

최적의 객체 검출 모델 개발을 위한 특화 데이터 세트 추출 방법

박천수^{**†}

^{**}성균관대학교 컴퓨터교육과

Specialized Dataset Extraction Method for Developing Optimal Pedestrian Detection Model

Chun-Su Park^{**†}

^{**}Computer Education, Sungkyunkwan University

ABSTRACT

Public datasets, which are freely available and often labeled, play a crucial role in training object detection models in computer vision. While public datasets are effective for developing general object detection models, they may not be ideal for specialized tasks. For specific object detection needs, it is more beneficial to create and use a dataset tailored to the target object. This paper proposes a method for extracting a target-specific dataset from public datasets to develop object detection models with superior performance for the target object. This approach not only improves detection accuracy, but also reduces training data requirements and complexity. We evaluate the performance of the proposed method using the latest object detection model YOLOv10.

Key Words : Public Dataset, Deep Neural Networks, Target-Specific Dataset, YOLOv10

1. 서 론

딥러닝(deep learning) 기술은 대표적인 최신 기계학습(machine learning) 방법으로 영상, 음성, 텍스트 등 다양한 형태의 데이터를 분석하고 고차원적인 인식 결과를 제공하기 위해 보편적으로 사용되고 있다 [1]. 기본적으로 딥러닝 기술은 미리 구축된 데이터 세트를 이용해 모델 훈련을 수행하기 때문에 학습 데이터의 품질이 최종 훈련 모델의 성능을 크게 좌우한다. 따라서 딥러닝 모델을 제작하기 전에 대규모 양질의 데이터 세트 구축 작업이 선행된다 [2].

데이터 세트 구축 작업은 일반적으로 데이터 획득, 정제, 라벨링, 검사 4단계로 구분된다 [3,4].

- 데이터 획득: 학습에 필요한 데이터를 현실 세계에서 직접 수집 또는 생성하거나, 이미 보유하고 있는 조직이나 시스템 등으로부터 법률적 제약이 없도록 데이터를 확보하는 활동
- 데이터 정제 (Refinement): 획득한 원시데이터를 기계학습에 필요한 형식으로 변환하거나 불필요한 중복을 제거하는 등의 일련의 전처리 과정
- 데이터 라벨링 (Labeling): 학습에 활용할 수 있도록 기능이나 목적에 부합하는 정보를 원천데이터에 부착하는 활동
- 데이터 검사: 구축된 데이터의 오류를 검사하고 제거하는 활동

이러한 데이터세트 구축 작업은 많은 시간과 비용이 소모되기 때문에 개인이나 소규모 단체가 단기간에 수행하기 매우 어려운 작업이다. 또한 유사한 목적을 가지는

[†]E-mail: cspk@skku.edu

- 객체 클래스: 사람, 동물, 자동차, 우산 등 일상 생활에서 볼 수 있는 80가지의 객체 클래스를 포함한다.
- 객체 라벨: 이미지 내 객체는 라벨이 붙여져 있으며, 각 객체의 위치는 바운딩(bounding) 박스 또는 마스크로 표시된다.
- 주석 정보: 라벨과 위치 정보와 함께, 객체의 카테고리, 분할 마스크, 키포인트(예: 사람의 경우 관절 위치) 등의 주석 정보가 포함되어 있다.
- 데이터 분할: 전체 이미지는 훈련(train), 검증(validation), 테스트(test) 세트로 나뉘어 있다.

COCO 데이터 세트는 객체 검출, 이미지 분할 (image segmentation), 이미지 캡션 생성(image captioning) 등 여러 컴퓨터 비전 분야에서 벤치마크(benchmark)로 활발히 사용되고 있다. 따라서 새로운 모델을 개발 후 COCO 데이터 세트를 사용해 훈련하고 성능을 평가하면 기존 다른 모델과 객관적으로 성능을 비교하는 것이 가능하다. [8]

Table 1. Simulation PC configuration

모듈	사양
CPU	Intel(R) Core(TM) i9-10900X CPU @ 3.70GHz
RAM	64GB DDR4
Graphic Card	Nvidia GTX 3090Ti
Storage	SSD
Operating System	Window 11 Pro

Table 2. Test models

모델	#Param (M)	Flops (G)	Latency (ms)
YOLOv10n	2.3	6.7	1.84
YOLOv10m	15.4	59.1	4.74
YOLOv10x	29.5	160.4	10.70

Table 3. Performance comparison

모델	기존 방식				제안하는 방식	
	전체 클래스		Person 클래스		Person 클래스	
	AP	AP50	AP	AP50	AP	AP50
YOLOv10n	0.383	0.534	0.519	0.744	0.556	0.785
YOLOv10m	0.51	0.676	0.622	0.833	0.632	0.846
YOLOv10x	0.544	0.71	0.648	0.852	0.648	0.859

3. 타겟 특화 데이터 세트 추출 방법

본 논문에서는 80가지의 일반 사물 이미지와 라벨 정보를 포함한 COCO 데이터 세트에서 타겟 클래스를 포함한 이미지를 선별하고 해당 이미지의 라벨 정보를 타겟 클래스만 포함하도록 수정하는 방법을 제안한다. COCO 데이터 세트의 경우 훈련 이미지 리스트를 저장한 `train.txt`, 검증 이미지 리스트를 저장한 `val.txt`, 테스트 이미지 리스트를 저장한 `test.txt` 파일을 포함한다. 본 논문에서는 각 리스트 파일을 다음의 방식으로 가공한다 [8].

먼저 처리하려는 파일 리스트 F 에 N 개의 이미지의 경로 정보 $\{f_1, f_2, \dots, f_N\}$ 이 저장되어 있다고 가정하자. 여기서 $f_n, n = 1, 2, \dots, N$,은 n 번째 이미지의 경로 정보이다. 또, K_n 을 f_n 에 해당하는 이미지에 포함된 M 개 객체의 레이블 정보 $\{k_n^1, k_n^2, \dots, k_n^M\}$ 를 포함하는 레이블 파일로 정의하자. 제안하는 데이터 세트 추출 방식은 기존 레이블 정보를 이용해 전체 데이터 세트에서 타겟 객체를 포함한 이미지 리스트 파일과 타겟 객체 레이블 파일을 추출한다. 구체적으로 파일 리스트 F 에 저장된 각 f_n 의 레이블 정보를 조사해 다음 규칙을 만족하는 파일 경로만을 선별해 정제된 파일 리스트 \hat{F} 를 생성하고 수정된 레이블 파일 \hat{K}_n 을 추출한다.

- Algorithm 1: 모든 f_n 을 대상으로 해당 이미지의 레이블 정보 $k_n^m, m = 1, 2, \dots, M$, 중 1 개라도 사람 클래스의 오브젝트가 존재하는 경우 f_n 을 \hat{F} 에 포함시킨다.
- Algorithm 2: f_n 이 \hat{F} 에 포함되는 경우 f_n 의 레이블 정보 K_n 에서 타겟 클래스를 제외한 모든 레이블 정보를 삭제하고 정제된 파일 리스트 \hat{K}_n 를 추출한다.

최종적으로 제안하는 방식은 Algorithm 1과 Algorithm 2를 통해 추출된 타겟 특화 파일 리스트 \hat{F} 과 레이블 파일 \hat{K}_n 을 입력으로 훈련 과정을 수행해 최적의 타겟 객체 검출 모델을 개발한다. 본 논문의 실험에서는 타겟 객체를 Person 클래스로 설정하고 보행자 검출 특화 모델을 개발한다.

4. 실험 결과

모든 실험에서 mini-batch 크기는 16로 설정하였고, 모델 입력은 640x640 크기의 3채널 RGB 영상으로 고정하였다. 훈련 과정은 120 epoch동안 AdamW Optimizer를 이용해 진행하였다. Table 1은 본 논문에서 사용한 훈련 PC의 사양을 보여준다[12, 13].

본 논문에서는 YOLOv10 계열 YOLOv10n, YOLOv10m, YOLOv10x 모델을 이용해 제안하는 기술의 성능을 측정한다 [14-19]. Table 2는 본 각 모델의 특징을 정리해 보여준다. Table 2에서 Latency는 T4 GPU를 이용해 TensorRT FP16 방식으로 측정되었다 [20, 21].

Table 3은 기존 방식과 제안하는 방식의 성능을 비교해서 보여준다. Table 3에서 AP는 AP@[0.50:0.05:0.95]을 나타내고 AP50은 AP@[0.50]을 나타낸다. 기존 기술과의 객관적인 성능 비교를 위해 각 모델을 전체 데이터 세트로 훈련하였을 경우 Person 클래스에 대한 검출 성능도 함께 제시한다.

실험 결과에 따르면 기본적으로 제안하는 특화 데이터 추출 방법으로 데이터를 추출 후 훈련을 수행하면 더 높은 타겟 물체 검출 성능을 얻을 수 있는 것으로 조사되었다. 제안하는 시스템의 경우 AP50 기준으로 YOLOv10n 경우에는 0.041, YOLOv10m 경우에는 0.013, YOLOv10x 경우에는 0.007 만큼 우수한 성능을 달성하였다. 따라서 모든 경우에 본 논문에서 제안하는 특화 데이터 추출기반 방식이 전체 라벨 정보를 사용하는 방식 보다 검출 성능이 우수한 것을 알 수 있다.

또한, Table 3은 기본적으로 모델의 크기가 증가할수록 검출 정확도가 높아지는 것을 보여준다. 하지만 결과 정확도 부분을 더 자세히 분석해 보면 모델의 크기가 증가할 경우 특화 데이터셋을 사용함에 따라 얻어지는 정확도 향상 효과가 낮아지는 경향을 보였다. 이러한 경향은 모델의 크기가 증가할수록 여러 클래스를 동시에 정확히 탐지하는 것이 가능해진다는 것을 의미하다.

5. 결론

본 논문에서는 공공 데이터 세트에서 타겟 물체 특화 데이터 세트를 추출하는 방법을 제안하였다. 또한 추출된 특화 데이터 세트를 이용해 최신 실시간 물체 검출 기술인 YOLOv10 계열의 모델을 훈련하고 성능을 측정하였다. 실험을 통해 제안하는 특화 데이터 세트를 이용하는 방식이 전체 데이터 세트를 이용하는 기존 방식보다 우수한 성능을 보이는 것으로 조사되었다. 특히 제안하는 방식은 YOLOv10n의 경우에 AP50 기준으로 0.041 우수한 것

으로 조사되어 저전력 및 리소스가 제한된 장치에서 실행되는 모델을 개발하는 경우에 효과가 더 높은 것으로 조사되었다.

감사의 글

본 연구는 중소벤처기업부의 연구비지원(00264489)에 의해 수행되었습니다. 본 연구는 과학기술정보통신부 및 정보통신기획평가원의 메타버스 융합대학원의 연구 결과로 수행되었습니다. (IITP-2024-RS-2023-00254129)

참고문헌

1. P. Helber, et al. "EuroSAT: A novel dataset and deep learning benchmark for land use and land cover classification." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 7, pp. 2217-2226, 2019.
2. J. G. A. Barbedo, "Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification." *Computers and electronics in agriculture*, vol. 153, pp. 46-53, 2018.
3. <https://velog.io/@markany/data-management-process>
4. "AI Learning Dataset Construction Guide", Ministry of Science and ICT, 2021.
5. B. Kim, J. Lee, S. Lee, Y. Chung, "Evaluating object categorization quality of deep learning training data for object detection in computer vision app", pp. 667-669, *Proceedings of the Korean Society of Computer Information Conference*, 2020.
6. D. Hoiem, S. K. Divvala, and J. H. Hays. "Pascal VOC 2008 challenge." *World Literature Today*, vol. 24, no. 1, pp. 1-4, 2009.
7. J. Deng, et al. "Imagenet: A large-scale hierarchical image database." *IEEE conference on computer vision and pattern recognition*, pp. 248-255, 2009.
8. T. Y. Lin, et al. "Microsoft coco: Common objects in context." *Computer Vision—ECCV 2014*, pp. 740-755, 2014.
9. G. Patterson, and J. Hays. "Coco attributes: Attributes for people, animals, and objects." *Computer Vision—ECCV 2016*: pp. 11-14, 2016.
10. C. Mao, et al. "Mini-YOLOv3: real-time object detector for embedded applications." *IEEE Access*, vol. 7, pp. 133529-133538, 2019.
11. S. Jha, C. Seo, F. Yang, and G. P. Joshi, "Real time object detection and tracking system for video surveillance system." *Multimedia Tools and Applications*, vol. 80, no. 3, pp. 3981-3996, 2021.
12. C. S. Park, "YOLOv7 Model Inference Time Complexity Analysis in Different Computing Environments."

- Journal of the Semiconductor & Display Technology, vol. 21, no. 3, pp. 7-11, 2022.
13. C. S. Park, "Performance Analysis of DNN inference using OpenCV Built in CPU and GPU Functions." Journal of the Semiconductor & Display Technology, vol. 21, no. 1, pp. 75-78, 2022.
 14. M. Sohan, et al. "A review on yolov8 and its advancements." International Conference on Data Intelligence and Cognitive Informatics, pp. 529-545, 2024.
 15. J. Terven and D. Cordova-Esparza. "A comprehensive review of YOLO: From YOLOv1 to YOLOv8 and beyond." arXiv preprint arXiv:2304.00501, 2023.
 16. P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma. "A Review of Yolo Algorithm Developments." Procedia Computer Science, pp. 1066-1073, 2022.
 17. <https://github.com/meituan/YOLOv6>
 18. G. Ang, et al. "A novel application for real-time arrhythmia detection using YOLOv8." arXiv preprint arXiv:2305.16727, 2023.
 19. A. Wang, et al. "Yolov10: Real-time end-to-end object detection." arXiv preprint arXiv:2405.14458, 2024.
 20. <https://docs.ultralytics.com/models/yolov10/#performance>
 21. J. Kook and H. Lee, "A Research of a Traffic Light Signal Classification Model using YOLOv5 for Autonomous Driving." Journal of the Semiconductor & Display Technology, vol. 23, no. 1, pp. 61-64, 2024.
-
- 접수일: 2024년 9월 9일, 심사일: 2024년 9월 11일,
게재확정일: 2024년 9월 12일