Segment Anything Model을 활용한 객체 기반 이미지 검색 시스템 설계 및 성능 분석

Design and Performance Analysis of Object-based Image Retrieval System Using Segment Anything Model

요 약

본 연구는 Meta AI의 Segment Anything Model(SAM)과 OpenAI의 CLIP을 결합한 객체 기반 이미지 검색 시스템을 구현하고 성능을 분석하였다. 이미지 내 객체를 분할하고 시각적 특징을 추출하여 유사 객체를 검색하는 과정에서, 객체 분할 방식(마스킹/크롭)과 거리 측정 방식(코사인/유클리디안)에 따른 성능 차이를 검증하였다. 실험 결과, 크롭 방식이 마스킹 방식보다 우수한 성능을 보였으며(mAP: 0.53 vs 0.29), 정규화된 벡터에서는 두 거리 측정 방식 간 성능 차이는 없었다. 본 연구는 범용 이미지 분할 모델과 비지도 학습 기반 임베딩을 결합한 검색 시스템의 효과를 입증하며, 특히 모델 학습 데이터 분포와 유사한 데이터 처리 방식의 중요성을 확인하였다.

1. 서 론

이미지 검색 시스템은 사용자의 질의에 대응하여 관련 이미지를 검색하는 기술로, 다양한 분야에서 활용되고 있다. 기존 시스템들이 주로 전체 이미지를 대상으로 검색을 수행하는 반면, 사용자는 이미지 내 특정 객체에만 관심을 갖는 경우가 빈번하여 객체 단위 검색 시스템의 필요성이 증가하고 있다.

객체 기반 이미지 검색의 핵심 요소는 객체 분할과 특징 추출이다. Meta AI의 Segment Anything Model(SAM)[1]은 다양한 객체를 정확하게 분할하는 우수한 성능을 보이며, OpenAI의 CLIP[2] 모델은 풍부한 시각적 특징 표현을 생성한다. 본 연구에서는 SAM의 정밀한 객체 분할 능력과 CLIP의 강력한 시각적 임베딩을 결합한 객체 기반 이미지 검색 시스템을 제안하고, 객체 분할 방식(마스킹 vs 크롭)과 거리 측정 방식(코사인 유사도 vs 유클리디안 거리)에 따른 성능 차이를 분석하였다. 본 연구의 주요 기여점은 다음과 같다. 첫째, SAM과 CLIP을 효과적으로 결합한 객체 기반 검색 시스템의 구현 방법론을 제시한다. 둘째, 객체 분할 방식과 거리 측정 방식이 검색 성능에 미치는 영향을 실험적으로 규명한다. 셋째, 사전학습된 모델의 학습 데이터 분포와 유사한 데이터 처리 방식이 검색 성능 향상에 기여하는 정도를 검증한다.

2. 방법론

본 연구에서는 객체 기반 이미지 검색 시스템을 구현하기 위해 다음과 같은 단계별 접근 방식을 취하였다. 전체 시스템 파이프라인은 데이터셋 준비, 객체 분할, 특징 추출, 검색 인덱스 구축, 성능 평가의 다섯 단계로 구성된다.

2.1 데이터셋

MS COCO 2017 데이터셋[3]의 validation 세트에서 30개 이미지를 샘플링하였다. '사람', '개', '고양이', '자동차', '자전거' 등 주요 카테고리를 포함하도록 하였으며, 각 카테고리별로 최소 2개 이상의 객체가 포함되도록 선별하였다. 최종적으로 선별된 30개 이미지에서 총 47개의 객체를 추출하였으며, 각 객체는 마스킹과 크롭 두 가지 방식으로 처리되어 총 94개의 객체 이미지가 분석에 사용되었다.

2.2 객체 분할

객체 분할을 위해 SAM의 ViT-B 버전을 사용하였다. COCO 데이터셋의 바운딩 박스 정보를 프롬프트로 사용하여 객체 마스크를 생성하였다. SAM을 통해 생성된 객체 마스크는 두 가지 방식으로 처리하였다. 첫째, 마스킹(Masking) 방식은 원본 이미지에서 객체 영역만 유지하고 배경 부분을 검은색(픽셀값 0)으로 대체하는 방법이다. 이 방식은 객체 자체에만 집중할 수 있다는 장점이 있으나, 객체의 문맥 정보가 손실될 수 있다. 둘째, 크롭(Cropping) 방식은 바운딩 박스 영역을 기준으로 이미지를 잘라내는 방법이다. 이 방식은 객체 주변의 배경 정보를 일부 포함하므로 객체의 문맥 정보가 보존되지만, 관련 없는 배경이 포함될 수 있다는 단점이 있다.

2.3 특징 추출

분할된 객체 이미지의 특징 추출을 위해 CLIP(ViT-B/32) 모델을 사용하였다. CLIP은 이미지와 텍스트를 동일한 임베딩 공간에 매핑하는 사전학습 모델로, 본 연구에서는 이미지 인코더를 활용해 각 객체를 512차원 특징 벡터로 변환하였다. CLIP 모델의 특징 추출 과정은 다음과 같다. 먼저 입력 이미지를 224×224 크기로 리사이즈하고 모델에서 요구하는 정규화 과정을 적용한다. 이후 CLIP의 이미지 인코더를 통과시켜 512차원의 특징 벡터를 얻는다. 마지막으로 추출된 특징 벡터에 L2 정규화를 적용하여 단위 벡터로 변환한다.

2.4 검색 인덱스 구축

추출된 특징 벡터를 기반으로 효율적인 유사도 검색을 위해 Faiss 라이브러리[4]를 활용하여 검색 인덱스를 구축하였다.
본 연구에서는 두 가지 유형의 인덱스를 구현하였다. 첫째, 코사인 유사도 기반 인덱스는 내적(Inner Product) 계산을 통해 벡터 간 유사도를 측정한다. L2 정규화된 벡터의 내적은 코사인 유사도와 동일하며, 값이 1에 가까울수록 높은 유사도를 나타낸다. 둘째, 유클리디안 거리 기반 인덱스는 두 벡터 간 직선 거리를 계산하며, 이 경우 값이 0에 가까울수록 높은 유사도를 의미한다.

2.5 평가 지표

검색 성능 평가를 위해 같은 카테고리의 객체들을 관련성 있는 것으로 간주하고 다음 지표들을 사용하였다.

$$\text{Precision@K} = \frac{|\{\text{relevant items}\} \cap \{\text{retrieved items at K}\}|}{K}$$

(그림 1) Precision@K(정밀도)

(그림 1)은 상위 K개 검색 결과 중 관련 있는 항목의 비율을 측정한다. 이는 검색된 결과가 얼마나 정확한지를 나타낸다.

$$\text{Recall@K} = \frac{|\{\text{relevant items}\} \cap \{\text{retrieved items at K}\}|}{|\{\text{relevant items}\}|}$$

(그림 2) Recall@K(재현율)

(그림 2)는 관련 있는 전체 항목 중 상위 K개 검색 결과에 포함된 항목의 비율을 측정한다. 이는 관련 항목 중 얼마나 많이 검색되었는지를 나타낸다.

$$AP = \sum_{k=1}^{n} P(k) \cdot rel(k)$$

(그림 3) mAP(mean Average Precision)

(그림 3)은 검색 결과의 순위를 고려한 정밀도의 평균으로, 검색 시스템의 전반적인 성능을 종합적으로 평가한다. 모든 관련 항목의 위치에 가중치를 부여하여 상위 순위에 관련 항목이 있을수록 높은 점수를 부여한다.

여기서 P(k)는 k번째 위치까지의 정밀도, rel(k)는 k번째 항목이 관련 있으면 1, 아니면 0을 의미한다.

본 연구에서는 K값을 5로 설정하여 Precision@5와 Recall@5를 측정하였으며, 각 쿼리 객체마다 이러한 지표를 계산하고 전체 쿼리에 대한 평균값을 최종 성능 지표로 사용하였다.

3. 실험 결과

30개의 이미지에서 SAM 모델을 통해 총 47개의 객체가 분할되었으며, 각 객체마다 마스킹과 크롭 두 가지 방식으로 처리하여 총 94개의 임베딩이 생성되었다. 분할된 객체의 카테고리별 분포는 사람(42.6%), 자동차(17.0%), 고양이(14.9%), 개(12.8%), 자전거(8.5%), 기타(4.2%)로 나타났다.

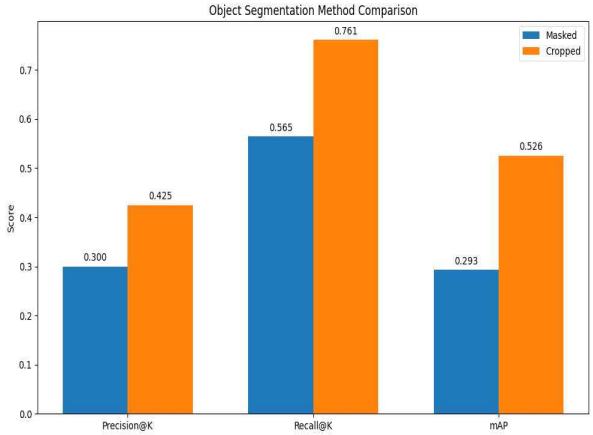
3.1 분할 방식에 따른 성능 비교

분할 방식에 따른 검색 성능 비교 결과는 <표 1>과 같다. 크롭 방식이 마스킹 방식보다 모든 평가 지표에서 우수한 성능을 보였다.

<표 1> 분할 방식별 검색 성능 비교

분할 방식	Precision@K	Recall@K	mAP
마스킹 (Masked)	0.3000	0.5647	0.2933
크롭 (Cropped)	0.4250	0.7612	0.5257

특히 mAP 지표에서는 약 0.23의 큰 차이를 보였다. 이는 크롭 방식이 객체의 문맥 정보를 더 잘 보존하고, CLIP 모델의 학습 데이터 분포와 더 유사하기 때문으로 해석된다.



(그림 1) 객체 분할 방법 비교

(그림 1)은 분할 방식별 검색 성능을 비교한 결과를 보여준다. 모든 평가 지표에서 크롭 방식이 마스킹 방식보다 우수한 성능을 보인다. 특히 Recall@K에서 가장 큰 차이(0.761 vs 0.565)를 보여, 크롭 방식이 관련 객체를 더 많이 검색해낼 수 있음을 알 수 있다.

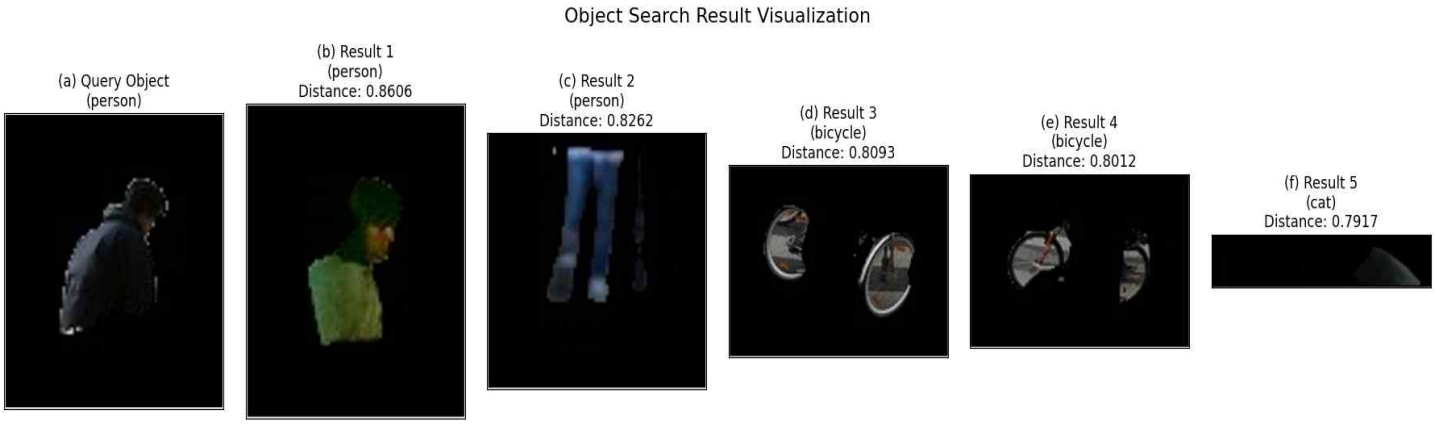
3.2 거리 측정 방식에 따른 성능 비교

코사인 유사도와 유클리디안 거리 방식의 검색 성능 비교 결과는 <표 2>와 같다. 정규화된 벡터를 사용하였기 때문에 두 거리 측정 방식 간 성능 차이는 없는 것으로 나타났다.

<표 2> 거리 측정 방식별 검색 성능 비교

거리 측정 방식	Precision@K	Recall@K	mAP
코사인 (Cosine)	0.3800	0.5732	0.3783
유클리디안 (Euclidean)	0.3800	0.5732	0.3883

이는 단위 벡터 간의 유클리디안 거리의 제곱이 코사인 유사도와 단조 관계에 있기 때문이며, 정규화된 벡터를 사용할 때는 계산 효율성에 따라 거리 측정 방식을 선택할 수 있음을 의미한다.



(그림 2) 객체 검색 결과 시각화

3.3 검색 결과 사례 분석

(그림 2)는 '사람' 카테고리 객체를 쿼리로 사용한 검색 결과 예시를 보여준다. (a)는 쿼리 객체(사람)이며, (b)부터 (f)까지는 상위 5개 검색 결과를 나타낸다. 결과 1(b)과 결과 2(c)는 같은 '사람' 카테고리이지만, 결과 3(d)와 결과 4(e)는 '자전거' 카테고리, 결과 5(f)는 '고양이' 카테고리로 나타났다. 각 결과 위에 표시된 거리 값은 쿼리 객체와의 코사인 유사도를 나타낸다.

이는 CLIP 모델이 객체의 카테고리 뿐만 아니라 형태, 색상, 질감 등 다양한 시각적 특성을 고려하여 임베딩을 생성하기 때문으로 해석된다. 특히 자전거 객체가 사람 객체와 유사하게 나타난 경우, 자전거를 타고 있는 사람이 함께 포함되어 있어 이러한 결과가 나타난 것으로 보인다.

4. 논의

크롭 방식이 마스킹 방식보다 우수한 성능을 보인 이유로는 여러 요인을 고려할 수 있다. 우선 크롭 방식은 객체 주변의 배경 정보를 일부 포함하여 객체의 문맥 정보를 더 잘 보존한다. 또한 마스킹 방식은 객체 외부를 검은색으로 처리하여 인공적인 경계가 생기며, 이는 CLIP 모델의 특징 추출에 부정적 영향을 줄 수 있다. 마지막으로 CLIP 모델은 주로 크롭된 이미지로 학습되었으므로, 크롭 방식의 이미지가 모델의 학습 데이터 분포와 더 유사할 수 있다.

코사인 유사도와 유클리디안 거리 간 성능 차이가 없는 것은 벡터 정규화의 영향으로, L2 정규화된 벡터의 경우 두 거리 측정 방식은 수학적으로 다음과 같은 관계를 갖는다.

$$d^2(u,v) = ||u||^2 + ||v||^2 - 2 \cdot u \cdot v = 2 - 2 \cdot \cos(u,v)$$

(그림 3) 유클리디안 거리와 코사인 유사도 간의 관계

여기서 ||u|| = ||v|| = 1이므로, 두 방식은 동일한 순위의 검색 결과를 생성한다.

본 연구의 한계점으로는 소규모 데이터셋 사용으로 인한 통계적 신뢰도의 제한이 있다. 향후 연구에서는 더 대규모 데이터셋, 다양한 분할 프롬프트, 다양한 임베딩 모델, 다양한 배경 처리 기법 등을 통해 연구를 확장할 필요가 있다.

5. 결론

본 연구에서는 SAM과 CLIP을 활용한 객체 기반 이미지 검색 시스템을 구현하고 성능을 분석하였다. 실험 결과, 크롭 방식이 마스킹 방식보다 우수한 성능을 보였으며, 정규화된 벡터에서는 거리 측정 방식 간 성능 차이가 없었다. 이러한 결과는 객체 검색 시스템 설계 시 객체 분할 방식의 중요성을 보여주며, 특히 CLIP과 같은 범용 시각 모델을 사용할 때는 모델의 학습 데이터 분포와 유사한 데이터 처리 방식을 선택하는 것이 중요함을 시사한다. 본 연구는 최신 컴퓨터 비전 모델을 활용한 객체 기반 이미지 검색의 가능성을 보여주며, 향후 다양한 확장 연구를 통해 더 견고한 시스템을 개발할 수 있을 것이다.

참고문헌

[1] Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Tao, A., Moorosi, N., Gonzalez, J., Keutzer, K., Malik, J., Ross, D. A., Doll□r, P., & Girshick, R., "Segment anything," Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 4015–4026, 2023.

[2] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., & Sutskever, I., "Learning transferable visual models from natural language supervision," International Conference on Machine Learning, pp. 8748–8763, 2021.

[3] Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Doll□r, P., & Zitnick, C. L., "Microsoft COCO: Common objects in context," European Conference on Computer Vision, pp. 740–755, 2014.

[4] Johnson, J., Douze, M., & J□gou, H., "Billion-scale similarity search with GPUs," IEEE Transactions on Big Data, Vol. 7, No. 3, pp. 535–547, 2019.