COSE474-2024F: Final Project
"YOLO와 CLIP을 결합한 텍스트 기반 객체 탐지 및 분류 시스템의 가능성과 한계"

Daehyun Kim

Abstract

객체 탐지와 분류는 자율주행, 감시 시스템 등 다양한 컴퓨터 비전 응용에서 중요한 역할을 한다. 본 연구는 YOLO의 실시간 객체 탐지 기능과 CLIP의 텍스트 기반 제로샷 분류 기능을 결합한 새로운 파이프라인을 제안한다. 이를 통해 고정된 클래스 라벨의 제약을 극복하고, 텍스트 프롬프트를 기반으로 유연하고 확장 가능한 객체 탐지 및 분류 시스템을 구현하였다.

COCO 데이터셋을 활용한 실험 결과, YOLO와 CLIP의 결합이 정량적으로 기존 모델 대비 성능을 개선하였으나, 정성적 평가에서 텍스트 매칭의 한계가 관찰되었다. 본 연구는 이를 기반으로 향후 SAM과 같은 최신 분할 모델과의 통합을 통해 더 정교한 탐지 및 분류 시스템을 설계할 가능성을 제시한다.

1. Introduction

1.1. Motivation

YOLO는 실시간 객체 탐지에서 강력한 성능을 발휘하지만, 고정된 클래스 라벨에 의존하여 새로운 클래스나 상황에 유연하게 대응하기 어렵다. 반면, CLIP은 텍스트 프롬프트를 활용한 제로샷 분류가 가능하지만, 객체의 위치 탐지 기능이 부족하다. 본 연구는 YOLO와 CLIP을 결합하여 두 기술의 강점을 통합한 시스템을 설계하고, 객체 탐지와 텍스트 기반 분류를 동시에 수행할 수 있는 환경을 제공하고자 한다. 이를 통해 데이터 라벨링 자동화, 유연한 검색 시스템, 실시간 모니터링 등 다양한 응용 분야에 기여할 잠재력을 지닌다.

1.2. Problem Definition

기존 객체 탐지 및 분류 기술은 복잡하고 변화가 많은 실제 환경에서 유연성과 확장성에 제약을 받고 있다. YOLO는 높은 탐지 정확도를 제공하지만 고정된 클래스 구조로 인해 새로운 조건에 대응하기 어렵고, CLIP은 텍스트 기반으로 유연한 분류가 가능하나 객체 위치 탐지나 세분화와 같은 기능이 부족하다. 이러한 한계는 실제 응용에서의 활용성을 저해한다.

본 연구는 이러한 한계를 해결하기 위해 YOLO와 CLIP을 결합한 새로운 파이프라인을 제안한다. 구체적으로, YOLO를 통해 객체의 위치를 탐지한 후 CLIP을 활용하여 텍스트 프롬프트 기반으로 객체를 분류하는 방식을 설계한다. 이를 통해 "a blue cup"과 같은 특정 텍스트 프롬프트에 부합하는 객체를 탐지하고 분류할 수 있는 유연한 접근 방식을 구현하며, COCO 데이터셋을 활용하여 시스템의 성능과 두 기술의 결합이 가지는 가능성과 한계를 평가한다.

1.3. Concise Description of Contribution

본 연구는 YOLO와 CLIP의 강점을 결합하여 텍스트 기반 객체 탐지 및 분류를 위한 새로운 통합 시스템을 설계하였다. YOLO의 실시간 객체 탐지 능력과 CLIP의 텍스트 프롬프트 기반 제로샷 분류 기능을 활용하여 고정된 클래스 구조의 한계를 극복하였으며, COCO 데이터셋을 통해 성능을 평가하였다. 또한, 두 기술의 결합 과정에서 드러난 한계를 분석하여 향후 연구 방향에 대한 통찰을 제공한다.

2. Methods

2.1. Significance/Novelty

본 연구는 YOLO와 CLIP의 결합을 통해 객체 탐지 및 텍스트 기반 분류 문제를 동시에 해결할 수 있는 새로운 시스템을 설계하였다. 기존의 객체 탐지 모델은 고정된 클래스 라벨에 의존하여 새로운 클래스나 변형된 상황에 대해 유연하게 대응할 수 없었다. 반면, CLIP은 제로샷 분류를 가능하게 하지만, 객체의 위치 탐지 기능이 부족하다. 본 연구는 이러한 두 기술의 결합을 통해 고정된 클래스 내에서 더 세밀하고 유연한 분류를 가능하게 하고, 동적 데이터 라벨링 및 유연한 객체 탐지 시스템을 구현하였다. 이를 통해 실시간 모니터링과 데이터 라벨링 자동화 등 여러 응용 분야에서 실질적인 변화를 가져올 수 있다.

2.2. Algorithm

본 연구에서 제안하는 모델의 전체적인 파이프라인은 Figure 1에 나타낸 바와 같으며, 아래와 같이 설명된다.

1. 객체 탐지(YOLO): 입력 이미지는 먼저 YOLO 모델에 전달된다. YOLO는 실시간 객체 탐지에 특화된 모델로, 이미지 내에 존재하는 다양한 객체들을 탐지하고 각 객체에 대한 bounding box 정보를 반환한다. 이 단계에서는 각 객체의 위치와 클래스가 함께 탐지된다. YOLO 모델은 사전 학습된 COCO 데이터셋을 기반으로 훈련되어 있으며,

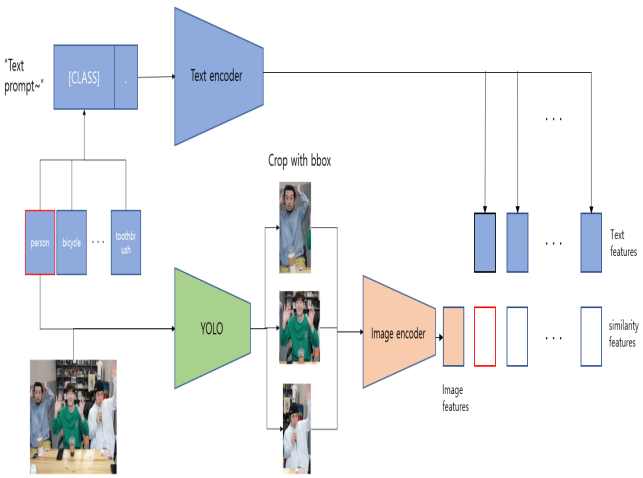

Figure 1. YOLO and CLIP for Object Detection and Text-Based Classification

이미지에서 특정 객체를 정확하게 식별할 수 있도록 최적화되어 있다. 이 과정에서 각 객체에 대한 bounding box 좌표가 추출된다.

2. 텍스트 기반 분류(CLIP): YOLO가 탐지한 객체들에 대해서는 해당 객체들의 bounding box 영역을 잘라내어 CLIP 모델에 입력한다. CLIP 모델은 텍스트와 이미지를 결합하여 학습된 모델로, 주어진 텍스트 프롬프트와 이미지 간의 유사도를 계산하는 데 사용된다. 각 객체 이미지가 CLIP 모델을 통해 처리되며, 이때 주어진 텍스트 프롬프트와의 유사도가 계산된다. CLIP은 제로샷 학습을 지원하기 때문에, 사전 학습되지 않은 텍스트 프롬프트에 대해서도 분류를 수행할 수 있다.

3. 결과 산출: YOLO와 CLIP의 출력을 통합하여 최종 결과를 산출한다. YOLO는 객체의 위치를 정확하게 파악하고, CLIP은 텍스트 프롬프트와의 유사도를 계산한다. 이 두 정보를 결합하여, 유사도가 높은 객체들만을 필터링하고, 해당 객체들이 주어진 텍스트 프롬프트와 일치하는지 여부를 판단한다. 예를 들어, "a blue cup"이라는 텍스트 프롬프트에 대해, 모델은 파란색 컵을 탐지하고, 그 유사도 점수가 일정 임계값 이상인 경우에만 해당 객체를 최종적으로 선택한다.

이러한 알고리즘을 통해, 기존의 YOLO와 CLIP 모델이 각기 다른 역할을 수행하면서도 유기적으로 결합될 수 있다.

2.3. reproducibility

본 프로젝트에 사용된 모델 및 버전은 아래와 같다.

1. YOLOv8: 객체 탐지 모델로 YOLOv8 Nano를 사용하였으며, 버전은 8.3.48의 Ultralytics 패키지를 사용하였다.

2. CLIP: 텍스트-이미지 매칭 및 제로샷 분류를 위해 OpenAI의 CLIP 모델을 사용하였으며, 사용된 CLIP 버전은 1.0이다.

3. Experiments

3.1. Datasets

본 연구에서는 모델 평가를 위하여 COCO validation 데이터셋(val2017)을 사용하였다. COCO 데이터셋은 다양한 객체 카테고리와 현실적인 이미지들이 포함되어 있어, 객체 탐지와 텍스트 기반 분류의 성능을 평가하기에 적합하다. 또한 COCO 데이터셋은 객체 탐지 및 분류 작업을 위한 표준 벤치마크로 널리 사용되므로, 기존 연구들과의 비교가 용이하다는 장점이 있다. 따라서 COCO validation 데이터셋을 사용하여 모델 평가를 진행했다.

3.2. Computer resources experimental design

본 실험은 Google Colab 환경에서 진행되었으며, 아래와 같은 사양을 사용했다.

- **플랫폼**: Google Colab (Google Compute Engine 백엔드, GPU 사용)
- **하드웨어 사양**:
  - **RAM**: 1.90 GB / 12.67 GB
  - **디스크**: 34.39 GB / 112.64 GB
  - **GPU**: NVIDIA Tesla T4
- **Python 버전**: Python 3.10
- **사용된 라이브러리**:
  - **PyTorch**: 2.5.1 (CUDA 12.1 지원)
  - **YOLO (Ultralytics)**: 8.3.48
  - **CLIP**: 1.0

정량적 평가에서는 기존 YOLO 모델을 활용한 객체 탐지 결과와, YOLO와 CLIP 모델을 결합한 객체 탐지 결과를 비교하였다. 이를 통해 두 모델 간의 성능 차이를 분석하고, YOLO+CLIP 파이프라인이 텍스트 기반 분류와 결합했을 때의 성능 향상을 평가하였다.

정성적 평가의 경우 "a blue cup"이라는 텍스트 프롬프트를 사용하여, 여러 개의 컵이 있는 이미지에서 파란 컵만을 구별하는 실험을 수행하였다. 이를 통해, 제시된 텍스트 프롬프트를 기반으로 객체를 정확히 식별할 수 있는지 여부를 확인하였다.

3.3. quantitative results

정량적 평가에서는 COCO validation 데이터셋에서 10개의 이미지를 추출하여 기존 YOLO 모델과 YOLO와 CLIP을 결합한 모델에 대해 각각 평가를 수행하였다. 그 결과, 아래와 같은 성능을 보였다.

Metric	YOLO Only	YOLO + CLIP Pipeline
Precision	0.05	0.16
Recall	0.07	0.21
F1-Score	0.06	0.17
Accuracy	0.07	0.21

Table 1. Evaluation Results for YOLO Only and YOLO + CLIP Pipeline

Figure 2. qualitative result

수치적으로, YOLO와 CLIP을 결합한 모델이 기존의 YOLO 모델보다 성능이 개선된 것으로 나타났다. 그러나 기존 YOLO 모델은 이미 COCO 데이터셋에 대해 사전 학습(pre-trained)이 되어 있기 때문에, 해당 데이터셋에 대한 평가에서 상대적으로 더 유리한 조건을 가지고 있으며, 이는 두 모델 간의 성능 차이가 실제 성능 향상을 반영하지 못할 수 있음을 의미한다. 따라서, 본 연구의 평가 방법이 모델의 실제 능력을 제대로 반영하지 못했을 가능성이 있으며, 보다 다양한 데이터셋과 조건에서의 평가가 필요할 것으로 판단된다.

3.4. qualitative results

정성적 평가를 위한 실험의 결과는 Figure 2와 같다. 주어진 이미지에서, 텍스트 프롬프트로 ”a blue cup”을 입력하여 similarity scores에 softmax를 적용한 값이 임계값인 0.5를 초과할 경우 bounding box를 이미지 위에 similarity score과 함께 나타나도록 하였다. 이를 통해 세 개의 컵 중 파란 컵만을 분류하려 했으나, 세 컵 모두 similarity scores의 softmax 값이 1.00으로 나타나 파란 컵을 분류하는데 실패하였다.

3.5. discussion

실험 결과, YOLO와 CLIP을 결합한 모델은 기존의 YOLO 모델에 비해 향상된 성능을 보였으나, 정성적 평가에서 주어진 텍스트 프롬프트에 맞는 객체를 정확하게 구분하지 못하는 결과를 보였다. 특히, YOLO 모델이 COCO 데이터셋에서 사전 학습(pre-trained)된 점을 고려했을 때, 기존 YOLO 모델의 성능이 상대적으로 낮게 평가된 것은 이상적인 결과가 아니었다. 이는 정량적 평가 방법이 모델의 실제 성능을 적절히 반영하지 못했음을 시사한다. 정성적 평가에서도, YOLO와 CLIP을 결합한 모델은 기대했던 수준의 성능을 발휘하지 못했다. 또한, 본 연구에서 제시한 YOLO와 CLIP 결합 모델의 목적은 동일 클래스 내에서 보다 세밀하고 유연한 객체 탐지를 가능하게 하는 것이지만, 현재 사용된 정량적 평가 방법은 이 목적에 적합하지 않음을 알 수 있었다.

4. Future Directions

본 연구에서는 YOLO와 CLIP을 결합한 모델을 통해 텍스트 프롬프트 기반의 객체 탐지 및 분류를 시도하였으나, 실험 결과 모델의 성능이 기대에 미치지 못한 결과를 보였다. 향후 연구에서는 객체 탐지뿐만 아니라, 보다 정교한 객체 분할을 위한 모델을 도입할 수 있다. 예를 들어, SAM(Segment Anything Model)과 같은 최신 분할(segmentation) 모델을 사용하면, 클래스에 대한 제약 없이 텍스트 프롬프트를 기반으로 객체를 정확하게 분할하고 분류하는 제로샷(zero-shot) 방식의 탐지가 가능해질 것이다. SAM은 다양한 클래스에 대해 사전 학습된 지식 없이도 텍스트 프롬프트에 맞는 객체를 정확하게 분할할 수 있는 잠재력을 지니고 있으며, 이는 본 연구에서 제시한 접근 방식을 확장하고 성능을 향상시킬 수 있는 중요한 방향이 될 것이다.

또한, 본 연구에서 제시한 것처럼 단순히 두 모델을 파이프라인 방식으로 결합하는 것만으로는 최적의 성능을 이끌어내기 어려울 수 있다. 향후 연구에서는 두 모델을 더 효과적으로 결합하기 위한 방법을 모색할 필요가 있다. 예를 들어, 두 모델의 학습 과정을 서로 관련지어 학습시키거나, 두 모델을 통합한 새로운 모델을 설계하는 방법을 고려할 수 있다. 이를 통해 두 모델 간의 상호 보완적인 특성을 극대화하고, 보다 효율적이고 정교한 객체 탐지 및 분류 시스템을 구축할 수 있을 것이다. 이러한 모델 통합 방식은 파이프라인 방식에서 발생할 수 있는 성능 저하를 최소화하고, 모델 간의 시너지를 극대화하는 중요한 개선 방안이 될 것이다.

References

[1] Jiafeng Li, Shengyao Sun, Kang Zhang, Jing Zhang, and Li Zhuo, "Single-stage zero-shot object detection network based on CLIP and pseudo-labeling," *International Journal of Machine Learning and Cybernetics*, August 2024.

[2] Haoxiang Wang, Pavan Kumar Anasosalu Vasu, Fartash Faghri, Raviteja Vemulapalli, Mehrdad Farajtabar, Sachin Mehta, Mohammad Rastegari, Oncel Tuzel, and Hadi Pouransari, "SAM-CLIP: Merging Vision Foundation Models towards Semantic and Spatial Understanding," submitted on 23 Oct 2023 (v1), last revised 10 Jun 2024 (this version, v4).

[3] Sidra Aleem, Fangyijie Wang, Mayug Maniparambil, Eric Arazo, Julia Dietlmeier, Guenole Silvestre, Kathleen Curran, Noel E. O'Connor, and Suzanne Little, "Test-Time Adaptation with SaLIP: A Cascade of SAM and CLIP for Zero-shot Medical Image Segmentation," submitted on 9 Apr 2024 (v1), last revised 30 Apr 2024 (this version, v2).