

객체 검출을 위한 CNN과 YOLO 성능 비교 실험

이용환*·김영섭**†

*원광대학교 디지털콘텐츠공학과, **† 단국대학교 전자전기공학부

Comparison of CNN and YOLO for Object Detection

Yong-Hwan Lee* and Youngseop Kim**†

*Dept. of Digital Contents, Wonkwang University,

**† Dept. of Electronics and Electrical Engineering, Dankook University

ABSTRACT

Object detection plays a critical role in the field of computer vision, and various researches have rapidly increased along with applying convolutional neural network and its modified structures since 2012. There are representative object detection algorithms, which are convolutional neural networks and YOLO. This paper presents two representative algorithm series, based on CNN and YOLO which solves the problem of CNN bounding box. We compare the performance of algorithm series in terms of accuracy, speed and cost. Compared with the latest advanced solution, YOLO v3 achieves a good trade-off between speed and accuracy.

Key Words : Object Detection, CNN (Convolutional Neural Networks), YOLO (You Only Look Once), Computer Vision

1. 서 론

정지영상 또는 동영상에서의 객체 검출(Object Detection)은 영상처리와 컴퓨터 비전 분야에서 기본적인 것들이 많이 활용되는 기술 중에 하나이다[1]. 그러나 자연 환경에서의 이미지는 텍스처와 함께 다양한 모양, 칼라로 구별되는 실세계 적응성을 갖기 때문에 매우 어려운 문제로 봉착된다. 객체 검출의 주요 목적은 특정 이미지에서 물체를 정확하게 탐색하고 검출된 객체가 무엇을 의미하는지를 분류하는 것이다[2]. 이러한 작업은 사람이 육안으로 쉽게 수행할 수 있지만 컴퓨터는 특정 알고리즘을 통해서만 명확하게 구분할 수가 있다. 객체 검출 알고리즘에서는 주요 객체를 탐지하고 해당 객체를 중심으로 경계 박스(Bounding Box)를 표시하여 구분하며[5], 활용분야로는 차량 감지/추적(Vehicle Detection), 감시 시스템(Surveillance) 등

이 있다[3]. 최근에는 얼굴 인식(Face Detection), 이미지 분류(Image Classification) [25, 29], 비디오 인식(Video Recognition), 음성 인식(Voice Recognition) [28] 등과 같은 다양한 응용 분야에서 활용되는 주요 머신 러닝(Machine Learning) 도구로 convolutional 신경망(Convolutional Neural Networks) 알고리즘이 활용되고 있다[4, 5, 30].

초기의 객체 검출 기술은 SIFT(Scale Invariant Feature Transform) [6], SURF(Speeded-Up Robust Features) [9], HOG(Histogram of Oriented Gradients) [14] 기법과 같이 객체가 갖는 특징을 설계하고 검출하는 방법을 주로 사용하였으며[26], 인식을 향상하기 위해 SVM(Support Vector Machine)과 같은 기계 학습 방법을 적용한 연구도 있었다[22, 30]. Alex 논문[10]에서는 convolutional 신경망(CNN)을 적용한 딥러닝(Deep Learning) 기반의 객체 인식을 통해 기존 방식의 성능을 개선하는 방법을 제안하였다.

본 논문에서는 convolutional 신경망 기반의 객체 검출 알고리즘인 CNN 계열과 CNN의 후보 영역 탐지의 문제점을 해결하는 YOLO 계열 알고리즘을 살펴보고, 정확도 및 속

†E-mail: wangcho@dankook.ac.kr

도 측면에서 대표적인 알고리즘의 성능을 비교하여 살펴본다. 이를 통해, YOLO와 Faster R-CNN 알고리즘의 향후 활용을 논의한다.

본 논문의 구성은 다음과 같다. 2장에서 콘볼루션 신경망을 기반한 객체 검출 알고리즘을 살펴보고, 3장에서 실시간 객체 검출이 가능한 YOLO 알고리즘을 살펴본다. 4장에서는 정확도와 처리시간을 기준으로 대표적인 검출 알고리즘의 성능을 비교하여 살펴보고, 5장에서 결론으로 마무리한다.

2. CNN 관련 알고리즘 분석

2.1 Convolutional Neural Network

콘볼루션 신경망(CNN)은 입력된 이미지로부터 위상 특성(Topological Feature)을 추출하는 피드-포워드(Feed-forward) 네트워크 방식을 취한다. CNN은 입력 이미지에서 특성을 제거한 다음, 분류기(Classifier)를 통해 추출된 특징을 분류한다[10]. 따라서 스케일링(Scaling), 변환, 압착(Squeezing), 회전, 변형과 같은 기본 형상 변환에 강인한 특성을 갖는다. CNN 구조는 콘볼루션 계층(Convolution Layer), 풀링 계층(Pooling Layer)과 완전히 연결된 계층(Fully Connected Layer)으로 구성된다[27]. 로컬 링크와 상호 기능의 가중치로 구성된 콘볼루션 계층이 CNN의 핵심 부분이다. 이 계층에서는 입력 함수 표현을 해석하며, 다수의 기능 맵을 구성한다. 2개의 콘볼루션 계층의 중간에 위치한 풀링 계층은 기능 추출 효과를 갖으며, 기능 맵 크기를 줄이고 기능 추출의 견고성을 높이는 역할을 담당한다. 풀링 계층의 기능 맵 크기는 커널의 이동 단계에 따라 결정된다. 콘볼루션 신경망 분류기는 하나 이상의 완전히 연결된 계층이 존재하며, 한 계층의 모든 단일 뉴런은 다음 계층에서 모든 단일 뉴런으로 연결된다[23].

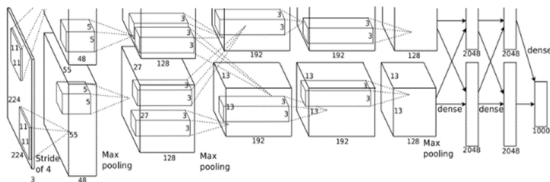


Fig. 1. Architecture of Convolutional Neural Network (CNN) [10].

2.2 Region-based CNN

CNN은 콘볼루션 연산을 통해 픽셀 주변의 지역정보를 표현하는 장점이 있지만, 다수의 객체 인식 및 객체의 위치 검출에는 한계점을 갖는다. R-CNN(Region-based Convolutional Neural

Networks)은 딥러닝 회귀(Regression) 방법을 적용하여 이러한 문제를 해결하였다[2]. R-CNN은 후보 영역(Region Proposal)을 생성하고 이를 기반으로 CNN을 학습시켜 이미지 내 위치한 객체를 검출하며, 3개의 모듈로 구성된다. 첫번째 모듈은 입력 이미지에서 카테고리 독립적인 후보 영역을 생성하며, 검출기에 이용 가능한 후보 검출 영역을 탐지한다. CNN을 구성하는 두번째 모듈은 모든 후보 영역으로부터 동일한 크기의 특징 벡터를 추출하고, 세번째 모듈에서는 클래스 고유의 선형 SVM을 이용하여 후보 영역 내의 객체를 분류한다[11].

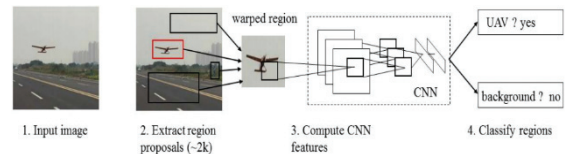


Fig. 2. Region-Convolutional Neural Network (R-CNN) System Overview [11].

2.3 Fast R- CNN

R-CNN은 CNN, SVM과 회귀 학습을 분리하여 수행하기 때문에 연산처리 시간이 오래 걸리는 문제점을 갖는다. 이를 보완하는 Fast R-CNN은 입력 이미지 전체를 하나의 후보 영역으로 간주하여 CNN을 학습한다. 학습된 CNN을 통해 생성된 하나의 컨벤션 기능 맵을 풀링하여 특징을 추출한다[24]. R-CNN과 Fast R-CNN의 주요 차이점으로, R-CNN은 지역적 검출 영역이 픽셀 단위로 입력되는 반면에, Fast R-CNN은 후보 영역을 기능 맵 단위로 입력 받는다.

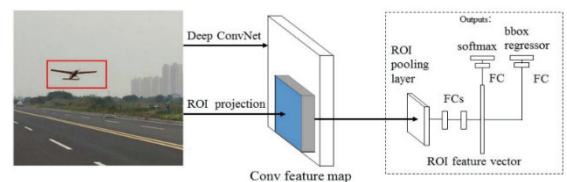


Fig. 3. Fast R-CNN System Overview [24].

2.4 Faster R- CNN

Fast R-CNN은 후보 영역 생성 모듈이 CNN과 독립적으로 별도의 모듈에서 수행되기 때문에 학습 및 실행 속도에서 비효율적인 문제가 있다. 이를 보완하는 Faster R-CNN은 동일한 콘볼루션 네트워크에서 객체 감지 및 후보 영역 생성을 수행한다. 후보 영역 생성에 대한 방법으로, 선택적 탐색(Selective Search) 기법 대신에, 별도의 영역 제안 네트워크(RPN, Region Proposal Network)를 적용하여 추출된 기능 맵을 후보 영역으로 추정하여 객체를 인식한

대[11]. 이전의 모든 CNN 모델과 비교하여, 기능 맵 추출 과정과 후보 영역 생성을 일련의 네트워크에서 수행하며, 기능 맵의 해상도가 입력 이미지의 해상도보다 낮기 때문에 계산량이 크게 감소하는 장점을 갖는다.

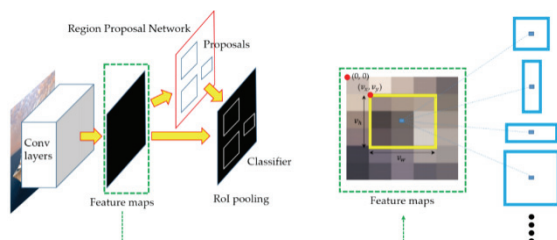


Fig. 4. Structures of Faster R-CNN and RPN [12].

2.5 CNN 관련 알고리즘 성능 비교

CNN 기반의 객체 검출 기술은 CNN, R-CNN, SppNET, Fast R-CNN, Faster R-CNN 순으로 연구 개발되었다. Figure 5 과 같이, R-CNN과 비교하여 Fast R-CNN은 학습과 실험에서 보다 빠른 결과를 얻었다. Fast R-CNN의 실험 시간을 살펴보면, 후보 영역을 생성하는 것과 생성하지 않는 것의 속도 차이가 확연히 나타났다. 이는 Fast R-CNN에서 후보 영역 생성이 성능 상에서 크게 영향을 미친다는 것을 의미한다. Table 1은 R-CNN, Fast R-CNN과 Faster R-CNN의 실행 속도를 비교하여 보여준다. Faster R-CNN은 선택적 검색에서 실행 시간을 감소하고 후보 영역을 추출하는 RPN을 설계하여 검출의 가속화를 얻었다.

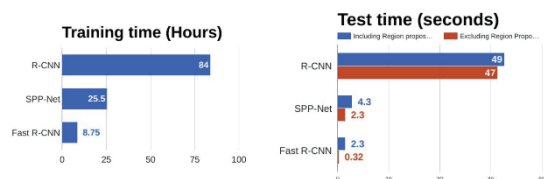


Fig. 5. Comparison of Training Time and Testing Time for CNN [13].

Table 1. Comparison of Speed for CNN [14]

	R-CNN	Fast R-CNN	Faster R-CNN
Test Time per Image	50sec	2sec	0.2sec
Speedup	1x	25x	250x
mAP(VOC2007)	66.0	66.9	66.9

3. YOLO 알고리즘 분석

객체 검출에 대한 다른 접근법으로 YOLO(You Only Look Once)가 있다[15]. 이미지 내에 존재하는 객체와 해당 객체의 위치는 이미지를 한번만 보고 예측할 수 있는 알고리즘이다. 이는 분류 대상의 객체로 감지하는 대신에, 경계 박스를 다차원적으로 분리하고 클래스 확률(Class Probability)을 적용하여 하나의 회귀 문제로 접근한다. 입력 이미지는 CNN을 통해 텐서(Tensor)의 그리드 형태로 나누고, 각 구간에 따라 객체인 경계 박스와 클래스 확률을 생성하여 해당 구역의 객체를 인식한다. YOLO는 후보 영역을 추출하기 위한 별도의 네트워크를 적용하지 않기 때문에 Faster R-CNN보다 처리시간 측면에서 월등한 성능을 보인다.

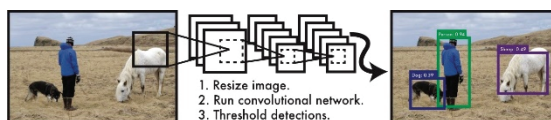


Fig. 6. YOLO Detection System [15].

3.1 YOLO v1

입력 이미지를 $S \times S$ 그리드(grid)로 나누고, 특정 객체의 중심이 그리드 셀의 중심에 해당하면 해당 그리드 셀은 객체 검출에 대한 작업을 수행한다[15]. 그리드 셀은 각각의 그리드에 대해 경계 박스(Bounding Box), 신뢰 점수(Confidence Score)와 클래스 확률을 예측한다. 이러한 예측은 텐서인 $S \times S \times (B * 5 + C)$ 로 계산되며, B 는 경계 박스의 개수, C 는 셀의 조건부 클래스의 수를 나타낸다. 신뢰 점수는 경계 박스 내에 객체가 포함되는지를 예측할 때 모델의 신뢰 수준과 정확도를 나타내는 수치이며, (수식1)과 같이 계산된다.

$$CS = Pr(Obj) * IOU_{pred}^{truth} \quad (1)$$

IOU 는 교차 겹침 결합(Intersection over Union)을 의미하며, 해당 셀에 객체가 존재하지 않으면 신뢰 점수는 0이 되고 객체가 존재하면 Ground Truth와 예측 박스간의 IOU 값을 갖는다. 각 그리드 셀은 조건부 클래스 확률을 따르며, 각각의 경계 박스는 $x, y, w, h, confidence$ 로 구성된다. 실행 시점에 모든 경계 박스에 대해 클래스별 특정 신뢰 점수를 계산하기 위해 조건부 클래스 확률과 경계 박스의 신뢰 점수를 곱하며, (수식2)로 계산된다.

Class Specific CS

$$= \text{Conditional Class Prob} * CS$$

$$= Pr(Class_i|Obj) * Pr(Obj) * IOU_{pred}^{truth} \quad (2)$$

$$= Pr(Class_i) * IOU_{pred}^{truth}$$

YOLO v1의 네트워크 구조는 이미지 분류를 위해 설계된 GoogleNet 모델을 기반으로 하며, 24개의 컨볼루션 계층과 2개의 완전히 연결된 계층으로 구성되며, 풀링 계층은 사용하지 않는다.

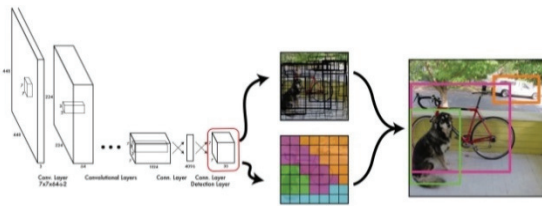


Fig. 7. Network Architecture and Detection Model of YOLO [15].

3.2 YOLO v2

YOLO v2는 대량의 분류 데이터를 활용하기 위해 고안된 방법으로, 공동 훈련 알고리즘을 통해 분류 데이터뿐만 아니라 객체 검출기의 훈련이 가능하다. YOLO v1에 비해 정확도와 속도 향상을 위해 일괄 정규화(Batch Normalization) 계층을 추가하였으며, 초기 학습의 안정화를 위해 컨볼루션 앵커 크기를 설정하였다. 또한 분류 네트워크를 미세 조정하여 고해상도 입력에서도 성능 향상을 도모하였으며, 경계 박스의 예측을 완전히 연결된 계층 대신에 앵커박스에서 수행하여 네트워크를 축소하면서 출력 해상도를 향상시켰다[16].

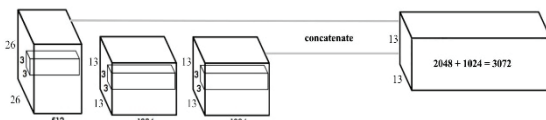


Fig. 8. Passthrough Network Design of YOLO v2 [16].

3.3 YOLO v3

YOLO v3는 로지스틱 회귀(Logistic Regression)를 적용하여 경계 박스의 객관성 점수(Objectness Score)를 예측한다. 이러한 매칭 전략을 통하여 각 경계 박스마다 객체가 있는지 없는지를 산출하는 객관성 점수를 산출하고 앵커 박스와 Ground Truth 박스의 교차 겹침 결합(IOU)가 가장 높은 박스를 1로 매칭시킨다. 경계 박스가 사전에 정의된

특정 임계값 만큼 겹치지 않을 경우 예측은 무시될 수 있다. 또한 경계 박스 예측, 클래스 예측, 특징 검출기 및 반복적 검출 방식을 개선하였으며, 특징 피라미드 네트워크 (Feature Pyramid Network) 개념과 유사한 방식으로 3개의 서로 다른 스케일 박스를 예측하여 다양한 스케일에서 특징을 추출한다. 이를 통해 결합된 특징 맵을 처리하고 보다 큰 텐서를 예측하기 위해 추가적인 컨볼루션 계층이 포함되었다[17].

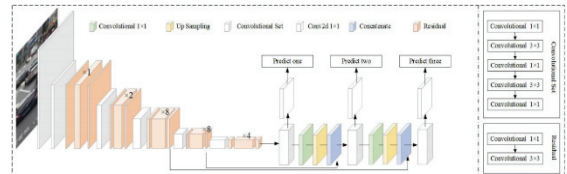


Fig. 9. Structure Detail of YOLO v3 [18].

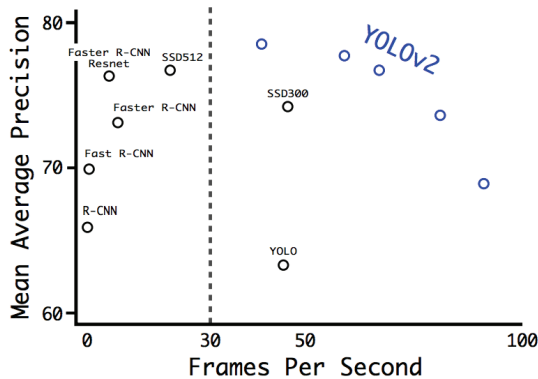
3.4 YOLO 알고리즘 성능 비교

3.4.1 YOLO v1 vs. YOLO v2

Table 2에서 v1과 v2의 성능을 비교하였으며, 설계 변화를 통해 mAP에서 성능 향상을 보였다. 이를 통해, (1) 모든 컨볼루션 계층에 일괄 정규화를 포함시켜 대략 2%의 mAP 개선 효과를 보였다. (2) v2에서 224×224 이미지로 분류기를 훈련시키고 적은 메모리를 사용하여 고해상도 분류기에 적용함에 따라 mAP 4% 향상을 보였다. (3) 앵커 박스를 사용하는 v2는 이미지에 약 1,000개의 박스를 예측하지만 앵커 박스가 없는 v1은 100개 수준의 박스를 예측하였다. v1은 mAP 69.5%에서 81% 회수율(Recall)을 보인 반면에, v2는 69.2% mAP에서 88% 회수율을 보였다. (4) v2는 26×26 해상도에서 이전 계층의 특징을 가져오는 통과(Pass-through) 계층이 추가되어, 저해상도와 고해상도의 특징을 연관시켜준다, 인접 특징을 다른 채널로 배치하여 mAP 1% 향상을 보였다. (5) v2는 사전 식별을 위해 일련의 경계 박스에서 k-means 클러스터링을 수행하여 개선 효과를 보였다. (6) 제한된 위치 예측을 극복하기 위해 다차원 클러스터링을 통해 경계 박스 중심 위치를 직접 예측하여 v2에서 mAP 5% 향상을 보였다. (7) v2에서는 입력 이미지 크기를 반복적으로 다중스케일 훈련을 시키고 동일한 네트워크에서 서로 다른 이미지 해상도에 대한 검출 예측을 통해 작은 크기 이미지에서 보다 빠르고 정확한 성능을 보였다. Figure 10은 VOC2007[20]에서 서로 다른 객체 검출 알고리즘의 속도와 정확도를 측정한 결과이며, mAP와 처리시간을 고려할 때, YOLO v2가 좋은 성능을 보여주었다.

Table 2. Improvements from YOLO to YOLOv2 [16]

	YOLO								YOLOv2
batch norm?		✓	✓	✓	✓	✓	✓	✓	✓
hi-res classifier?			✓			✓	✓	✓	✓
convolutional?				✓	✓	✓	✓	✓	✓
anchor boxes?				✓		✓			✓
new network?					✓				✓
dimension priors?						✓	✓	✓	✓
location prediction?							✓	✓	✓
passthrough?								✓	✓
multi-scale?								✓	✓
hi-res detector?									✓
VOC2007 mAP	63.4	65.8	69.5	69.2	69.6	74.4	75.4	76.8	78.6

**Fig. 10.** Accuracy and Speed of Different Object Detection Models on VOC2007 [19].

3.4.2 YOLO v2 vs. YOLO v3

YOLO v3에서는 특징 추출을 수행하기 위해 Darknet-19와 잔상(Residual) 네트워크를 결합한 새로운 네트워크가 수행된다. v3는 연속적인 3×3 과 1×1 컨볼루션 계층을 사용하며, 53개의 컨볼루션 계층을 사용하여 Darknet-53이라 호칭한다. Darknet-53은 초당 높은 부동소수점 연산을 수행하며, GPU 활용성이 좋아 성능 효율성이 크게 향상되었다. YOLO v3는 동등 성능에서 다른 객체 검출 알고리즘보다 훨씬 빠른 결과를 보였다.

COCO 데이터셋에서 YOLO v3의 mAP 매트릭스는 SSD(Single Shot Multi-box Detector)의 편차를 보이지만 속도 측면에서 3배 수준으로 빠름을 보였으며, $IOU = 0.5$ 에서 기존 검출 매트릭스 맵과 비교하여 강인성을 보였다. 또한 다중스케일 예측의 경우, YOLO v3가 비교적 높은 성능을 보였으며, 객체 검출을 위한 경계 박스 생성에서 다른 알고리즘보다 우수한 검출 성능을 보였다. Table 3은 YOLOv3의 성능을 요약하여 보여준다.

Table 3. Object Detection Single-Model Results vs. state-of-the-art on COCO dataset. [8, 17]

	backbone	AP	AP ₅₀	AP ₇₅	AP _S	AP _M	AP _L
<i>Two-stage methods</i>							
Faster R-CNN++ [5]	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w FPN [8]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN w G-RMI [6]	Inception-ResNet-v2 [21]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN w TDM [20]	Inception-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	52.1
<i>One-stage methods</i>							
YOLOv2 [15]	DarkNet-19 [15]	21.6	44.0	19.2	5.0	22.4	35.5
SSD513 [11, 3]	ResNet-101-SSD	31.2	50.4	33.3	10.2	34.5	49.8
DSSD513 [3]	ResNet-101-DSSD	33.2	53.3	35.2	13.0	35.4	51.1
RetinaNet [9]	ResNet-101-FPN	39.1	59.1	42.3	21.8	42.7	50.2
RetinaNet [9]	ResNeXt-101-FPN	40.8	61.1	44.1	24.1	44.2	51.2
YOLOv3 608 × 608	Darknet-53	33.0	57.9	34.4	18.3	35.4	41.9

4. CNN과 YOLO 성능 비교

본 절에서는 Faster R-CNN과 YOLO을 비교 분석한다. 이들은 CNN을 기반으로 성능 개선을 주도한 대표적인 알고리즘이다. 성능 개선의 모태는 동일하지만, 프레임워크에서 차이점을 갖는다. R-CNN은 전체 이미지를 유지하면서 후보 영역을 나중에 나누는 반면에, YOLO는 CNN을 처리하기 전에 이미지를 나눈다. YOLO는 슬라이딩 윈도우와 지역 후보 영역을 사용하지 않고 입력 이미지를 그리드로 분리한다. Faster R-CNN은 9개의 앵커 박스를 사용하지만, YOLO v2는 5개의 앵커 박스를 사용하며, YOLO v1은 아예 앵커 박스를 사용하지 않는다. 또한 YOLO v2는 정확도 향상을 위해 다차원 클러스터를 생성하지만, Faster R-CNN은 이를 지원하지 않는다.

Table 4는 고속 검출기에 대한 속도와 성능을 비교하여 보여준다. 공용적으로 활용되는 PASCAL VOC 데이터셋에 대해, Fast YOLO가 가장 빠른 처리 속도를 보였으며, YOLO가 다른 버전에 비해 정확도 측면에서 우수한 결과를 보였다. 검출기들의 상대적 mAP와 속도에서 상호 트레이드오프를 보이며, Fast YOLO가 mAP 52.7%에서 다른 검출기보다 2배 정확도를 보였고 YOLO는 실시간 성능을 유지하면서 mAP 66.4%를 보였다.

Table 4. Performance Compare of the Detectors. [7]

	Detection Frameworks	Train	mAP ↑	FPS	PS (mAP×FPS)	PS Order
Less Than Real-Time Detectors	1 Fastest DPM [26]	2007	30.4	15	456	11
	2 R-CNN Minus R [27]	2007	53.5	6	321	13
	3 Faster R-CNN ZF[20]	2007+2012	62.1	18	1118	9
	4 YOLO VGG-16[24]	2007+2012	66.4	21	1394	8
	5 Fast R-CNN[22]	2007+2012	70.0	0.5	35	14
	6 Faster R-CNN VGG-16[20]	2007+2012	73.2	7	512	10
	7 Faster R-CNN ResNet[20]	2007+2012	76.4	5	382	12
Real-Time Detectors	1 Fast YOLO [24]	2007+2012	52.7	155	8169	1
	2 YOLO(YOLOv1)[24]	2007+2012	63.4	45	2853	7
	3 YOLOv2 288×288[24]	2007+2012	69.0	91	6279	2
	4 YOLOv2 352×352[24]	2007+2012	73.7	81	5970	3
	5 YOLOv2 416×416[24]	2007+2012	76.8	67	5146	4
	6 YOLOv2 480×480[24]	2007+2012	77.8	59	4590	5
	7 YOLOv2 544×544[24]	2007+2012	78.6	40	3144	6

Figure 11은 20개 클래스에서 배경과 지역화 오류 (Localization Error)의 비율을 보여준다. 검출 객체에 대한 정확한 지역화에서 YOLO는 Fast R-CNN에 비해 2배 낮은 정확도를 보였다. 이에 비해 Fast R-CNN은 배경 오류가 상대적으로 높게 나타났다.

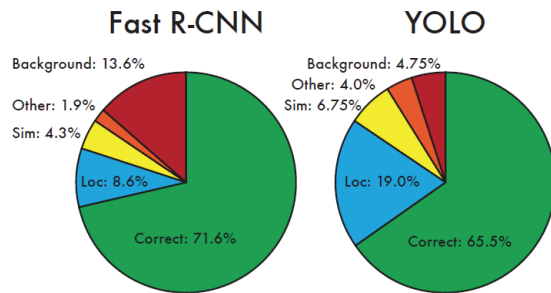


Fig. 11. Error Analysis: Fast R-CNN vs. YOLO [15].

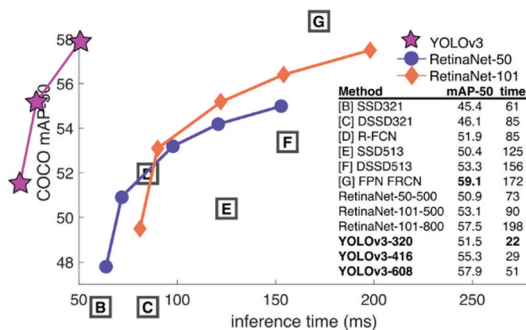


Fig. 12. Comparable Performance of Time [17].

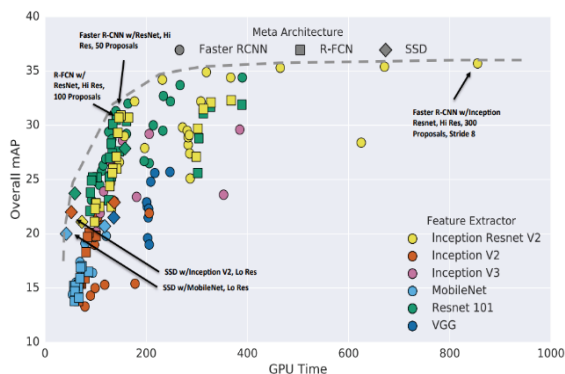


Fig. 13. Accuracy vs. Time for Object Detector [21].

Figure 12는 COCO 데이터셋을 대상으로 객체 검출 알고리즘에 대한 객체 추론시간을 보여준다. SSD 변형과 비슷한 성능에서 속도가 3배 정도 빠르고, RetinaNet 보다는 약간 성능이 떨어지는 것으로 나타났다. 그러나 AP₅₀ 수준에서는 YOLOv3가 가장 강력한 검출 도구임을 성능적으로 보였다.

Figure 13은 동일한 조건에서 실험을 통해 검출 정확도 대비 처리 시간을 비교하였다. 훈련 시간과 정확도에 따라, R-FCN과 SSD는 훈련에 소요되는 시간이 상대적으로 적었지만 낮은 정확도를 보였으며, Faster R-CNN은 높은 정확도를 보이지만 훈련시간이 많이 소요되는 단점을 보였다.

5. 결 론

본 논문에서는 CNN 기반과 YOLO를 포함한 객체 검출 알고리즘에 대해 살펴보았다. YOLO는 통합된 객체 검출 모델로써, CNN과 비교하여 구성이 간단하고 전체 이미지를 직접 학습하여 실제 응용 프로그램에서 활용 가능하기에 적합한 것으로 보인다. 다른 분류기 기반의 접근 방법과 달리, YOLO는 검출 성능에 직접적으로 대응하는 손실 함수에 대해 훈련하고 처리시간 측면에서 실시간 객체 검출이 가능하다. 또한 YOLO는 다른 검출기에 비해 객체 표현을 보다 일반화하여 빠르고 강력한 검출이 가능하다. 이러한 여러 장점을 통해 객체 검출에서 보다 대중화될 가능성이 높다고 판정된다. 여러 검출기의 알고리즘 구조 외에, 머신 러닝의 중요한 이슈는 데이터셋의 범위라 할 수 있다. 적절한 훈련 데이터의 가용성은 응용 프로그램에서 중요한 과정이 되며, 이에 대한 적절성 판별도 알고리즘과 함께 연구되어야 할 것이다.

감사의 글

본 논문은 2019학년도 원광대학교의 교비 지원에 의해 수행됨.

참고문헌

- Asifullah Khan, Anabia Sohail, Umme Zahoora, Aqsa Saeed Qureshi, "A Survey of the Recent Architectures of Deep Convolutional Neural Networks", Computer Vision and Pattern Recognition, Available at <https://arxiv.org/ftp/arxiv/papers/1901/1901.06032.pdf> [Accessed Mar. 13, 2020].
- Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation", IEEE

- Conference on Computer Vision and Pattern Recognition, pp.580-587, 2013.
3. Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, Matti Pietikainen, "Deep Learning for Generic Object Detection: A Survey", *International Journal of Computer Vision*, vol.128, pp.261-318 2020.
 4. Kaimin He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition", *European Conference on Computer Vision*, Part 3, pp.346-361, 2014.
 5. Licheng Jiao, Fan Zhang, Fang Liu, Shuyuan Yang, Lingling Li, Zhizi Feng, Rong Qu, "A Survey of Deep Learning-based Object Detection", *IEEE Access*, vol.7, pp.128837-128868, 2019.
 6. David G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", *International Journal of Computer Vision*, vol.60, pp.91-110, 2004.
 7. Juan Du, "Understanding of Object Detection based on CNN Family and YOLO", *Journal of Physics, Conference Series*, vol.1004, issue.1, 2018.
 8. Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, Piotr Dollar, "Focal Loss for Dense Object Detection", *International Conference on Computer Vision*, pp.2999-3007, 2017.
 9. Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool, "Speeded-Up Robust Features (SURF)", *Computer Vision and Image Understanding*, vol.110, issue.3, pp.346-359, 2008.
 10. Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", *Communications of the ACM*, vol.60, no.6, 2017.
 11. Yurong Yang, Huajun Gong, Xinhua Wang, Peng Sun, "Aerial Target Tracking Algorithm Based on Faster R-CNN Combined with Frame Differencing", *Aerospace*, vol.4, no.32, 2017.
 12. Kwanghyun Kim, Sungjun Hong, Baehoon Choi and Euntai Kim, "Probabilistic Ship Detection and Classification using Deep Learning", *Applied Sciences*, vol.8, no.6, 2018.
 13. Rohith Gandhi, "R-CNN, Fast R-CNN, Faster R-CNN, YOLO - Object Detection Algorithms," 2018. Available at <https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e>. [Accessed: Mar. 13, 2020].
 14. N. Dalal, B. Triggs, "Histograms of Oriented Gradients for Human Detection", *IEEE Conference on Computer Vision and Pattern Recognition*, pp.886-893, 2005.
 15. Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, "You Only Look Once: Unified, Real-time Object Detection", *IEEE Conference on Computer Vision and Pattern Recognition*, pp.779-788, 2015.
 16. Joseph Redmon, Ali Farhadi, "YOLO9000: Better, Faster, Stronger", *IEEE Conference on Computer Vision and Pattern Recognition*, pp.6517-6525, 2016.
 17. Joseph Redmon, Ali Farhadi, "YOLO v3: An Incremental Improvement", *Computer Vision and Pattern Recognition*, 2018. Available at <https://pjreddie.com/media/files/papers/YOLOv3.pdf> [Accessed Mar.13, 2020].
 18. Qi-Chao Mao, Hong-Mei Sun, Yan-Bo Liu, Rui-Sheng Jia, "Mini-YOLOv3: Real-time Object Detector for Embedded Applications", *IEEE Access*, vol.7, pp.133529-133538, 2019.
 19. Adrian Carrio, Sai Vemprala, Andres Ripoll, Srikanth Saripalli, Pascual Campoy, "Drone Detection using Depth Maps", *International Conference on Intelligent Robots and Systems*, pp.1032-1037, 2018.
 20. University of Oxford, "Pascal Visual Object Classes Homepage", Available at <http://host.robots.ox.ac.uk/pascal/VOC/index.html> [Accessed: Mar.13, 2020].
 21. Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, "Speed/Accuracy Trade-offs for Modern Convolutional Object Detectors", *IEEE Conference on Computer Vision and Pattern Recognition*, pp.7310-7319, 2017.
 22. Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, Deva Ramanan, "Object Detection with Discriminatively Trained Part-based Models", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.32, issue.9, 2010.
 23. Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, Stefan Carlsson, "CNN Features Off-the-Shelf: An Astounding Baseline for Recognition", *IEEE Conference on Computer Vision and Pattern Recognition*, pp.512-519, 2014.
 24. Ross Girshick, "Fast R-CNN", *Computer Vision and Pattern Recognition*, pp.1440-1448, 2015.
 25. Hyochang Ahn, Yong-Hwan Lee, "A Research of CNN-based Object Detection for Multiple Object Tracking in Image", *Journal of the Semiconductor and Display Technology*, vol.18, no.3, 2019.
 26. Yong-Hwan Lee, Youngseop Kim, "Implementation of Object Feature Extraction within Image for Object

-
- Tracking”, Journal of the Semiconductor and Display Technology, vol.17, no.3, 2018.
27. Rahul Haridas, Jyothi RL, “Convolutional Neural Networks: A Comprehensive Survey”, International Journal of Applied Engineering Research, vol.14, no.3, pp.780-789, 2019.
 28. Akshi Kumar, Sukriti Verma, Himanshu Mangla, “A Survey of Deep Learning Techniques in Speed Recognition”, International Conference on Advances in Computing, Communication Control and Networking, pp.179-185, 2018.
 29. M. Sornam, Kavitha Muthusubash, V. Vanitha, “A Survey on Image Classification and Activity Recognition using Deep Convolutional Neural Network Architecture”, International Conference on Advanced Computing, pp.121-126, 2017.
 30. Azeddine Elhasouny, Florentin Smarandache, “Trends in deep convolutional neural networks architectures: a review”, International Conference of Computer Science and Renewable Energies, pp.1-8, 2019.
-
- 접수일: 2020년 3월 15일, 심사일: 2020년 3월 19일,
 게재확정일: 2020년 3월 23일