

객체 탐지 성능 향상을 위한 데이터 증강 기법에 관한 연구

최은실, 한재웅, 김은경
국립한밭대학교 인공지능소프트웨어학과

{eunsil, hanjaewoong}@edu.hanbat.ac.kr, ekim@hanbat.ac.kr

An Approach to Data Augmentation for Object Detection

Eunsil Choi, Jaewong Han, Eunkyung Kim
Dept. of Artificial Intelligence Software, Hanbat National University

요약

본 연구에서는 데이터 증강 기법을 사용하여 적은 양의 원본 학습 데이터로도 높은 객체 탐지 정확도를 달성하는 방법을 제안한다. 객체 탐지를 위해 색상 마스크를 활용하여 객체의 윤곽선을 구분한 후, 가우시안 블러를 통해 객체를 제외한 배경 정보를 흐리게 하는 이미지 블러링 방식을 적용하였다. 신호등 이미지 데이터를 활용한 실험 결과, 원본 데이터 1100장을 사용할 때보다 전체 블러링 방식 1000장을 사용했을 때 1.484% 성능이 저하되었으나, 이는 원본 데이터보다 적은 양의 전처리된 데이터로도 상당한 성능을 유지할 수 있음을 보여준다. 또한, 객체를 제외한 배경 블러링 방식 1000장을 사용했을 때는 원본 데이터 1100장 대비 0.2% 더 높은 성능을 발휘하였다. 이를 바탕으로 본 연구의 데이터 증강 기법의 유용성을 입증하였고, 다양한 객체 탐지 응용 분야에 적용할 수 있는 연구 방향을 제시한다.

I. 서론

이동 로봇(또는 자율주행 자동차)이 주변 환경을 인식하고 객체 탐지(Object Detection)[1] 기능은 매우 중요하다. 특히, 도로 주행을 위해 중요한 신호등 탐지와 신호등 검출이 중요하다. 색상뿐만 아니라 채도와 명도 값을 조절하여 특정 색상을 더 정교하게 설정하는 HSV 필터링을 통해 신호등 객체 탐지와 빨간색과 초록색을 88.8%, 84.6%의 정확도로 구분할 수 있다[2]. 그러나 HSV 변환에서 H(색상)와 S(채도) 값을 변화시키면 원본 사진의 색상이 크게 달라져 전체적인 노이즈가 증가하고, 현실적인 이미지에서 벗어나 비현실적인 결과가 나타난다. 이로 인해 모델 성능이 감소하는 문제가 발생하며, 실제로 이러한 사진들을 CNN 모델에 학습시킨 결과, 동일한 Epoch=25에서 원본 대비 성능이 38.73%나 낮아지는 결과를 확인하였다. 이를 방지하기 위해 H와 S를 그대로 유지하고, V(명도)만을 증가시키는 방식으로 HSV 전처리를 대체하였다. 그러나 이 방법 역시 원본 데이터 학습 모델의 성능 대비 약 3% 낮은 성능을 보여, HSV를 활용한 데이터 전처리가 항상 최적의 방법은 아닐 수 있음을 시사하였다.

본 연구에서는 이미지에서 고주파 성분을 제거하여 부드럽게 만드는 효과를 가지며, 결과적으로 불필요한 세부 패턴이나 노이즈를 제거하는 데 유용한 가우시안 블러를 적용하여 신호등 탐지와 신호등 검출 방식을 제안한다. 제안하는 방식은 객체 탐지를 위해 색상 마스크를 활용하여 객체의 윤곽선을 구분한 후, 가우시안 블러를 통해 객체를 제외한 배경 정보를 흐리게 하는 이미지 블러링 방식을 적용한다. CNN 모델에서 신호등 이미지 데이터를 활용한 실험 결과, 원본 데이터 1100장을 사용할 때보다 전체 블러링 방식 1000장을 사용했을 때 1.484% 객체 탐지 성능이 저하되었으나, 객체를 제외한 배경 블러링 방식 1000장을 사용했을 때 원본 데이터 1100장 대비 0.2% 더 높은 성능을 보인다.

II. 가우시안 블러를 활용한 데이터 증강 및 객체 탐지

본 연구에서는 객체 탐지를 위해 가우시안 블러[3]를 활용하여 객체를 강조하고 윤곽선을 명확하게 하여 객체와 배경을 분리(그림2 참조)하고, 분리된 배경 정보를 흐리게 하는 이미지 블러링 방식을 적용(그림1 참조)한다. [그림 1]은 객체 탐지 후 해당 객체를 강조하고, 배경을 흐리게 처리하는 과정을 나타내고 있다.

```
1. if largest_mask is not None:
2.     kernel = np.ones((4,4),np.uint8)
3.     largest_mask_dilated = cv2.dilate(largest_mask,kernel,iterations=1)
4.     mask_inv = cv2.bitwise_not(largest_mask_dilated)
5.     colored_part = cv2.bitwise_and(img,img,mask=largest_mask_dilated)
6.     blurred_img = cv2.GaussianBlur(img,(31,31),0)
7.     background_part = cv2.bitwise_and(blurred_img,blurred_img,mask=mask_inv)
8.     result = cv2.add(colored_part,background_part)
```

그림 1. 가우시안 블러를 적용한 객체 탐지

우선, 객체 탐지를 통해 생성된 마스크가 존재할 경우, 마스크의 경계를 확장(dilation)하여 객체의 윤곽을 더 명확하게 만든다. 이때, 4x4 크기의 커널을 사용하여 마스크를 확장하고, 이 확장된 마스크를 기반으로 객체를 강조한다. 확장된 마스크를 반전하여 배경 영역을 추출한 후, 이미지 전체에 가우시안 블러를 적용하여 배경을 흐리게 만든다. 블러링된 배경과 객체를 결합함으로써 최종적으로 객체가 강조된 이미지를 생성한다.

```
9. hsv = cv2.cvtColor(img,cv2.COLOR_BGR2HSV)
10. mask_green = cv2.inRange(hsv,np.array([35,50,50]),np.array([85,255,255]))
11. mask_red = cv2.bitwise_or
12. (cv2.inRange(hsv,np.array([0,50,50]),np.array([10,255,255])),
13. cv2.inRange(hsv,np.array([170,50,50]),np.array([180,255,255])))
14. largest_mask_green = find_largest_contour(mask_green)
15. largest_mask_red = find_largest_contour(mask_red)
```

그림 2. 색상 마스크 코드(find_largest_contour는 [8] 참조)

[그림 2]는 OpenCV[4]를 통한 색상 마스크[5] 기법으로, 신호등을 제외한 배경 블러링을 통해 데이터를 증강시키는 방법이다. 색상 마스크는 특정 색상을 기준으로 이미지를 분리하는 기술로, 주로 HSV 색상 공간에서 특정 색상의 범위를 지정하여 해당 색상을 포함하는 영역을 마스크로 생성한다. 먼저, 객체를 제외한 배경만 블러 처리하기 위해 색상 마스크를 활용하여 신호등의 색상을 구분한다. HSV 색상 공간으로 변환된 이미지에서 신호등의 녹색과 빨간색을 정확히 식별하고, 이 색상 정보를 기반으로 녹색 및 빨간색 마스크를 각각 생성한다. 이렇게 생성된 마스크를 결합하여 빨간색 또는 초록색 신호등 객체를 포함하는 영역을 분리한다. 이후, 가장 큰 윤곽선을 찾아 신호등 객체를 정확하게 추출하고, 신호등 객체를 제외한 나머지 배경 부분에 대해 역마스크를 생성한다. 역마스크는 신호등 객체를 보호하고 배경만을 타겟으로 하기 위해 사용되며, 배경 부분에 가우시안 블러를 적용하여 배경 정보를 흐리게 한다. 가우시안 블러는 이미지의 가장자리를 부드럽게 처리하여 배경의 불필요한 패턴을 제거하고, 이를 통해 객체 탐지 모델이 신호등을 더욱 명확하게 인식할 수 있도록 한다.

III. 실험 결과

본 연구에서는 Convolutional Neural Network (CNN) 모델[6]을 활용하여 학습을 진행한다. 실험을 위해 EV3 카메라 모듈[7]로 촬영한 원본 이미지 데이터 1100장, 전체 블러링한 이미지 데이터 1000장, 객체를 제외한 배경 블러링 이미지 데이터 1000장을 사용¹⁾한다.



그림 3. 원본 이미지(왼쪽), 사진 전체를 블러링 한 이미지(가운데), 객체 제외 배경만 블러링 한 이미지(오른쪽)

[그림 3]은 원본 이미지와 사진 전체를 블러링 한 이미지, 객체 제외 배경만 블러링 한 이미지를 나타낸다. 사진 전체 블러링 이미지는 객체까지 블러링을 적용하게 되어, 신호등을 포함하는 모든 이미지의 세부 정보가 명확하게 보이지 않게 된다. 그러나 객체 제외 배경만 블러링 처리한 데이터는 객체의 정보는 최대한 보존할 수 있다.

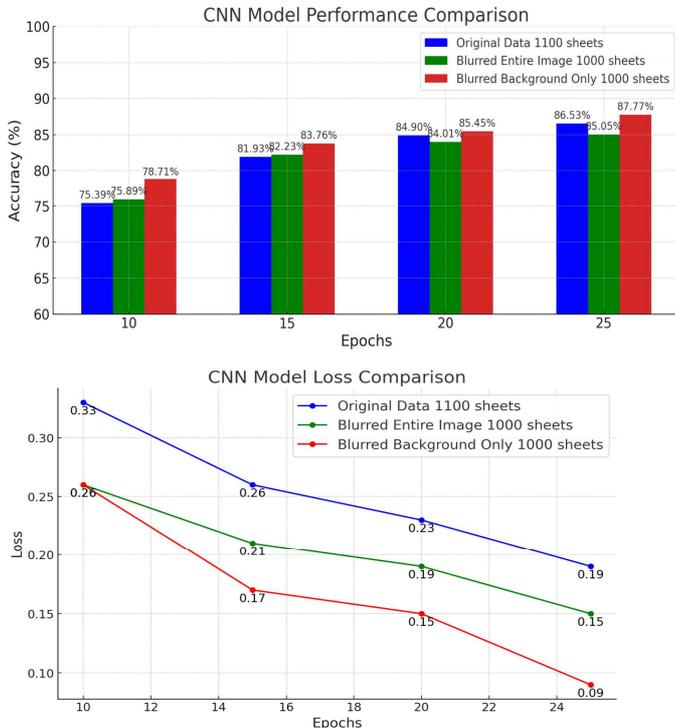


그림 4. 신호등 객체 탐지 정확도(위), Loss(아래) 비교(Epoch=10, 15, 20, 25)

[그림 4]는 [그림 3]과 같이 원본 데이터(1100장)와 전체 블러링 데이터(1000장), 객체를 제외한 배경 블러링 데이터(1000장)의 3가지 종류의 데이터 셋의 Epoch에 따른 객체 탐지 정확도(위)와 Loss(아래)를 나타낸다.

원본데이터를 모델에 학습시킨 경우, Epoch가 증가함에 따라 모델의 정확도도 증가하여, Epoch=25에서 86.53%에 도달한다. 사진 전체 블러링 데이터를 학습한 모델은, 초기 Epoch에서는 원본 데이터와 유사한 성능을 보였으나, Epoch=25에서는 정확도가 85.05%로 감소한다. 이는 이미지 전체를 블러링하여 중요한 정보까지 손실됨으로써 모델 성능이 저하되기 때문이다. 반면, 객체 제외 배경 블러링 데이터를 학습한 모델은 Epoch=15 이후부터 원본 데이터 사용 모델보다 더 높은 정확도를 기록하며, Epoch=25에서는 87.37%로 가장 높은 성능을 보인다.

원본 데이터를 학습시킨 모델을 사용한 경우, Epoch가 증가함에 따라 손실이 점진적으로 감소하는 경향을 보이나, 여전히 다른 기법들에 비해 상대적으로 손실 값이 높다. Epoch=10에서 시작된 손실 값이 0.33에서 Epoch=25에서는 0.19로 감소하지만, 여전히 다른 방법들보다 손실율이 높다. 전체 배경 블러링 데이터를 학습시킨 모델은, Epoch=10에서 손실이 0.26으로 시작하여 Epoch=25에서는 0.15로 감소한다. 이 방법은 원본 데이터보다 손실 값이 낮으나, 객체 제외 배경만 블러링한 데이터를 학습시킨 모델보다는 여전히 손실 값이 높다. 반면, 객체 제외 배경만 블러링한 데이터를 학습시킨 모델은 Epoch=10에서 시작된 손실 값이 0.20에서 Epoch=25에서는 0.09으로 손실 값이 가장 낮다.

또한 Epoch= 25를 기준으로 모델을 10번 반복 학습한 결과 원본 데이터를 사용한 모델의 정확도는 86.533% (표준편차 ±1.465)로, 비교적 안정적인 성능을 나타낸다. 반면, 전체 블러링 기법을 적용한 경우, 정확도는 85.049%로 감소하였고, 표준편차는 ±2.143으로 가장 큰 변동성을 보인다. 이는 이미지의 모든 부분을 블러링 처리함으로써 중요한 정보가 손실되어 모델 성능에 부정적인 영향을 미친 결과이다. 한편, 객체 제외 배경 블러링 기법을 적용한 경우, 정확도는 86.733% (표준편차 ±1.993)로 가장 높다. 이는 CNN 모델이 중요한 정보인 객체에 더 집중할 수 있게 한 결과이다. 따라서, 이 연구는 객체 제외 배경 블러링 기법이 객체 탐지 성능 향상에 효과적임을 입증하였으며, 데이터 증강을 통해 적은 양의 원본 데이터로도 높은 정확도를 달성할 수 있음을 보여준다.

IV. 결론

본 연구에서 신호등 이미지 데이터를 활용하여 이미지 마스킹을 통한 객체 제외 배경 블러링 기법을 제안하였다. 실험을 통해 원본 이미지 데이터만 사용했을 때 보다 제안한 방법을 사용했을 때 객체 인식의 정확도가 개선된 것을 확인하였다. 특히, 더 적은 양의 사진을 활용했음에도 불구하고 제안된 데이터 증강 기법 중 색상 마스킹과 배경 블러링 기법이 신호등 객체 탐지 성능을 향상시킬 수 있음을 확인하였다.

ACKNOWLEDGMENT

본 연구는 2024년 과학기술정보통신부 및 정보통신기획평가원의 SW중심대학사업의 연구결과(2022-0-01068)와 대학ICT연구센터사업의 연구결과(IITP-2024-RS-2024-00437886)로 수행되었음.

참고 문헌

- [1] Shao, S., Li, Z., Zhang, T., & Xiao, J. (2019). Objects365: A large-scale, high-quality dataset for object detection. Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019.
- [2] 김동건. *딥러닝과 HSV를 활용한 신호등 검출 및 신호 분류 방법* Diss. 한양대학교, 2018.
- [3] Singhal, P., Verma, A., & Garg, A. (2017). A study in finding effectiveness of Gaussian blur filter over bilateral filter in natural scenes for graph based image segmentation. In 4th International Conference on Advanced Computing and Communication Systems (ICACCS), 2017.
- [4] OpenCV [Website]. (Aug 06, 2024). <https://opencv.org/>
- [5] Smith, J. R. (1977). The application of color to image segmentation. Computer Graphics and Image Processing, 6(2), pp. 254-277, 1977.
- [6] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). R-CNN: Rich feature hierarchies for accurate object detection and semantic segmentation. arXiv preprint arXiv:1311.2524, 2014.
- [7] Lego Mindstorms. (Aug 06, 2024). Lego. <https://www.lego.com/ko-kr/themes/mindstorms?consent-modal=show&age-gate=gro>
- [8] Choi E. and Han, J. *Traffic Light Classification Code*. 2024. GitHub, <https://github.com/fSWLab/An-Approach-to-Data-Augmentation-for-Object-Detection>

1) 객체 제외 배경 블러링 데이터 84.6%, 88.8%의 객체 탐지 성공률로 빨간색 신호등과 초록색 신호등을 각각 제대로 탐지(IoU ≥ 0.5)하여 데이터 전처리 진행함[8].