https://doi.org/10.7840/kics.2023.48.6.722

자율주행 노면 청소차의 쓰레기 객체 탐지 성능 향상을 위한 이미지 합성 기법

배 장 훈◆, 박 병 준*, 최 인 우*, 김 재 원*, 김 민 호°

Image Synthesis Techniques for Improving Trash Object Detection Performance of Self-Driving Road Cleaning Vehicles

Jang Hoon Bae◆, Byoung Jun Park*, In U Choi*, Jaewon Kim*, Minhoe Kim°

요 약

컴퓨터 비전 분야에서 딥러닝을 이용한 객체 인식이 활발히 연구되면서 다양한 분야에서 사용되고 있다. 여러 분야에서 딥러닝 객체 인식 모델을 활용하기 위해서 학습 데이터의 양과 질뿐만 아니라 다양성 역시 중요한 요소로 자리 잡게 되었다. 하지만 객체 인식 모델을 활용하고자 하는 분야가 특수하여 데이터가 희귀한 경우에는 모델의 학습에 필요한 많은 양의 데이터를 수집할 수 없는 상황이 발생할 수 있다. 본 논문에서는 도로 위의 쓰레기를 인식하여 청소하는 자율주행 노면 청소차라는 특수한 환경에서의 객체 인식 모델을 학습시키기 위해 이미지를 합성하는 데이터 증강 기법을 연구한다. 나아가서 도로 환경에서 좀 더 사실적으로 합성하기 위한 원근 변환을 적용하는 방법을 제안하고 객체 인식 모델에 끼치는 영향을 실험을 통해 분석한다.

키워드 : 객체 인식, 데이터 증강, Copy-Paste, 도로 환경, 이미지 합성, 원근 변환

Key Words : Object detection, Data augmentation, Copy-Paste, Road environment, Image synthesis, Perspective transform

ABSTRACT

Object detection using deep learning has been actively studied in the computer vision task and is used in various tasks. In order to utilize the deep learning object detection model in various tasks, not only the quantity and quality of training data but also diversity has become an important factor. However, if the data is rare because the task which utilizes the object detection model is special, it may not be possible to collect the large amount of data required for the model's training. In this paper, we study a data augmentation technique that synthesizes images to learn an object recognition model in a special environment called an autonomous road cleaner that recognizes and cleans trash on the road. Furthermore, we propose a method of applying perspective transformations to synthesize more realistic data and analyze the impact on object detection model through experiments.

※ 본 연구는 2023년도 교육부의 재원으로 한국연구재단의 지원을 받아 수행된 지자체-대학 협력기반 지역혁신 사업의 결과입니다.(2021RIS-004)

◆ First Author : Computer and Information Science Department, Korea University, zptk97@korea.ac.kr, 학생회원

° Corresponding Author : Computer and Information Science Department, Korea University, kimminhoe@korea.ac.kr, 정회원

* Department of Computer Convergence Software, Korea University Sejong Campus, pbjun3348@korea.ac.kr; chedel@korea.ac.kr; galaxy78@korea.ac.kr

논문번호 : 202302-028-C-RU, Received February 15, 2023; Revised April 4, 2023; Accepted April 5, 2023

Ⅰ. 서 론

객체 인식 분야에서 Faster-RCNN[1], SSD[2], FCOS[3], YOLO[4]와 같은 딥러닝 기술 기반 객체 인식 모델들은 높은 정확도를 보여준다. 이러한 객체 인식 모델들은 많은 학습 데이터 양을 기반으로 좋은 성능을 보여줄 수 있다. 하지만 데이터의 양이 확보되지 않았을 때는 검출 성능이 보장되지 않는다는 문제가 있다. 따라서 객체 인식 모델을 학습하려 할 때, 주석이 달린 많은 양의 데이터 확보는 중요한 과제이다. 하지만 학습에 이용할 수 있는 데이터를 직접 수집하는 것은 시간과 비용의 소모가 크고, 데이터에 주석을 추가하는 것 역시 큰 비용이 소모된다. 필요로 하는 데이터 셋을 확보할 수 있는 경우에는 이러한 부분을 고려하지 않아도 되지만 확보하기 어려운 희귀 객체를 인식하는 모델을 학습하려 할 때, 적합한 데이터를 확보하는 것이 문제가 된다.

앞서 설명한 문제 해결을 위해 렌더링 객체를 합성하여 데이터를 생성하는 방법[5]이 연구되었고 이를 통해 직접 수집하는 것보다 희귀 객체 데이터를 쉽게 확보할 수 있고 주석 추가에 드는 비용도 절감할 수 있다. 하지만 실제 객체와 렌더링 객체 사이의 간극을 줄이는데 많은 노력이 필요하고 실제 데이터에 일반화하기 어렵다는 단점이 있다[6]. 실제 객체를 복사 후 붙여넣기 하여 데이터를 생성하는 방법(Copy-Paste)이 연구되어 이러한 문제가 해결되었고, 나아가 배경 이미지에 붙여 넣은 객체가 더욱 자연스럽게 보이기 위해 gaussian blur, poisson blending과 같은 기법들이나 크기 변환, 색 변환, 시각적 문맥 기반 배치와 같은 방법들이 사용되었다[7-11].

본 논문에서 언급하는 자율주행 노면 청소차는 도로 환경에서 쓰레기 객체를 인식하여 청소한다. 이때 쓰레기를 인식하는 것은 자율주행 노면 청소차의 임무 수행을 위해 필수적으로 선행되어야 하기에 가장 중요한 기능이고, 정확한 인식을 위해 카메라를 통해 입력 받은 이미지를 이용해 쓰레기 객체를 인식하여야 한다. 이러한 자율주행 노면 청소차의 객체 탐지 모델의 학습을 위해서는 도로 환경에 쓰레기가 놓여 있는 이미지 데이터가 요구된다. 하지만 도로 환경에 쓰레기가 존재하는 데이터는 굉장히 적고, 그러한 데이터들만 수집한 데이터 셋은 존재하지 않는다. 이런 경우 자율주행 노면 청소차의 객체 인식 모델의 학습을 위한 데이터 확보는 큰 비용이 소모되기 때문에 Copy-Paste 데이터 증강 기법이 효과적이다. 하지만 도로 환경의 쓰레기 중에는 전단지와 같이 높이가 0.5cm 이하인 납작한 객체가 존재하고 그림 3에서 볼 수 있듯이 기존의 Copy-Paste 데이터 증강 기법[7~11]으로는 이러한 객체를 자연스럽게 배치할 수 없다.

본 논문에서는 도로 환경에서 납작한 객체를 자연스럽게 붙여넣기 위해 도로 환경에서의 원근감을 구현할 수 있는 원근 변환 방법을 제안하여 Copy-Paste 데이터 증강 기법을 개선한다. 그리고 제안한 방법이 자율주행 노면 청소차의 임무 수행을 위한 쓰레기 객체 탐지 모델의 성능에 미치는 영향을 연구한다.

Ⅱ. 관련 연구

2.1 객체 탐지

객체 탐지는 최근 많이 연구되는 분야로서, 이미지에 어떠한 객체들이 어떠한 위치에 존재하는지 구별할 수 있는 기술이다. 객체 탐지 모델은 주로 2-stage detector와 1-stage detector로 구분할 수 있다. 2-stage detector는 지역 제안 단계와 분류 단계로 나누어 객체를 검출한다. 지역 제안 단계에서는 객체가 있을 만한 영역을 찾아내는 단계이다. 분류 단계에서는 객체가 어떤 객체인지, 객체의 종류를 판별하는 단계이다. 대표적인 2-stage detector로는 R-CNN[12], FasterRCNN[1] 등이 있다. 1-stage detector는 앞에 언급한 두 단계를 동시에 해결하여 속도를 높이는 방법이다. 대표적으로 YOLO[4], SSD[2] 등이 있다. 본 논문에서는 YOLOv7[13] 모델을 이용하여 도로 환경에서의 쓰레기 객체 탐지를 위한 데이터 증강을 연구한다.

2.2 희귀 객체 탐지

희귀 객체 탐지는 일반적인 객체 탐지에 비해 모델 학습을 위한 학습 데이터를 확보하기가 어렵다는 문제가 있다. 희귀한 객체에 대한 원초적인 이미지의 양이 적으며 이런 데이터를 이용하려는 곳 또한 적기 때문에 주석이 추가 되어있는 데이터 셋은 찾아보기 힘들고 데이터 셋이 있더라도 희귀 객체의 비율이 굉장히 적은 long-tail 데이터 셋일 가능성이 크다. 최근에는 이러한 문제를 해결하기 위하여 적은 양의 데이터 셋만으로 학습된 모델의 성능을 끌어올릴 수 있는 few-shot learning[14]이 사용되고 있다. 또 다른 해결책으로서 long-tail 데이터 셋의 tail 카테고리 데이터의 수와 head 카테고리 데이터의 수를 sampling하여 데이터의 분포를 조정하는 re-sampling[15]과 손실 값을 조절하여 클래스의 균형을 맞추어 학습하는 방법인 loss re-weighting[16] 등을 사용하여 long-tail 데이터 셋의 불균형을 해소하거나, 데이터 증강 기법을 사

용해 희귀 객체의 수를 증가시키는 방법을 사용한다.

2.3 데이터 증강

일반적으로, 데이터 증강은 학습 데이터 셋의 규모, 크기를 키우는 방법으로 다양한 컴퓨터 비전 과제에서 성능 향상에 도움이 된다. 기존 데이터에서 훈련 데이터 셋을 인위적으로 증가시키기 위해 다양한 변환을 사용한다. 일반적으로 이미지 수준 데이터 증강은 기하학적 변환 증강과 광학적 증강으로 나뉠 수 있다[17]. 기하학적 변환 증강은 flipping, scaling, cropping 등이 있고, 광학적 증강은 색상, 채도, 명도 조절 등이 있다.

객체들끼리 겹쳐서 탐지 성능을 저하시키는 문제를 해결하기 위해 Random Erasing[18], Hide-And-Seek[19], Cutout[20]과 같이 폐색 현상 기반 데이터 증강 기법들이 연구되었다.

2.4 Copy-Paste 데이터 증강

본 논문에서 연구하는 Copy-Paste 데이터 증강 기법은 객체 수준 데이터 증강 기법이다. 임의의 객체를 임의로 선택한 배경 이미지로 붙여 넣어 합성하는 방법으로 데이터를 증강시킨다. D. Dwibedi 등은 객체를 배치할 때, 전체적인 이미지의 특징보다는 객체가 위치할 영역과의 조화가 중요하다고 주장한다[8]. 이를 위해 gaussian blurring, poisson blending과 같은 기법을 사용하고 정답 외의 객체를 배치하는 방법으로 객체 붙여넣기 시 발생하는 경계 픽셀 불일치에 객체 탐지 모델이 집중하지 않게 한다. 본 논문에서는 D. Dwibedi 등이 연구한 "Cut, Paste and learn: Surprisingly easy synthesis for instance detection"[8]을 baseline으로 사용하고, 이후 baseline으로 표기한다. N. Li 등과 N. Dvornik 등은 전체적인 맥락을 파악하여 객체 배치 시 사용하여 탐지 성능을 향상시켰다[9-10]. G. Ghiasi 등은 전체적인 맥락과 지역적인 조화 없이도 large, standard scale jittering 정도만 사용하는 간단한 Copy-Paste로도 효과적임을 보였다[11]. N. Li 등은 도로 환경에서의 희귀 객체(교통 원뿔)에 대한 Copy-Paste 데이터 증강 기법을 연구하면서, 차선, 도로, 일반 객체(차량, 보행자, 등)와 같은 도로 환경 문맥을 객체의 크기와 채도 등의 변경과 객체의 배치에 이용하여 희귀 객체 탐지 성능을 향상시켰다[9]. 최근에는 적대적 생성 신경망(GAN)을 활용하여 Copy-Paste에 사용할 객체를 생성하는 방법이 연구되었다[21].

본 논문에서는 Copy-Paste 데이터 증강 기법을 이용하여 도로 환경에서의 쓰레기 객체 이미지 데이터를 생성하고 이를 학습 데이터로 사용하여 객체 탐지 모델을 학습하는 방법을 연구한다. 본 논문의 기여점은 다음과 같이 정리할 수 있다.

- 본 논문에서는 도로 환경에서의 원근 보정을 위해 원근 변환을 이용한 Copy-Paste 데이터 증강 기법을 제안한다.
- 시뮬레이션을 통해, 객체 탐지 모델의 학습 데이터로 사용하기 위해 도로 환경에서 쓰레기 객체(병, 캔, 박스 등)를 합성할 때, 제안한 Copy-Paste 데이터 증강 기법이 일부 객체들(원근 변환 적용 객체)의 탐지 성능 향상에 효과적임을 보인다.

Ⅲ. Copy-Paste 데이터 증강 기법 기반 도로 환경에서의 쓰레기 객체 데이터 합성 방법

그림 1은 본 논문에서 제안하는 Copy-Paste 데이터 증강 기법을 이용한 학습 데이터 생성 과정을 보여준다. 우선 객체와 객체의 마스크 이미지 그리고 배경 이미지를 수집한다. 깊이 추정 모델과 차선, 도로, 객체를 파악하는 panoptic driving perception 모델을 활용하여 배경 이미지에서 맥락 정보를 얻어낸다. 그 후, 앞서 얻어낸 맥락 정보를 활용하여 객체를 배치할 위치를 정하고 그에 맞춰 객체에 크기 변환과 원근 변환 등 지역 적응 변환을 가한다. 변환한 객체를 배경 이미지에 배치하여 데이터를 생성하고, 자동으로 주석을 추가한다.

3.1 객체 이미지 & 객체 마스크, 배경 이미지 수집

배경 이미지에 배치할 객체들은 다양한 종류의 쓰레기를 수집, 촬영하여서 객체 이미지를 확보한다. 확보한 객체 이미지는 객체의 특성에 따라 원근 변환을 적용할 객체와 적용하지 않을 객체로 분류한다. 원근 변환을 적용할 객체들은 2D에 가까워야 하기에 높이가 0.5cm 이하인 경우에 적용할 객체로 분류한다. 예를 들어, 도로 위에 배치될 병, 캔, 비닐, 전단지 등의 객체들을 확보하였고 비닐(일부), 전단지 등은 원근 변환을 적용할 객체로 분류된다. 수집된 객체는 segmentation 주석 처리하고 COCO API를 활용하여, 객체 마스크를 확보한다.

배경 이미지는 실제 도로를 촬영하여 확보하고, 깊이 추정 모델과 panoptic driving perception 모델을 사용하여 배경 이미지의 전역적인 맥락을 추출한다. 추출된 전역적인 맥락은 크기 변환, 원근 변환, 맥락을 고려한 배치 등에 사용된다.

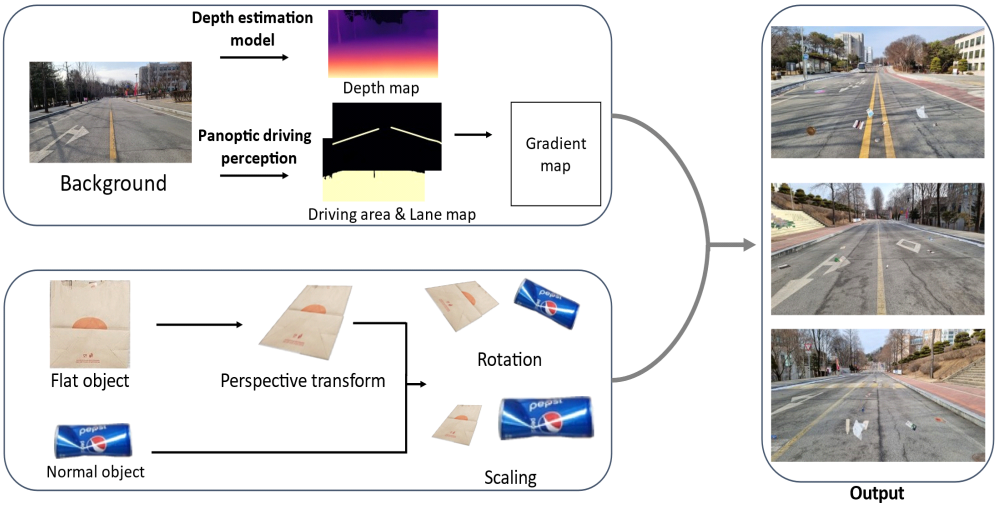

그림 1. 제안하는 학습 데이터 생성 과정
Fig. 1. Proposed train data generation process

3.2 객체 변환

확보한 객체를 이미지에 아무 작용 없이 배치하게 되면 물체 크기의 비율, 원근감 등이 시각적으로 현실과 동떨어지게 된다. 이를 객체에 다양한 변환을 적용해 해결한다.

3.2.1 크기 변환

도로 환경상에서 객체는 원근감에 의해 멀어질수록 이미지상에서 작아진다. 원근감을 고려하지 않고 단순히 배치하게 된다면, 시각적으로 굉장히 어색한 이미지가 생성된다. 이러한 원근감을 구현하기 위해 배경 이미지에 물체가 배치될 위치를 고려하여 크기 변환을 적용한다.

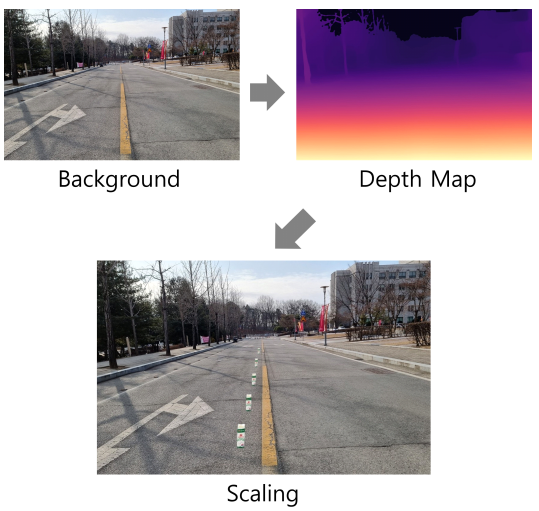

그림 2. 크기 변환 적용 과정
Fig. 2. Scale transform process

객체의 크기를 얼마만큼 변화시킬지에 대한 크기 변환 정도는 그림 2의 위의 두 그림에서 볼 수 있듯이 깊이 추정 모델의 결과로 나온 깊이 맵에 기반하여 정하게 된다. 깊이 맵의 각 픽셀은 해당 이미지에서 가장 먼 지점으로부터 각 픽셀의 지점까지의 상대적인 거리 정보를 포함하고 있다. 각 픽셀이 포함하고 있는 상대적인 거릿값을 0과 1 사이의 값으로 정규화하고 배치할 위치에서의 크기 변환 정도로 사용한다. 그림 2의 아래 그림에서 크기 변환이 잘 적용되었음을 볼 수 있다.

3.2.2 원근 변환

실제로 촬영한 도로 사진을 보거나, 차도에서 육안으로 차도를 보면 실제로는 일직선인 차선과 차도가 가운데로 기울어지고 또한 좁아지는 모습을 확인할 수 있다. 이러한 현상은 그림 3의 왼쪽 그림과 같이 바닥에 붙어 배치된 물체(특히 전단지, 비닐과 같이 납작한 물체)에도 동일하게 나타난다. 이 현상을 고려하지 않고 객체를 배치한다면 그림 3의 중간 그림과 같이 시각적으로 굉장히 어색한 이미지가 생성될 수 있으므로, 이러한 현상이 두드러지게 나타나는 객체를 배치할 때 동일한 현상이 나타나게끔 해주어야 한다. 본 논문에서는 이 현상을 구현하기 위해 해당하는 객체에 원근 변환을 적용하는 방법을 제안한다.

이때 실제 이미지의 차선 중 가운데에 위치한 차선은 가장자리에 위치한 차선에 비해 기울어지거나

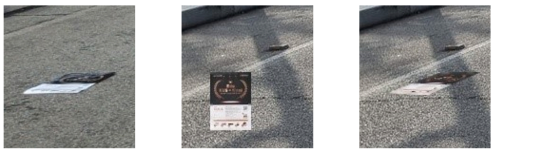

그림 3. 원근 변환 예시
Fig. 3. perspective transform example

좁아지는 현상이 적게 나타난다는 점을 고려해, 원근 변환은 객체가 배치될 이미지상의 위치에 따라 다르게 적용되어야 한다. 차선과 같은 위치에 배치되는 물체는 해당 차선의 기울기와 비슷한 기울기를 보이게 되고, 두 차선 사이에 물체가 배치되는 경우에 물체는 두 차선의 중간 정도의 기울기를 보인다.

위의 사항을 고려한 원근 변환 적용을 위해 배경 이미지에서 panoptic driving perception 모델을 이용해 확보한 차선 정보를 바탕으로 이미지의 픽셀마다 기울기 값을 가지는 배경 이미지와 같은 크기의 2차원 배열(기울기 맵)을 생성한다. 배경 이미지의 가장 왼쪽과 오른쪽 차선들의 연장선이 만나는 접점을 구하고, 각 픽셀에서 접점까지의 선의 기울기 계산해 기울기 맵을 생성할 수 있다. 이를 수식으로 나타내면 다음과 같다:

$$G = \begin{cases} 0, & if \quad x_C - x = 0, \\ -\dfrac{y_C - y}{x_C - x}, & else. \end{cases} \tag{1}$$

수식 (1)에서 G는 배경 이미지와 같은 크기의 2차원 배열인 기울기 맵이고, x_c, y_c는 접점의 픽셀 좌표, x, y는 기울기 맵의 픽셀 좌표다. 이때 픽셀의 좌표는 이미지의 좌 상단 꼭짓점이 (0, 0)인 것을 기준으로 한다. x_c - x가 0인 픽셀에서의 계산은 오류가 발생하기에 기울기 값을 0으로 예외 처리하고, 이후 선분과 y축과의 각도를 구할 때 선분의 기울기가 0인 경우 y축과의 각도를 0도로 설정한다. y_c - y가 0으로 실제 기울기 값 계산이 0인 픽셀들은 도로를 벗어나기에 객체가 배치되지 않으므로 추후 계산에 사용되지 않고, 오류가 발생하지 않는다.

원근 변환 행렬을 구하기 위해서는 원근 변환 전 객체 이미지의 네 꼭짓점의 픽셀 좌표와 변환 후의 네 꼭짓점의 픽셀 좌표가 필요하다. 객체 이미지의 좌우 하단 꼭짓점이 배치될 예정 위치의 기울기 값에 따라 객체 이미지의 왼쪽 면과 오른쪽 면을 회전시키고 회전시킨 후의 꼭짓점 좌표를 구한다. 이를 수식으로 나타내면 다음과 같다:

$$LeftTop(x_1', y_1')$$
$$= \begin{cases} x_1' = sin\,\theta_1 * h, y_1' = 0, & if \quad \theta_1 \geq 0, \\ x_1' = 0, y_1' = 0, & if \quad \theta_1 < 0, \end{cases}$$
$$RightTop(x_2', y_2')$$
$$= \begin{cases} x_2' = w + sin\,\theta_2 * h, y_2' = 0, & if \quad \theta_1 \geq 0, \\ x_2' = w - sin\,\theta_1 * h + sin\,\theta_2 * h, y_2' = 0, & if \quad \theta_1 < 0, \end{cases}$$
$$RightBottom(x_3', y_3')$$
$$= \begin{cases} x_3' = w, y_3' = cos\,\theta_2 * h, & if \quad \theta_1 \geq 0, \\ x_3' = w - sin\,\theta_1 * h, y_3' = cos\,\theta_2 * h, & if \quad \theta_1 < 0, \end{cases}$$
$$LeftBottom(x_4', y_4')$$
$$= \begin{cases} x_4' = 0, y_4' = cos\,\theta_1 * h, & if \quad \theta_1 \geq 0, \\ x_4' = -sin\,\theta_1 * h, y_4' = cos\,\theta_1 * h, & if \quad \theta_1 < 0, \end{cases} \tag{2}$$

수식 (2)에서 $LeftTop(x_1', y_1')$, $RightTop(x_2', y_2')$, $RightBottom(x_3', y_3')$, $LeftBottom(x_4', y_4')$은 각각 원근 변환 적용 후 객체 이미지의 좌 상단, 우 상단, 우 하단, 좌 하단 꼭짓점의 픽셀 좌표다. w, h는 원근 변환 전 객체 이미지의 폭과 높이고 단위는 픽셀이다. θ_1과 θ_2는 객체 이미지의 좌우 하단 꼭짓점이 배치될 예정 위치의 픽셀과 기울기 값을 이용해 만든 선분과 y축 사이의 각도다. 이때 픽셀의 좌표는 이미지의 좌 상단 꼭짓점이 (0, 0)인 것을 기준으로 한다.

원근 변환의 예시와 함께 수식 (2)의 기호들을 그림 4에서 보인다. 그림 4의 회색 사각형은 변환 전 객체 이미지, 주황색 평행사변형은 변환 후 객체 이미지를 나타낸다. (x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)은 각각 변환 전 객체 이미지의 좌 상단, 우 상단, 우 하단, 좌 하단 꼭짓점의 픽셀 좌표다. 마찬가지로 픽셀의 좌표는 이미지의 좌 상단 꼭짓점이 (0, 0)인 것을 기준으로 한다.

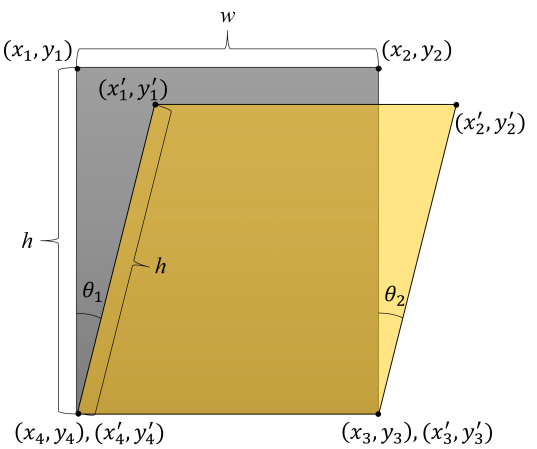

그림 4. 원근 변환 적용 도식
Fig. 4. perspective transform application schema

수식 (2)를 통해 원근 변환 적용 후의 객체 이미지의 네 꼭짓점의 좌표를 구하고 나면, 8개의 꼭짓점(원근 변환 이전 4개 + 원근 변환 이후 4개) 좌표를 이용하여 원근 변환 행렬을 구할 수 있다. 원근 변환 행렬을 구하고 이를 이미지에 적용하는 과정을 수식으로 표현하면 다음과 같다:

$$Perspective\ Transform\ Matrix = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix}, \quad (3)$$

$$\begin{bmatrix} x_1' \\ y_1' \\ x_2' \\ y_2' \\ x_3' \\ y_3' \\ x_4' \\ y_4' \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1' \cdot x_1 & -x_1' \cdot y_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -y_1' \cdot x_1 & -y_1' \cdot y_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2' \cdot x_2 & -x_2' \cdot y_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -y_2' \cdot x_2 & -y_2' \cdot y_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x_3' \cdot x_3 & -x_3' \cdot y_3 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -y_3' \cdot x_3 & -y_3' \cdot y_3 \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -x_4' \cdot x_4 & -x_4' \cdot y_4 \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -y_4' \cdot x_4 & -y_4' \cdot y_4 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{bmatrix}, \quad (4)$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \quad (5)$$

이때 x, y는 변환 전 이미지의 좌표이고, x', y'는 변환 후 이미지의 좌표이다.

수식 (3)과 (4)를 통해 구한 원근 변환 행렬과 수식 (5)를 이용하여 객체 이미지에 원근 변환을 적용해 그림 3의 오른쪽 그림과 같이 객체가 이미지에 배치될 위치에 따라 기울어지고 좁아지는 모습을 재현해낼 수 있고, 원근 변환이 적용되기 적합한 높이가 0.5 cm 이하인 납작한 객체들은 원근 변환을 적용하기 전보다 시각적으로 자연스러워 보인다. 또한 표 1에서 볼 수 있듯이 Copy-Paste 데이터 증강 기법에 원근 변환을 적용하여 객체 탐지 모델의 mean average precision을 49.1에서 63.1로 14.0 만큼 상승시켰다.

3.2.3 블렌딩

배경 이미지에 객체를 블렌딩 없이 바로 배치하게 되면 눈에 확 띄는 경계선이 발생한다. 객체 탐지 모델은 학습 시 배치한 물체보다는 이 경계선에 집중하게 되며 이는 실제 데이터에서 추론하였을 때 성능이 떨어지는 문제를 야기한다[8]. Baseline과 N.Li.등은 이를 해결하기 위해 gaussian blur와 poisson blending[22] 두 가지 기법을 사용한다[8-9].

하지만 poisson blending은 크기가 작은 객체에 적용될 때 ghost 현상을 일으킬 수 있다[23]. 본 논문의 실험에서는 대체로 작은 객체를 배경 이미지에 배치하기 때문에 원래의 색과 형태를 잃는 ghost 문제가 발생하여 모델의 성능을 감소시키므로 제외하였다.

Alpha blending: 주석을 이용해서 객체 마스크를 추출할 때 COCO API 객체 마스크 추출 라이브러리가 객체 마스크 이미지의 값을 정확히 0 또는 1로 분류하는 것이 아닌 경계선 근처에서는 0과 1 사이의 값을 가지면서 경계선 근처에 노이즈가 발생한다는 문제점이 있다. 이를 제거하지 않은 객체 마스크를 이용하여 객체를 배치하게 되면 객체 이미지의 경계선에도 노이즈가 심하게 발생한다.

노이즈를 제거하기 위해 배경 이미지와 객체 이미지의 각 픽셀과 객체 마스크의 각 픽셀을 요소 별로 곱한 뒤 합하는 Alpha blending 기법을 사용하였다. 수식으로 나타낸다면 다음과 같다:

$$I^{output} = (1 - \alpha)B + \alpha I. \quad (6)$$

여기서 B는 객체 이미지가 배치될 지역의 배경 이미지이고 I와 α는 각각 붙여 넣을 객체 이미지와 객체 마스크이다. 이러한 Alpha blending 기법을 사용하여 경계면 주위에 발생하는 노이즈 값을 제거하였다.

Gaussian blur: 배경 사진에 직접 이미지를 붙여 넣는다면 객체와 배경 이미지 사이의 경계가 너무 뚜렷해 보이는 현상이 발생한다. 육안상으로 확인해보면 큰 차이가 없어 보일 수도 있지만 객체 탐지 모델은 학습 중 이러한 경계면에 집중하여 학습하게 되므로

표 1. 여러 Copy-Paste 데이터 증강 기법으로 생성한 데이터를 이용해 학습한 객체 탐지 모델 간 성능 비교
Table 1. Performance comparison in object detection models trained by data generated by various Copy-Paste data augmentation method

Model	AP					mAP
	Bottle	Box	Can	Vinyl	Flyer	
Baseline[8]	26.5	56.8	38.8	72.8	8.1	40.6
Baseline+	35.3	69.7	**49.4**	76.4	14.5	49.1
Ours	**36.4**	**72.0**	48.7	**81.5**	**76.7**	**63.1**

탐지 성능이 감소하게 된다[8]. 따라서 경계면을 모호하게 하기 위한 방법으로 경계면에 대해서 gaussian blur를 적용하여 좀더 매끄럽게 처리하였다.

Distractor: 앞서 설명한 블렌딩 기법들을 사용하여 객체 이미지의 부자연스러운 경계선을 완화하더라도, 객체 탐지 모델이 블렌딩을 거친 경계선에 집중하여 학습하게 될 수 있다. 이를 해결하기 위해, 데이터셋의 정답 클래스에 포함되지 않는 객체를 distractor로 배경 이미지에 배치하여 경계면에 집중해 학습할 가능성을 낮췄다[8].

3.3 데이터 증강

객체의 시각적 다양성을 위해 몇 가지 데이터 증강 기법을 추가하였다.

3.3.1 회전

실제로 객체가 도로 환경에서 다양한 각도로 존재할 수 있는 상황을 고려하기 위해 객체를 2D 회전시켜 배치한다. 실내 주방 환경을 고려하여 -30도~30도 사이의 각도로 회전시켜 배치하는 baseline[8]과는 다르게 본 논문에서 고려하는 도로 환경에서의 쓰레기 객체는 실제로 어느 방향으로 배치되어 있어도 이상하지 않기에 객체를 360도 중 임의의 각도로 회전시켜 배치한다.

쓰레기 객체는 바라보는 방향에 따라 시각적으로 다른 모습을 보여줄 가능성이 높다. 이를 고려하기 위해 수집한 객체를 여러 방향에서 촬영해 3D 회전시킨 객체 이미지들을 확보하였다.

3.3.2 폐색/잘림 현상

실제 이미지에서는 객체들이 겹쳐 있는 폐색 상황이 발생하게 된다. 따라서 객체를 배치하는 상황에서 이러한 점을 고려하였고, 두 객체의 Intersection over Union이 0.5 이하가 되는 경우에만 폐색된 배치를 허용하였다.

또한 객체가 이미지의 가장자리에 위치하여 객체가 잘리게 되는 현상도 존재하는데, 객체

이미지의 최소 25% 이상은 남을 수 있게 객체를 배치하였다.

3.4 맥락을 고려한 배치

Copy-Paste 데이터 증강 기법을 사용할 때 시각적 맥락을 고려하여 배치하는 것이 중요하다[9-10]. 실제 이미지에서는 쓰레기가 하늘에 위치하거나 공중에 떠 있을 가능성은 없다. 또한 자율주행 노면 청소차의 경우에는 도로에서 임무 수행을 한다는 특징이 있다. 이러한 환경을 고려해 시각적 맥락을 이용하여 도로에 쓰레기를 배치해야 한다.

따라서 본 논문의 Copy-Paste 데이터 증강 기법에서는 그림 1에서 볼 수 있듯이 panoptic driving perception 모델을 이용해 도로와 도로 이외의 지역을 구분하고 시각적 맥락을 고려해 도로 위에만 쓰레기 객체를 배치한다.

Ⅳ. 시뮬레이션 결과

본 논문에서 제안하는 Copy-Paste 데이터 증강 기법의 효과를 보이기 위해, 기존의 Copy-Paste 데이터 증강 기법[8]을 사용하여 생성한 학습 데이터 셋과 본 논문의 Copy-Paste 데이터 증강 기법을 사용하여 생성한 학습 데이터 셋으로 각각 학습시킨 객체 탐지 모델의 성능을 비교한다. 먼저 실험 환경을 설명한다.

4.1 학습 데이터 셋 및 평가 데이터 셋

학습 데이터 셋 생성을 위해 5개 카테고리(병, 캔, 박스, 비닐, 전단지)의 객체 28개를 수집 후 촬영하였고, 객체의 높이가 0.5cm 이하면 원근 변환을 적용할 객체 이미지로 그렇지 않다면 적용하지 않을 객체 이미지로 나누었다.

배경 이미지는 고려대학교 세종캠퍼스 내의 도로를 총 15장 직접 촬영하였고, 전역적인 맥락을 파악하기 위해 YOLOP[24] 모델과 MiDaS[25] 모델을 이용하였다.

모든 실험에서 이미지 1장당 10개의 객체를 배치하여 5,000장의 학습 데이터 셋을 생성하였다. 또한 본 논문에서 제안한 원근 변환을 이용한 Copy-Paste 데이터 증강 기법이 객체 탐지 성능에 미치는 영향을 확인하기 위해 원근 변환의 사용 유무를 제외하고는 정확히 같은 위치에 같은 객체를 배치하여 데이터를 생성하였다.

평가 데이터 셋은 수집한 객체를 실제로 여러 도로에 무작위로 배치하고 촬영하여 총 205장 생성하였다.

4.2 객체 탐지 모델 및 하이퍼파라미터

본 논문에서는 YOLOv7[13] 모델을 학습시켜 성능을 확인하였다. 학습을 위해 COCO 데이터 셋에 사전 훈련된 최신 버전의 YOLOv7 모델 가중치를 사용하여 전이 학습을 진행하였고, 이를 Copy-Paste 데이터 증강 기법으로 생성한 도로 환경에서의 쓰레기 객체 학습 데이터 셋으로 미세 조정하여 학습하였다.

학습에는 Nvidia GeForce RTX 2080ti GPU 8개를 사용하였고, GPU당 배치 크기는 32, 학습을 위한

optimizer는 SGD, SGD의 모멘텀은 0.937, weight decay는 5e-4, 학습률은 0.01로 시작하고 One cycle 정책을 사용하였다. 모든 모델 학습은 같은 하이퍼파라미터를 사용하고 200 epoch 동안 미세 조정되었다.

4.3 평가지표 및 성능 평가

본 논문에서 비교를 위한 평가지표로는 기존에 객체 탐지 분야에서 주로 쓰이는 성능 평가지표인 카테고리별 Average Precision(AP)과 AP의 평균인 mean Average Precision(mAP)을 사용하였고, Intersection over Union(IoU) 임곗값은 0.5로 설정하였다.

본 논문에서 제안하는 Copy-Paste 데이터 증강 기법으로 생성한 학습 데이터로 학습한 모델을 평가하기 위해 기준 모델로 poisson blending을 제외한 baseline의 Copy-Paste 데이터 증강 기법[8]을 사용하여 생성한 데이터로 학습한 모델을 사용하였고 이 모델을 Baseline이라 표기하였다. 또한 도로 환경에서의 쓰레기 객체 합성이라는 환경을 고려하여 Baseline의 학습에 사용된 데이터 증강 기법에 크기 변환, 360도 회전, 맥락을 고려한 배치 기법을 추가하여 생성한 데이터로 모델을 학습해 Baseline과 비교하였고 이 모델을 Baseline+라 표기하겠다. 마지막으로 본 논문에서 제안한 원근 변환 기법을 추가한 Copy-Paste 데이터 증강 기법으로 데이터를 생성하여 모델을 학습하였고 이 모델을 Ours라 표기하였다. 4.1 절에서 언급한 바와 같이 제안한 원근 변환 기법의 효과를 확인하기 위해 원근 변환 기법 적용을 제외하고 객체의 배치와 사용된 기법 등은 정확히 같다.

실험 결과는 표 1에 정리되어 있다. 카테고리별 가장 높은 지표를 볼드체로 표시하였다. 크기 변환, 360도 회전, 맥락을 고려한 배치를 추가한 Baseline+의 mAP가 Baseline의 mAP와 비교해 40.6에서 49.1으로 8.5 만큼 증가한 것을 보아 Copy-Paste 데이터 증강 기법으로 도로 환경에서의 쓰레기 객체 데이터를 생성할 때 이러한 기법들이 효과적임을 확인하였다.

본 논문에서 제안한 원근 변환을 이용한 Copy-Paste 데이터 증강 기법이 효과적임을 Baseline+과 Ours의 성능 비교를 통해 확인할 수 있다. 원근 변환이 성능에 미치는 영향을 분석하기 위해 4.1 절에서 언급했듯이 두 모델의 학습 데이터는 원근 변환을 적용할 객체에 원근 변환 적용 유무를 제외하고는 배경, 객체, 배치 위치 등 다른 사항들은 정확히 같다.

원근 변환을 적용할 납작한 객체가 없는 병과 캔 카테고리의 AP는 Baseline+에서는 각각 35.3, 49.4이고 Ours에서는 36.4, 48.7로 차이는 각각 1.1, 0.7이다. 병과 캔 카테고리의 경우 원근 변환을 적용한 객체 이미지가 학습 데이터에 포함되지 않으므로 이러한 AP의 변화는 학습 편차에 의한 결과라고 할 수 있다. 또한 병 카테고리와 캔 카테고리의 AP가 다른 카테고리에 비해 낮은 모습을 볼 수 있는데 이는 각 객체들의 특성에 의해 발생한다. 병 카테고리에는 투명한 객체가 많이 존재하고 이러한 객체들은 실제 이미지에서는 배경에 의해 색과 모습이 잘 변하는 특성이 있다. 따라서 수집한 병 카테고리 객체와 실제 데이터 사이에 시각적인 차이가 발생할 수 있고 이는 탐지 성능의 저하로 이어진다. 캔 카테고리의 객체는 다른 카테고리의 객체에 비해 크기가 작다는 특성이 있다. 본 논문에서 사용한 Yolov7 모델은 작은 객체에 대한 탐지 성능이 비교적 떨어지기 때문에[13] 캔 카테고리의 탐지 성능이 낮아진다.

박스와 비닐 카테고리에는 원근 변환을 적용할 납작한 객체와 그렇지 않은 객체가 동시에 존재한다. 박스 카테고리는 박스 카테고리 객체 이미지의 약 14%, 비닐 카테고리는 비닐 카테고리 객체 이미지의 약 38%가 원근 변환을 적용할 납작한 객체로 이루어져 있다. 박스와 비닐 카테고리의 AP는 Baseline+에서는 각각 69.7, 76.4이고 Ours에서는 72.0, 81.5로 차이는 각각 2.3, 5.1로 Ours에서 더 높다. 또한 원근 변환을 적용할 객체들만 포함된 전단지 카테고리의 AP의 경우엔 Baseline+에서는 14.5이고 Ours에서는 76.7로 62.2 만큼 상승하였으며, mAP는 49.1에서 63.1로 14.0 상승하였다. 이를 통해, 높이가 낮은 객체를 도로 환경에서 Copy-Paste 데이터 증강 기법으로 합성할 때 본 논문에서 제안한 원근 변환 기법을 사용한 Copy-Paste 데이터 증강 기법이 효과적임을 알 수 있다.

V. 결 론

본 논문에서는 자율주행 노면 청소차의 쓰레기 객체 탐지 성능 향상을 위한 이미지 합성 기법을 연구하였다. 도로 환경에서의 쓰레기 객체 데이터 합성이라는 환경을 고려해 크기 변환, 360도 회전, 맥락을 고려한 배치와 같은 기법들을 추가하였고 실험을 통해 객체 탐지 모델의 성능을 향상시켰음을 보였다. 또한 원근 변환을 추가한 Copy-Paste 데이터 증강 기법을 제안하였고 도로 환경에서 높이가 낮은 납작한 객체들의 합성에 제안한 기법이 효과적임을 보였다.

하지만 병같이 투명한 물체에 대해서는 탐지 성능이 유독 낮음을 볼 수 있는데, 이를 해결하기 위해 향

후 투명한 물체에 채도 변환을 적용해보고, 기존의 poisson blending 기법을 발전시켜 작은 객체에도 효과적인 개선된 poisson blending 기법을 적용하여 객체 탐지 성능을 향상시키는 Copy-Paste 데이터 증강 기법을 연구할 예정이다.

References

[1] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances NIPS*, pp. 91-99, Montreal, Canada, Sep. 2007. (https://doi.org/10.48550/arXiv.1506.01497)

[2] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. ECCV 2016*, Springer, Cham, 2016. (https://doi.org/10.1007/978-3-319-46448-0_2)

[3] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one-stage object detection," in *Proc. IEEE Conf. ICCV*, pp. 9627-9636, 2019. (https://doi.org/10.48550/arXiv.1904.01355)

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. CVPR*, pp. 779-788, Las Vegas, USA, Jun. 2016. (https://doi.org/10.1109/CVPR.2016.91)

[5] Y. Movshovitz-Attias, T. Kanade, and Y. Sheikh, "How useful is photo-realistic rendering for visual learning?," in *Proc. ECCV 2016,* Springer, Cham, 2016. (https://doi.org/10.1007/978-3-319-49409-8_18)

[6] W. Chen, H. Wang, Y. Li, H. Su, Z. Wang, C. Tu, D. Lischinski, D. Coehn-Or, and B. Chen, "Synthesizing training images for boosting human 3D pose estimation," in *Fourth Int. Conf. 3D Vision(3DV)*, Oct. 2016. (https://doi.org/10.1109/3DV.2016.58)

[7] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *Proc. IEEE Conf. CVPR*, pp. 2315-2324, 2016. (https://doi.org/10.48550/arXiv.1604.06646)

[8] D. Dwibedi, I. Misra, and M. Hebert, "Cut, paste and learn: Surprisingly easy synthesis for instance detection," in *Proc. IEEE Conf. ICCV*, pp. 1301-1310, 2017. (https://doi.org/10.48550/arXiv.1708.01642)

[9] N. Li, F. Song, Y. Zhang, P. Liang, and E. Cheng, "Traffic context aware data augmentation for rare object detection in autonomous driving," in *2022 ICRA*, pp. 4548-4554, 2022. (https://doi.org/10.48550/arXiv.2205.00376)

[10] N. Dvornik, J. Mairal, and C. Schmid, "Modeling visual context is key to augmenting object detection datasets," in *Proc. ECCV*, pp. 364-380, 2018. (https://doi.org/10.48550/arXiv.1807.07428)

[11] G. Ghiasi, Y. Cui, A. Srinivas, R. Qian, T.-Y. Lin, E. D. Cubuk, Q. V. Le, and B. Zoph, "Simple copy-paste is a strong data augmentation method for instance segmentation," in *Proc. IEEE/CVF Conf. CVPR*, pp. 2918-2928, 2021. (https://doi.org/10.48550/arXiv.2012.07177)

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. CVPR*, pp. 580-587, 2014. (https://doi.org/10.1109/CVPR.2014.81)

[13] C.-Y. Wang, A. Bochkovshiky, and H. M. Liao, *Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors*(2022), from https://github.com/WongKinYiu/yolov7. (https://doi.org/10.48550/arXiv.2207.02696)

[14] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," in *IEEE Trans. Pattern Anal. and Mach. Intell.,* vol. 28, no. 4, pp. 594-611, 2006. (https://doi.org/10.1109/TPAMI.2006.79)

[15] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *Proc. IEEE/CVF Conf. CVPR*, pp. 9268-9277, 2019.

(https://doi.org/10.1109/CVPR.2019.00949)

[16] A. K. Menon, S. Jayasumana, A. S. Rawat, H. Jain, A. Veit, and S. Kumar, "Long-tail learning via logit adjustment," in *Conf. ICLR*, 2020. (https://doi.org/10.48550/arXiv.2007.07314)

[17] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," in *J. Big Data*, vol. 6, no. 1, pp. 1-48, 2019. (https://doi.org/10.1186/s40537-019-0197-0)

[18] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," in *Proc. AAAI Conf. Artificial Intell.*, vol. 34, no. 07, pp. 13001-13008, 2020. (https://doi.org/10.1609/aaai.v34i07.7000)

[19] K. K. Singh and Y. J. Lee, "Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization," in *2017 IEEE ICCV,* pp. 3544-3553, 2017. (https://doi.org/ 10.1109/ICCV.2017.381)

[20] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," *arXiv preprint arXiv:1708.04552*, 2017. (https://doi.org/10.48550/arXiv.1708.04552)

[21] L. Su-a and H. Ji-hyeong, "Copy-paste based image data augmentation method using," *J. KIISE*, vol. 49, no. 12, pp. 1056-1061, 2022. (https://doi.org/10.5626/JOK.2022.49.12.1056)

[22] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," *ACM SIGGRAPH 2003*, pp. 313-318, 2003. (https://doi.org/10.1145/1201775.882269)

[23] S. R. Klomp, D. W. J. M. van de Wouw, and P. H. N. de With, "Real-time small-object change detection from ground vehicles using a siamese convolutional neural network," *J. Imaging Sci. and Technol.*, pp. 60402-1-60402-16, 2019. (https://doi.org/10.2352/J.ImagingSci.Technol.2019.63.6.060402)

[24] D. Wu, M. W. Liao, W. T. Zhang, X. G. Wang, X. Bai, W. Q. Cheng, and W. Y. Liu, "Yolop: You only look once for panoptic driving perception," in *Mach. Intell. Res.*, vol. 19, pp. 550-562, 2022. (https://doi.org/10.1007/s11633-022-1339-y)

[25] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," in *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. 44, no. 3, pp. 1623-1637, 2020. (https://doi.org/10.1109/TPAMI.2020.3019930)

배 장 훈 (Jang Hoon Bae)

2022년 2월 : 고려대학교 세종캠퍼스 컴퓨터정보학과 학사 졸업

2022년 3월~현재 : 고려대학교 컴퓨터정보학과 석사과정

<관심분야> Lightweight Deep Neural Network

[ORCID:0009-0001-9095-1613]

박 병 준 (Byoung Jun Park)

2017년 3월~현재 : 고려대학교 세종캠퍼스 컴퓨터융합소프트웨어학과 학사과정

<관심분야> Deep Learning, Pose Estimation, Action Recognition

최 인 우 (In U Choi)

2018년 3월~현재 : 고려대학교 세종캠퍼스 컴퓨터융합소프트웨어학과 학사과정

<관심분야> Deep Learning, Image Processing

김 재 원 (Jaewon Kim)

2021년 3월~현재 : 고려대학교 세종캠퍼스 컴퓨터융합소프트웨어학과 학사과정

<관심분야> Object Detection, Robot Operating System

김 민 호 (Minhoe Kim)

2012년 2월 : KAIST 전기 및 전자 공학과 학사 졸업

2014년 2월 : KAIST 전기 및 전자 공학과 석사

2018년 2월 : KAIST 전기 및 전자 공학과 박사

2018년~2019년 : EURECOM, France, 박사후 연구원

2021년 3월~현재 : 고려대학교 컴퓨터정보학과 교수

<관심분야> 인공지능기반 무선통신, 연합학습, 분산 기계 학습