

객체탐지모델 YOLO의 버전별 특성 비교 연구

김준용^o

^o서울신학대학교 IT융합소프트웨어학과

e-mail: musimk@stu.ac.kr^o

A comparative study on the characteristics of each version of object detection model YOLO

Joon-Yong Kim^o

^oDept. of IT Convergence Softwaer, Seoul Theological University

● 요약 ●

본 논문은 객체탐지 모델 중 주류를 이루고 있는 YOLO의 v1부터 v8까지의 특성을 비교 분석하여 각각의 버전에 최적화할 수 있는 모델에 대한 연구이다. 연구 결과 v1, v2는 정확성이 최우선인 모델에 적합하다. 반면, v3, v4는 속도가 우선인 모델에 적합하다. 또한 v5, v6는 정확도와 속도 사이의 균형이 필요한 모델에 적합하다는 결론을 얻었다. v7, v8은 메모리 및 컴퓨팅 성능에 제약이 있는 경우 주로 적용이 가능하며, 적은 연산과 저 메모리 사용으로 객체를 탐지하여 포즈추정이나 객체 추적 등을 적용할 모델에 적합하다는 결과를 확인하였다.

키워드: 객체탐지(Object Detection), 객체분할(Object Segmentation), 객체분류(Object Classification), YOLO

I. Introduction

컴퓨팅 자원의 발전과 더불어 객체탐지에 대한 니즈가 지속적으로 대두되었으며 이를 위한 다양한 기법들이 제안되었다. 그 중 대표적으로 HOG와 DPM 등 전통적인 객체검출 기법과 AlexNet을 기점으로 딥러닝 기반의 CNN을 고도화한 Two stage detector 방식과 YOLO 계열의 One stage detector 방식이 주류를 이루고 있다. 이 중 YOLO는 다양한 버전을 통해 객체탐지의 성능을 지속적으로 개선하여 객체탐지 모델의 주류가 되어가고 있다.

특히 2020년 발표된 v5를 중심으로 활발한 연구와 적용사례가 있었으나, 이후 발표된 v6, v7에 대한 연구는 사례가 많지 않으며, 2023년 1월 발표된 v8에 대해서도 리뷰논문외에는 연구사례를 찾아 보기 힘든 실정이다.

이에 본 논문에서는 v1부터 최근 v8까지의 YOLO 각 버전별 특성을 비교분석하여 특정분야에 최적화할 수 있는 YOLO 버전별 최적모델을 분석하였다.

분석결과 객체크기, 속도와 정확도 문제를 해결한 v4와 경량화와 AutoML 기능을 추가한 v5 모델이 주류를 이루고 있음을 확인하였다. 또한 이후 v7과 v8을 통해 포즈추정과 객체 추적, 비전 AI까지도 영역이 확장되었음을 확인하였다.

이를 바탕으로 각 버전에 따른 모델을 제시할 수 있는 근거를 마련하였다.

v1, v2는 정확성이 최우선인 모델에 적합하며, v3, v4는 속도가 우선인 모델에 적합하다. 또한, v5, v6는 정확도와 속도 사이의 균형이 필요한 모델에 적합하다는 결론을 얻었다.

v7, v8은 메모리 및 컴퓨팅 성능에 제약이 있는 경우 주로 적용 가능하며, 적은 연산과 저 메모리 사용으로 객체를 탐지하여 포즈추정이나 객체 추적 등을 적용할 모델에 적합함을 확인하였다.

II. Preliminaries

1. Related works

1.1 객체탐지 모델(Object Detection Model)

객체 탐지모델은 2012년 ILSVRC 대회에서 처음 대두된 AlexNet을 기점으로 전통적인 탐지모델과 딥러닝 기반의 현대적 모델로 구분할 수 있다.

AlexNet 이전의 대표적인 모델로는 HOG와 DPM을 들 수 있다.

1.1.1. HOG(Histogram of Oriented Gradient for human detecting)

2005년 보행자 검출을 목적으로 만들어졌으며, SVM 알고리즘을 이용하여 객체를 검출하고, 이미지에서 Gradient 개념을 이용하여 Feature Vector를 정의하는 기법이다[1].

1.1.2. DPM(Deformable Part Model)

2008년 HOG의 확장버전으로 발표되었으며, 최초로 바운딩 박스(Bounding Box) 개념을 도입하였다. 미리 만들어진 이미지의 각 부위별 템플릿 필터와 바운딩 박스 영역을 매칭하여 일치 결과에 의해 객체를 검출하는 방식이다[2].

이러한 슬라이딩 윈도우 방식은 다양한 바운딩 박스가 발생하므로 이를 분류하는데 수 많은 시간이 소요되는 문제를 발생시켰다.

1.1.3. AlexNet

AlexNet은 최초로 딥러닝 기반의 CNN을 이용하여 객체를 검출하려는 방식이며[3], 발표 이후 CNN을 더욱 빠르게 고도화시킨 Two stage detector 방식과 One stage detector 방식의 YOLO로 구분되어 발전되었다.

1.2 YOLO(You Only Look Once)

One stage detector 방식의 YOLO는 영상에서 동일한 이름의 객체를 컨볼루션층의 그리드 셀별로 매칭이 되는 확률을 계산하여 분류와 위치를 확인하여 객체를 탐지하는 기법이다[4].

다음의 그림 1은 이러한 One stage detector 대한 아키텍처이다.

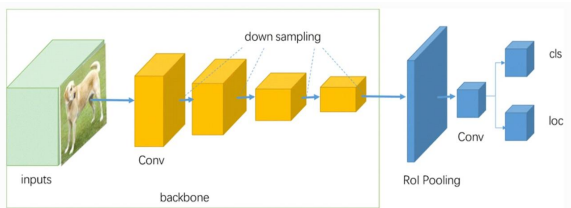


Fig. 1. architecture of One stage object detector

이러한 방식은 Two stage detector 방식보다 속도면에서 우수한 성능을 보인다[5].

특히 YOLO의 큰 장점 중 하나인 회귀 문제에 대한 연산을 한번에 가능케 한 것이다. YOLO는 사람이 어떤 물체를 봤을 때 한번의 시선으로도 물체를 인식하는 것과 유사한 형태로 만들어져 있다[4].

III. Research and Analysis

One stage detector 방식의 YOLO는 2016년 v1을 시작으로 2023년 현재 v8까지 지속적인 성능개선 모델이 발표되고 있다.

HOG와 DPM등 전통적인 객체검출 기법에서 AlexNet을 기점으로

CNN을 고도화한 Two stage detector와 YOLO계열의 One stage detector 방식으로 전환된 객체탐지 기법은 최근 YOLO 버전의 비약적인 성능향상으로 이에 대한 다양한 연구와 적용이 이루어지고 있다.

특히 2020년 발표된 v5를 중심으로 활발한 연구와 적용사례가 있으나 이후 발표된 v6, v7에 대한 연구는 사례가 많지 않으며, 2023년 초 발표된 v8에 대해서도 연구사례를 찾아보기 힘든 실정이다.

이에 본 논문에서는 최근 v8까지의 YOLO 각 버전별 특성을 비교분석하여 특정분야에 최적화할 수 있는 YOLO버전별 모델을 분석하였다.

다음의 표 1은 YOLO의 각 버전별 특성을 비교한 것이다.

Table 1. Characteristics of YOLO Models

Ver	Lunch	Feature
1	2016	- CNN기반, Feature Map 추출 - Bounding Box 적용 - Darknet-19 모델 적용 - 단일 크기의 객체감지
2	2017	- v1의 단점 보완 - Batch Normalization 기법 도입 - Darknet-53 모델 적용 - 다양한 크기의 객체감지 - 일부 객체 인식 시 문제 발생
3	2018	- FPN적용, 작은객체 감지성능 개선 - 3개의 Bounding Box 적용 - 위치개선을 위해 SPP-back 적용 - GPU이용하여 처리속도 개선(실시간 객체탐지 가능)
4	2020	- v3의 성능과 처리속도 개선(CSPNet 적용) - 작은객체 감지 개선(PANet 적용) - GPU를 이용한 v3에 정확도를 높임 - 데이터 증강모델과 모델규제를 위한 Mosaic Argument 기법 적용
5	2020	- Ultralytics에서 PyTorch 기반으로 발표 - v4대비 속도와 정확도 2~4배 개선 - 경량화모델 지원(모바일, 엣지) - CPU, GPU 모두 지원 - AutoML기능 적용
6	2022	- v5의 속도와 정확도 증대 - 포즈 추정기능 추가 - PAN을 강화한 RepPAN 적용
7	2022	- v4를 기반으로 확장 - 실시간 객체 검출을 위한 확장 및 스케일링 제안
8	2023	- 최첨단 SOTA모델 적용 - 감지, 분할, 포즈추정, 추적기능 고도화 - 비전 AI작업 지원

위의 표 1에서 살펴본 바, v3부터 실시간 객체 감지 기능이 추가되었고, v5에서는 AutoML이 가능하며 최신 모델인 v8은 객체 추적 및 비전 AI작업을 가능케하는 모듈이 추가되었다.

이외에도 2003년 발표된 v8보다 정확도와 속도를 향상시킨 v6.3.0과 Deci의 독립기술인 AutoNAC를 고도화시킨 YOLO-NAS가 있으나, 기술적 이슈를 중점으로 연구하기 위하여 v8까지의 모델만을 연구범위에 국한하였다.

다음의 그림 2는 v5에서부터 모델을 발표한 Ultralytics에서 각 모델별 정확도와 실시간 프레임처리속도를 비교 분석한 것이다[6].

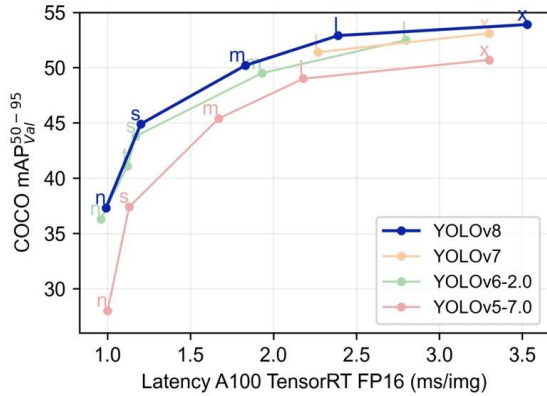


Fig. 2. Comparison of accuracy and speed by model

검증을 위해 COCO 데이터셋을 이용하였고, 각 모델의 정확도와 실시간 프레임처리 속도를 비교한 결과 실시간처리 신뢰 값인 3.0(FP16)을 기준으로 53(mAP)을 나타낸 v8이 다른 모델보다 좋은 성능을 보임을 입증하였다.

다음의 그림 3은 v5와 v8의 객체 크기에 따른 탐지, 분할과 분류성능을 비교한 것이다[7].

Performance Comparison of YOLOv8 vs YOLOv5

Model Size	Detection*	Segmentation*	Classification*
Nano	+33.21%	+32.97%	+3.10%
Small	+20.05%	+18.62%	+1.12%
Medium	+10.57%	+10.89%	+0.66%
Large	+7.96%	+6.73%	0.00%
Xtra Large	+6.31%	+5.33%	-0.76%

*Image Size = 640 *Image Size = 224

Fig. 3. performance comparison by object size

위의 그림 2의 결과 거의 모든 객체크기에서 v8의 성능이 우수함을 보였으며, 특히 나노크기의 객체에서 큰 성과를 보임을 알 수 있다.

YOLO의 버전별 특성을 비교한 표 1, 그림 1, 그림 2의 결과를 분석한 결과는 다음과 같다.

- v1, v2는 정확성이 최우선인 모델에 적합하다.
- v3, v4는 속도가 우선인 모델에 적합하다.
- v5, v6는 정확도와 속도 사이의 균형이 필요한 모델에 적합하다. 즉 v1, v2보다 처리속도는 빠르며 정확도 또한 우수한 결과가 필요한 모델에 적합하다. 또한, 실시간 객체 탐지가 가능한 모델에 적합하다.
- v7, v8은 메모리 및 컴퓨팅 성능에 제약이 있는 경우 주로 적용가능하며, 적은 연산과 저 메모리 사용으로 객체를 탐지하여

포즈추정이나 객체 추적 등을 적용할 모델에 적합하다.

IV. Conclusions

본 논문은 객체탐지 모델 중 주류를 이루고 있는 YOLO의 각 버전별 특성을 비교연구하여 각 버전에 적합한 최적의 모델에 대한 연구를 진행하였다.

연구결과 v1, v2는 정확성이 최우선인 모델에 적합함을 확인하였다. v3, v4는 속도가 우선인 모델에 적합하며, v5, v6는 정확도와 속도 사이의 균형이 필요한 모델에 적합하다. 즉 v1, v2보다 처리속도는 빠르며 정확도 또한 우수한 결과가 필요한 모델에 적합하다. 또한, 실시간 객체 탐지가 가능한 모델에 적합하다.

v7, v8은 메모리 및 컴퓨팅 성능에 제약이 있는 경우 주로 적용가능하며, 적은 연산과 저 메모리 사용으로 객체를 탐지하여 포즈 추정이나 객체 추적 등을 적용할 모델에 적합하다. 특히 v8의 객체추적에 대한 구체적인 적용사례와 성능 등 이에 대해서는 적극적인 향후 연구가 필요하다.

ACKNOWLEDGEMENT

This study was supported by the Research Program funded by the Seoul Theological University.

REFERENCES

- [1] Sultana, M., Ahmed, T., Chakraborty, P., Khatun, M., Hasan, M. R., & Uddin, M. S., "Object detection using template and HOG feature matching", International Journal of Advanced Computer Science and Applications, 11(7), pp233-238, 2020.
- [2] Ali, Alaa, Oladiran G. Olaleye, and Magdy Bayoumi. "Fast region-based DPM object detection for autonomous vehicles." 2016 IEEE 59th International Midwest Symposium on Circuits and Systems (MWSCAS). IEEE, pp1-4, 2016.
- [3] Arya, Sunayana, and Rajeev Singh, "A Comparative Study of CNN and AlexNet for Detection of Disease in Potato and Mango leaf", 2019 International conference on issues and challenges in intelligent computing techniques (ICICT), Vol. 1. IEEE, pp1-6, September, 2019.
- [4] Ik-Su Kim, Moon Gu Lee, Yongho Jeon, "Comparative Analysis of Defect Detection Using YOLO of Deep Learning". Journal of the Korean Society of Manufacturing

Technology Engineers, 30(6), pp514-519, 2021.

- [5] Soviany, P., Ionescu, R. T., 2018, "Optimizing the Trade-off between Single-Stage and Two-stage Deep Object Detectors using Image Difficulty Prediction", 20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), pp209-214, 2018.
- [6] Comparison of accuracy and speed by model, <https://learnopencv.com/ultralytics-yolov8/>
- [7] Performance comparison by object size, <https://learnopencv.com/ultralytics-yolov8/>