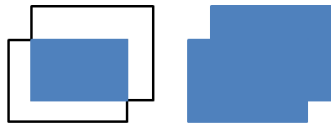
모델의 성능 평가 지표는 Pascal VOC2007 테스트 데이터와의 성능 비교를 위하여 mean Average Precision(mAP)과 각 Label의 Average Precision(AP)을 사용한다. 본 연구에서 개발한 도구에서 생성하는 이미지는 객체가 1개 존재하지만 Pascal VOC2007 테스트 데이터에는 여러 객체가 존재하는 이미지도 있으므로 mAP과 AP를 사용한다. Average Precision은 Label 별 Recall에 대한 Precision의 평균이며, mean Average

표 2 배경 이미지 Tag 별 이미지 수

Tag	Arctic	City	Desert	Field	Forest	Mountain	Road	Village	총합
이미지 수	3	3	4	2	3	3	4	2	24

Precision은 Average Precision의 평균이다.

객체 탐지 판단을 위한 IoU(Intersection over Union) Threshold 값은 평가 대상 모델이 학습할 때와 같이 0.5를 적용한다. IoU Threshold는 그림 10처럼 객체의 Bounding Box와 모델이 예측한 Bounding Box로 계산한다. IoU Threshold가 0.5라는 것은 실제 객체의 Bounding Box와 모델이 예측한 객체의 Bounding Box가 겹치는 영역이 두 Bounding Box를 합친 영역의 1/2 이상인 경우 객체를 탐지했다고 판단하여 객체를 Label에 따라 분류를 하고, 그렇지 않은 경우 객체를 탐지하지 못했다고 판단한다.



IoU Threshold = Overlapping Region / Combined Region

그림 32 IoU Threshold 계산식

4.2. 평가 결과

모델 평가는 테스트 이미지와 합성 이미지, 합성 이미지 Brightness, 합성 이미지 Noise, 합성 이미지 Weather로 각각 평가 대상 모델 YOLOv3를 평가하여 출력된 mean Average Precision(mAP)과 각 Label의 Average Precision(AP)으로 분석한다. 표 3은 각 테스트 데이터들을 사용하여 평가 대상 모델인 YOLOv3 모델을 평가한 결과 mAP와 각 Label의 AP를 정리한 표이다.

4.2.1 테스트 데이터 별 mAP 분석

테스트 이미지에 대한 mAP보다 합성 이미지에 대한 mAP가 더 낮게 측정되었다. 합성 이미지 Brightness와 합성 이미지 Noise, 합성 이미지 Weather에 대한 모델의 mAP는 합성 이미지의 mAP보다 더 낮아진 것을 확인할 수 있으며, 합성 이미지 Brightness, 합성 이미지 Weather, 합성 이미지 Noise 순서로 mAP가 작아진다.

이를 통해 평가 대상 모델은 테스트 데이터 중 다른 배경 이미지와 합성되어 생성된 합성 이미지의 mAP가 테스트 데이터의 mAP보다 낮아졌으며, 합성 이미지에

Data Augmentation을 적용한 데이터 셋들의 mAP가 더 많이 낮아지는 것을 확인하였다.

4.2.2 테스트 데이터 별 AP 분석

테스트 이미지의 AP와 비교하여, 합성 이미지에서 Bus와 Cow, Person의 AP는 다른 Label보다 상대적으로 적게 감소하였으며 절대적인 감소 수치도 0.05 미만이다. 반면에 Train의 AP가 약 0.23으로 가장 크게 감소하였고 Bicycle의 AP가 약 0.20으로 두 번째로 많이 감소하였다. Cat과 Dog의 AP 또한 0.16 이상 감소하였다.

이러한 결과를 통해 평가 대상 모델은 Bus와 Cow, Person에서 배경이 변경하여도 성능을 거의 유지할 수 있을 정도로 학습이 잘 되었다고 간주되며, Train과 Bicycle, Cat, Dog에서 배경이 변경된 이미지에 대한 학습이 부족한 것으로 간주된다.

합성 이미지의 AP와 비교하여, 합성 이미지 Brightness와 합성 이미지 Noise, 합성 이미지 Weather의 AP가 더 낮게 측정되었다. 합성 이미지 Brightness의 AP가 다른 두 Data Augmentation을 적용한 데이터 셋들보다 상대적으로 적게 감소하였으며, 합성 이미지 Noise의 AP는 다른 두 Data Augmentation 기법을 적용한 데이터 셋보다 전체적으로 합성 이미지의 AP에서 더 많은 폭으로 AP가 감소하였다. 예외로 Cat의 경우 합성 이미지 Weather에 대해 AP가 가장 많이 감소하였다. Cow와 Person의 경우 합성 이미지 Noise와 합성 이미지 Weather의 mAP가 약 0.02 차이를 보이며 다른 Label에 비해 그 차이가 상대적으로 작다.

이러한 결과를 통해 적용한 Data Augmentation 기법 중 Noise가 Brightness와 Weather보다 더 객체를 식별하기 힘들게 이미지를 변형했다고 추정한다. Noise 중에서도 Speckle Noise가 가장 객체를 식별하기 힘들게 변형시켰다고 판단한다. 또한 Brightness는 Noise와 Weather보다 덜 객체를 변형시켰다고 추정한다. 그러나 본 사례 연구에서는 배경이 변경된 이미지에 Data Augmentation을 적용함에 따라 이미지 변형에 시너지가 발생했을 가능성이 있다. 이는 기존 테스트 이미지에 Data Augmentation 기법만 적용한 이미지로 평가 대상 모델을 측정해야 보다 객관적으로 분석할 수 있다.

표 3 테스트 데이터 별 mAP와 각 Label의 AP

테스트 데이터	mAP	Label(AP)									
		Bicycle	Bus	Car	Cat	Cow	Dog	Horse	Motorbike	Person	Train
테스트 이미지	0.930	0.930	0.953	0.943	0.940	0.889	0.928	0.931	0.939	0.914	0.935
합성 이미지	0.816	0.735	0.906	0.861	0.759	0.877	0.760	0.783	0.883	0.901	0.697
합성 이미지 Brightness	0.735	0.686	0.871	0.826	0.673	0.738	0.667	0.685	0.802	0.819	0.585
합성 이미지 Noise	0.645	0.588	0.716	0.746	0.588	0.659	0.581	0.603	0.690	0.754	0.522
합성 이미지 Weather	0.698	0.671	0.850	0.811	0.542	0.676	0.634	0.654	0.793	0.773	0.578

4.2.3 관련 연구와의 비교

본 사례 연구를 통하여 본 연구에서 개발한 이미지 데이터 증강 도구로 생성한 이미지가 기존 테스트 이미지 보다 평가 대상 모델의 성능이 낮게 측정되는 것을 확인하였다. 기존 연구와 비교하여, 파이썬 라이브러리 scikit-image와 imgaug는 다양한 Data Augmentation 기법을 제공하지만 이미지의 배경을 다른 배경으로 변경하는 기능은 지원하지 않는다. Alhaija et al.[21]은 AR을 사용하여 여러 배경을 적용한 현실적인 이미지를 생성할 수 있다. 그러나 이미지 생성을 위해 360도 파노라마 이미지가 필요하며 객체 Label이 명시되지 않은 경우 수동으로 이미지 내 객체 Labeling 작업을 수행하므로 Alhaija et al.[21]은 본 연구에서 개발한 도구보다 이미지 생성을 위한 비용이 크다고 추정할 수 있다.

본 연구에서 부족한 점은 생성된 이미지가 올바르게 합성되어 테스트 데이터로 사용가능한 이미지라고 정량적으로 판단할 수 있는 기준이 없다. 본 연구의 도구에서 객체 이미지와 배경 이미지를 합성할 때 객체와 배경 간의 현실성을 고려하지 않고 배경 이미지 내 임의의 좌표에 객체 이미지를 합성한다. 예를 들어 자동차 객체 이미지와 해안 도로 배경 이미지를 합성할 경우 자동차가 바다 위에 존재하는 이미지가 생성될 수 있다. 이러한 비현실적인 이미지들을 직접 보면서 판단하지 않고 자동으로 검출하거나 생성되지 않도록 보완해야한다.

5. 결론 및 향후 연구

본 연구에서는 CNN 모델의 여러 환경에 대한 테스트 비용을 줄이기 위해 새로운 테스트 이미지를 생성하는 도구를 개발하였다. 테스트 데이터와 배경 이미지, 적용하는 Data Augmentation 기법의 설정 값을 이미지 데이터 증강 도구에 입력하면 자동으로 합성 이미지를 생성한다. 사례 연구로 Composed Image Generator를 사용하여 Pascal VOC2007로부터 배경을 변경한 새로운 테스트 데이터를 생성하여 YOLOv3 모델을 평가하였다. 그 결과 기존 테스트 데이터로 모델을 평가한 결과보다 배경을 변경한 테스트 데이터로 모델의 평가한 결과의 mAP와 AP가 전체적으로 더 낮게 나오는 것을 확인하였다.

본 연구에서는 이미지 합성을 할 때 객체 이미지와 배경 이미지 간의 연관성을 고려하지 않고 배경 이미지 내 임의의 좌표에 합성하였다. 이는 비현실적인 합성 이미지가 생성될 수 있어 모델 평가 시 성능 평가 결과를 신뢰할 수 없을 수 있다. 향후 연구에서는 테스트 이미지의 Label과 배경 이미지의 Tag 간 연관관계를 정의한다. 정의한 연관관계에 따라 합성 여부와 객체 이미지를 배경 이미지에 합성하는 좌표를 결정하여 이미지를 생성하여 모델을 평가하는 연구를 수행할 계획이다.

참 고 문 헌

- [1] P. Molchanov, A. Mallya, S. Tyree, I. Frosio, and J. Kautz, "Importance Estimation for Neural Network Pruning," in *CVPR*, pp. 11264-10272, 2019.
- [2] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *NIPS*, pp. 1097-1105, 2012.
- [3] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going Deeper with Convolutions," in *CVPR*, pp. 1-9, 2015.
- [4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *CVPR*, pp. 770-778, 2016.
- [5] C. Szegedy, A. Toshev, and D. Erhan, "Deep Neural Networks for Object Detection," in *NIPS*, pp. 2553-2561, 2013.
- [6] C. Szegedy, A. Toshev, and D. Erhan, "Deep Neural Networks for Object Detection," in *NIPS*, pp. 2553-2561, 2013.
- [7] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An Open Urban Driving Simulator," in *CoRL*, pp. 1-16, 2017.
- [8] S. Segura, G. Fraser, A.B. Sanchez, and A. Ruiz-Corts, "A Survey on Metamorphic Testing," *IEEE TSE*, Vol. 42, No. 9, pp. 805-824, 2016.
- [9] T. Y. Chen, S. C. Cheung, and S. M. Yiu, "Metamorphic Testing: A New Approach for Generating Next Test Cases," *Computer Science, Hong Kong Univ. of Science and Technology, Tech. HKUST-CS98-01*, 1998.
- [10] J. Canny, "A Computational Approach to Edge Detection," *IEEE TPAMI*, Vol. 8, No. 6, pp. 679-698, 1986.
- [11] C. Rother, V. Kolmogorov, and A. Blake, "'Grabcut': Interactive Foreground Extraction Using Iterated Graph Cuts," in *ACM TOG*, Vol. 23, No. 3, pp. 309-314, 2004.
- [12] J. Redmon, and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv preprint arXiv:1804.02767*, 2018.
- [13] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in *NIPS*, pp. 91-99, 2015.
- [14] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *ICCV*, pp. 2961-2969, 2017.
- [15] T.-Y. Lin, M.Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," in *ECCV*, pp. 740-755, 2014.
- [16] Y. Tian, K. Pei, S. Jana, and B. Ray, "DeepTest: Automated Testing of Deep-Neural-Network-driven Autonomous Cars," in *ICSE*, pp. 303-314, 2018.
- [17] M. Koziarski and B. Cyganek, "Image Recognition with Deep Neural Networks in Presence of

- Noise-Dealing with and Taking Advantage of Distortions," *ICAE*, Vol. 24, No. 4, pp. 337-349, 2017.
- [18] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *IJCV*, Vol. 88, No. 2, pp. 303-338, 2010.
- [19] J. Choi, D. Chun, H. Kim, and H.-j. Lee, "Gaussian YOLOv3: An Accurate and Fast Object Detector Using Localization Uncertainty for Autonomous Driving," in *ICCV*, pp. 502-511, 2019.
- [20] Q. Mao, H. Sun, Y. Liu, and R. Jia, "Mini-YOLOv3: Real-Time Object Detector for Embedded Applications," *IEEE Access*, Vol. 7, No. 1, pp. 133529-133538, 2019.
- [21] H. A. Alhaija, S. K. Mustikovela, L. Mescheder, A. Geiger, and C. Rother, "Augmented Reality Meets Computer Vision: Efficient Data Generation for Urban Driving Scenes," *IJCV*, Vol. 126, No. 9, pp. 961-972, 2018.



최영원

2020년 부산대학교 전기컴퓨터공학부 졸업 (학사). 2020년~현재 부산대학교 전기전자컴퓨터공학과 석사과정. 관심분야는 소프트웨어 공학, 소프트웨어 테스트, 데이터 기반 심층신경망 분석



이영우

2016년 부산대학교 정보컴퓨터공학부 졸업 (학사). 2018년 부산대학교 전기전자컴퓨터공학과 졸업(석사). 2018년~현재 부산대학교 전기전자컴퓨터공학과 박사과정. 관심분야는 소프트웨어 공학, 데이터 기반 결합 분석



채홍석

1994년 서울대학교 원자핵공학과 졸업(학사). 1996년 한국과학기술원 전산학과 졸업(석사). 2000년 한국과학기술원 전산학과 졸업(박사). 2000년~2003년 동양시스템즈 기술연구소 선임연구원. 2003년~2004년 한국과학기술원 전산학과 초빙교수. 2004년~현재 부산대학교 컴퓨터공학과 교수. 관심분야는 객체지향 방법론, 소프트웨어 테스트, 소프트웨어 검증