https://doi.org/10.21289/KSIC.2021.24.5.547 KSIC

자율주행을 위한 동적 객체 인식 방법에 관한 연구

A Study on the Motion Object Detection Method for Autonomous Driving

박승준[1*], 박상배[2], 김정하[3]

Seung-Jun Park[1*], Sang-Bae Park[2], Jung-Ha Kim[3]

〈Abstract〉

Dynamic object recognition is an important task for autonomous vehicles. Since dynamic objects exhibit a higher collision risk than static objects, our own trajectories should be planned to match the future state of moving elements in the scene. Time information such as optical flow can be used to recognize movement. Existing optical flow calculations are based only on camera sensors and are prone to misunderstanding in low light conditions. In this regard, to improve recognition performance in low-light environments, we applied a normalization filter and a correction function for Gamma Value to the input images. The low light quality improvement algorithm can be applied to confirm the more accurate detection of Object's Bounding Box for the vehicle. It was confirmed that there is an important in object recognition through image prepocessing and deep learning using YOLO.

Keywords : Deep Learning, Faster R-CNN, Machine Learning, Support Vector Machine, Object Detection, Unmanned Vehicle, YOLO

1* 국민대학교 자동차공학전문대학원 대학원생
E-mail: psjun72@gmail.com

2 폴리텍대학교 청주캠퍼스 교수
E-mail: sangbae81@gmail.com

3 국민대학교 자동차IT융합학과 교수
E-mail: jhkim@kookmin.ac.kr

1* Department of Unmanned Vehicle Research Laboratory, Graduate School of Automotive Engineering, Kookmin University, 77 Jeongneung-Ro Seongbuk-Gu, Seoul, 02707, Korea

2 Polytechnics University, Cheongju campus, Korea

3 Department of Automobile and IT Convergence, Kookmin University, 77 Kookmin University, 77 Jeongneung-Ro Seongbuk-Gu, Seoul, 02707, Korea

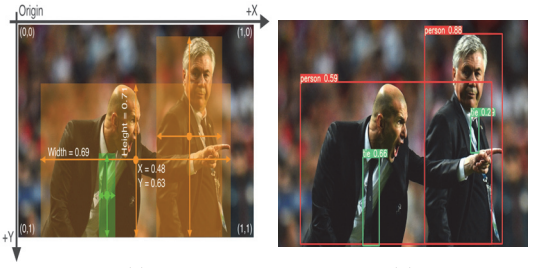
1. 서 론

차량의 주행 중 수행되어야 하는 이미지 기반의 객체 검출 및 분류에 대한 연구가 활발히 진행되고 있다. 그 중 최근 가장 좋은 성과를 보이는 분야는 CNN(Convolutional Neural Network)기반의 딥러닝이며, 분류와 위치 추정을 동시에 수행하는 네트워크의 개발로 인하여 객체 검출의 성능이 높아지고 있다[1].

YOLO(You Only Look Once) 알고리즘은 실시간으로 물체를 감지하기 위해 CNN을 사용한다. 알고리즘은 물체를 감지하기 위해 신경망을 통한 단 한번의 순전파(Forward Propagation)를 필요로 하고 전체 이미지의 예측은 단일 알고리즘 실행에서 수행된다. CNN은 다양한 클래스 확률과 경계 상자를 동시에 예측하는 데 사용된다.

YOLO알고리즘은 다음과 같은 3가지 이유로

- 속도 : 실시간으로 물체를 예측할 수 있기 때문에 탐지 속도가 향상됩니다.
- 높은 정확도 : YOLO는 최소한의 배경 오차만으로 정확한 결과를 제공하는 예측 기법입니다.
- 학습 기능 : 객체 표현을 학습하고 객체 검출에 적용할 수 있는 탁월한 학습 능력을 가지고 있습니다[2].



(a) (b)

Fig. 1 YOLOv5 (a) bounding box (b) detection results according to training [2]

본 연구에서는 저조도 환경(터널, 해질녘 등)에서도 객체를 검출할 수 있도록 저조도 상황에서의 입력 이미지를 개선하는 작업을 진행하였으며 이후 Fig. 1에서 같이 yolo를 이용하여 결과를 도출하였다.

2. 종래의 기술

2.1 Color Space 구조

색의 세가지 속성인 색상 (Hue), 명도 (Lightness), 채도 (Chroma)를 3차원 공간의 축으로 형성된 것이 색 공간 (Color Space)이다. 색상 (Hue)은 색상 각 (Hue Angle)으로 표현 했을 때, 0 ° ~ 360 ° 의 범위를 가지며, 시계 방향으로 변화된다. 또한 색 공간은 대응 색 (Opponent Color) 관계를 쉽게 나타낸다. 대응 색 관계란 명도 축을 기준으로 대칭의 위치에 있는 두 색의 관계를 말하며 서로 보색 관계에 있다. 모든 색들의 밝고 어둠을 나타내는 명도 (lightness)는 색 공간을 지구로 비유할 경우 남극과 북극을 연결하는 축으로서 남극을 검정색, 북극을 흰색으로 하며 그 사이에는 회색들로 배열된다. 모든 색들의 깨끗한 정도를 나타내는 채도 (Chroma, Saturation)는 색 공간의 명도 축을 0으로 하고 적도에 가까이 갈수록 커진다[3]. 본 연구에서 사용되는 색 공간 RGB, HSV, HLS 색 공간이다. RGB 색 공간은 색을 혼합하면 명도가 올라가는 가산 혼합 방식으로 색을 표현한다. RGB 가산 혼합의 삼원색은 빨강 (Red), 녹색 (Green), 파랑 (Blue)을 뜻한다[4]. RGBA는 RGB와 동일하며, 알파 (Alpha) 라는 투과도를 덧붙인 것이다. Fig. 2의 그림에서 보는 바와 같이, RGB 색 공간은 삼원색에 해당하는 세가지 채널의 밝기를 기준으로 색을 지정한다. RGB 색 공간은 웹

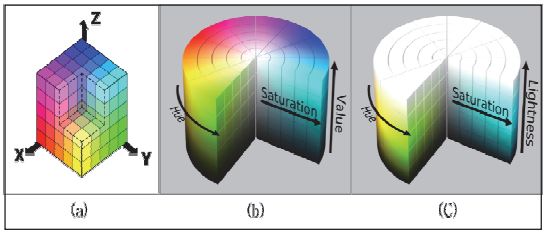


Fig. 2 Color space classification, (a) RGB space structure, (b) HSV space structure, (C) HLS space structure

색상 표현의 기본 원리이다. HSV 색 공간은 색상 (Hue), 채도 (Saturation), 명도 (Value)를 기준으로 색을 구성하는 방식이다. 감산 혼합이나 가산 혼합보다 색상의 지정이 직관적이기 때문에 시간 예술에 자주 쓰인다[5]. HLS는 HSV와 비슷하나, Value 대신 밝기 (Lightness) 선분이 들어간다.

2.2 카메라 캘리브레이션

카메라 캘리브레이션은 영상처리, 컴퓨터 비전 분야에서 번거롭지만 필요한 과정 중의 하나이다. 값싼 소형 카메라는 이미지의 왜곡을 발생시킨다. 주요한 왜곡은 방사 왜곡과 탄젠티얼(Tangential) 왜곡이 있다. 방사 왜곡으로 인해, 반듯한 형상이 휘어지게 됩니다. 이런 현상은 이미지의 중심으로부터 멀어질수록 심해진다. 카메라 영상은 3차원 공간상의 점들을 2차원 이미지 평면에 투사(Perspective Projection) 함으로써 얻어진다. 핀홀(Pinhole) 카메라 모델에서 이러한 변환 관계는 Fig. 3에서 보는 바와 같이, 모델링 된다. 핀홀 카메라 모델은 Fig. 3에서 보는 바와 같이, 하나의 바늘구멍(Pinhole)을 통해 외부의 상이 이미지로 투영된다는 모델이다. 이 때, 이 바늘 구멍(Pinhole)이 렌즈 중심에 해당되며 이곳에서 뒷면의 상이 맺히는 곳까지의 거리가 카메라 초점거리이다[6].

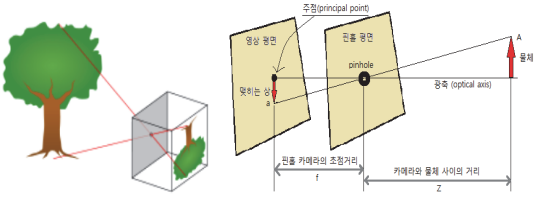


Fig. 3 Pinhole camera model

수식(1)에서, (X, Y, Z)는 월드 좌표계(World Coordinate System) 상의 3D 점의 좌표, [R | t]는 월드 좌표계를 카메라 좌표계로 변환 시키기 위한 회전 / 이동 변환 행렬이며 A는 Intrinsic Camera Matrix이다.

$$s\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & skew_c f_x & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = A[R \mid t] \begin{bmatrix} X \\ Y \\ Z \\ \end{bmatrix} \quad (1)$$

수직적으로 보면 카메라 캘리브레이션 (Camera Calibration)은 Fig. 4에서 보는 바와 같이, 3D 공간 좌표와 2D 영상 좌표 사이의 변환 관계 또는 이 변환 관계를 설명하는 파라미터를 찾는 과정이다[7].

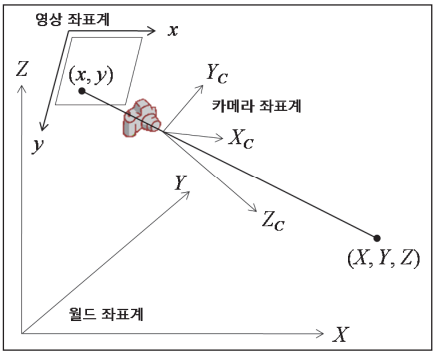


Fig. 4 Transformation relationship between 3D spatial coordinates and 2D image coordinates

수식(1)에서 [R | t]를 카메라 외부 파라미터(Extrinsic Parameter), A를 내부 파라미터(Intrinsic Parameter)라고 부른다. 그리고 A와 [R | t]를 합쳐서 Camera Matrix 또는 Projection Matrix라 부른다. 카메라 외부 파라미터는 카메라의 설치 높이, 방향(팬, 틸트) 등 카메라와 외부 공간과의 기하학적 관계에 관련된 파라미터이며, 내부 파라미터는 카메라의 초점 거리, Aspect Ratio, 중심점 등 카메라 자체의 내부적인 파라미터를 의미한다. 수식(1)에서 f_x, f_y 는 카메라 내부 파라미터(Intrinsic Parameters)로 초점거리(Focal Length)를 의미한다. 그리고 c_x, c_y는 주점(Principal Point)를 의미한다. 주점이란 카메라 렌즈의 핀홀에서 이미지 센서에 내린 수선의 영상 좌표를 의미한다. 비대칭계수를 의미하는 $skew_c$는 Fig. 5(b)에서 보는 바와 같이, 이미지 센서의 Cell Array의 y축이 기울어진 정도를 나타내며, 를 의미한다. 디지털 카메라 등에서 초점거리는 mm 단위로 표현되지만 Fig. 5(a) 에서 보는 바와 같이, 카메라 모델에서 말하는 초점거리(f)는 픽셀(pixel) 단위로 표현된다.

초점으로부터 거리가 1(Unit Distance)인 평면을 Normalized Image Plane이라고 부르며 이 평면상의 좌표를 보통 Normalized Image Coordinate라고 부른다. 이것은 실제는 존재하지 않는 가상의 이미지 평면이다. Fig. 6에서 보는 바와 같이, 카메라 좌표계 상의 한 점 (X_c, Y_c, Z_c)를 영상 좌표

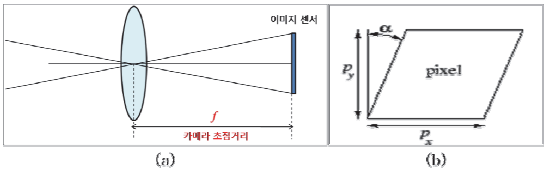


Fig. 5 Camera model, (a) Definition of camera focal length, (b) Cell array of image sensors

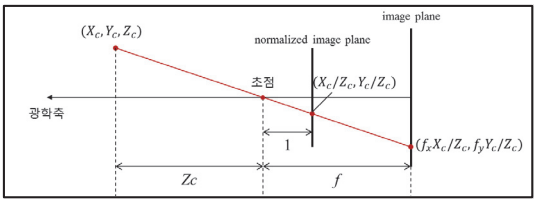


Fig. 6 Camera projection model

계로 변환할 때 먼저 (X_c, Y_c)를 Z_c(카메라 초점에서의 거리)로 나누는 것은 이 Normalized Image Plane 상의 좌표로 변환하는 것이며, 여기에 다시 초점거리 f를 곱하면 우리가 원하는 이미지 평면에서의 영상 좌표(pixel)가 나온다. 그런데, 이미지에서 픽셀 좌표는 이미지의 중심이 아닌 이미지의 좌 상단 모서리를 기준 원점으로 하기 때문에 실제 최종적인 영상 좌표는 여기에 (c_x, c_y)를 더한 값이 된다[7].

3. 동적 객체인식을 위한 이미지 처리

총 5단계로 구성되며 아래 단계에서 보는 것과 같이 구동하게 된다.

단계 1. 수식(2)에서 보는 것과 같이, BGR의 Color Space를 YUV Color Space로 변경한다. Y는 컬러의 밝기 정보 (Brightness or Luminance)를 가지고 있으며 U, V는 색 정보 (Chrominance)를 가지고 있다. 수식(2) 에서 보는 것과 같이, Y의 값은 R(Red), G(Green), B(Blue) Channel의 가중치에 대한 합이며 U, V는 Y와 R, B Channel의 차이 값에 대한 스케일링으로 계산할 수 있다.

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2)$$

단계 2. 입력된 이미지가 저조도 이미지 인지

판단하기 위하여 Fig. 1과 수식(3)에서 보는 바와 같이, 전체 이미지의 각 Pixel에 대한 Brightness Value (Y_k)를 이용하여 Brightness 평균(Y_{mean})을 구하게 된다. 그리고 Y_{mean} 값이 40.0보다 작으면 저조도 이미지로 판단을 내리게 된다. 40.0의 숫자는 사람의 눈으로 이미지를 확인했을 때 이미지가 어두워서 물체를 잘 구별할 수 없는 상태이다. 'k' 값은 Image의 Pixel 위치이며, 전체 이미지의 위치 정보를 가지고 있다. 그리고 'w' value는 입력 이미지의 Width(Image width)를 의미하며, 'h'는 입력 이미지의 Height (Image height)를 의미한다.

$$Y_{mean} = mean(\sum_{k=1}^{n=w\times h} Y_k : Brightness\ Value) \quad (3)$$

단계 3. 수식(4)에서 보는 것과 같이, Y_{mean} 이 '40.0' 보다 작은 저조도 이미지일 경우, 전체 Pixel에 Gamma Correction를 적용하기 위하여 수식(5)서 보는 것과 같이, Lookup Table()을 생성하여 각 Pixel 값에 대한 Gamma Value를 Update 해주게 된다. 수식(4)의 lim 함수는 각 Pixel의 값이 0보다 작아지거나, 255보다 커지는 경우를 방지하는 역할을 한다. 이 함수가 필요한 이유는 밝기 조절 결과값이 0보다 작거나 255보다 큰 값이 될 경우, BYTE 타입의 변수 $Dstimg_k$에 예상치 않은 값이 저장될 수 있기 때문이다.

$$\text{if}(x) = \begin{cases} x = True, & Y_{mean} < 40.0 \\ x = False, & Y_{mean} \geq 40.0 \end{cases} \quad (4)$$

$$\begin{cases} \Gamma Correction = \frac{1}{3}, \quad w = Img_{width}, \ h = Img_{height} \\ lookupTable_i = \left(\frac{1}{255^{\Gamma Correction}}\right) \times 255 (i = 1 \sim 255) \\ LUT = lookupTable_{Dstimg_k} \quad (k = 1 \sim w \times h) \\ Dstimg_k = LUT \times 255 + 0.5 \\ Dstimg_k = static_{case} \langle BYTE \rangle (\lim(Dst\_img_k)) \end{cases} \quad (5)$$

이미지의 Gamma Correction은 영상이나 이미지의 명암을 보정하기 위해 사용하는 알고리즘으로 밝거나 어둡게 보이는 영상의 Gamma를 조정하여 명암을 변경하게 된다.

단계 4. 본 연구에서 Gamma Correction을 적용하면서 발생한 노이즈를 Edge는 보존하면서 노이즈를 제거하기 위하여 Bilateral Filter를 적용하였다. Bilateral Filter는 Edge를 보존하면서 노이즈를 제거하는 비선형 필터이다. 단계 3와 단계 4는 이 '40.0' 보다 작은 저조도의 이미지일 경우에만 수행하게 된다.

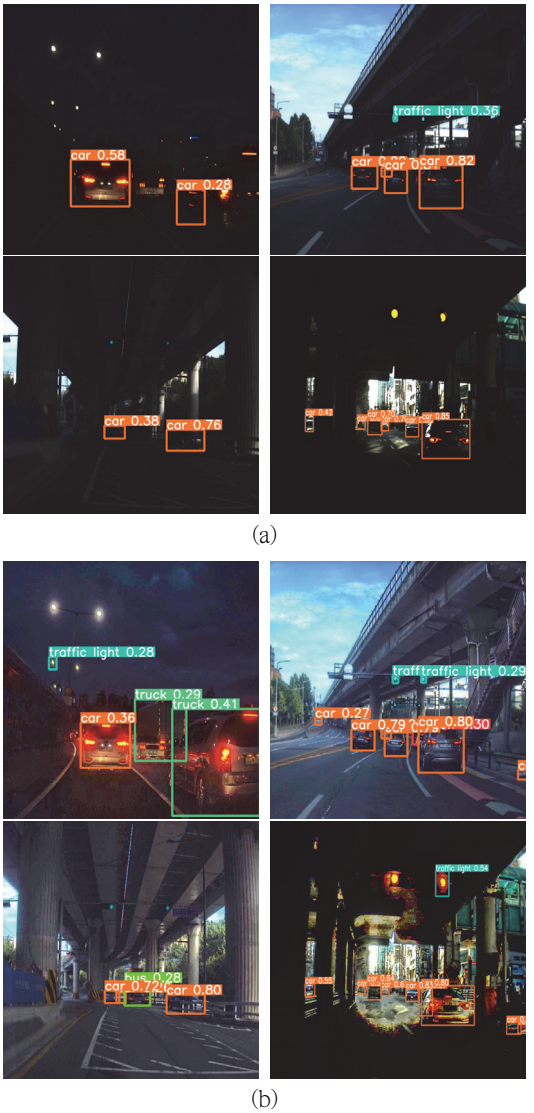
단계 5. 입력된 이미지 전체를 선명하게 만들기 위하여 Image Sharpening Filter를 적용한다. Sharpening Filter를 적용하기 위해서는 kernel()이 필요하며 본 연구에서는 수식(6)에서 보는 것과 같이, Kernel을 만들어 적용하였다.

$$Sharpen_{\text{kernal}} = \begin{bmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & 2 & 2 & 2 & -1 \\ -1 & 2 & 8 & 2 & -1 \\ -1 & 2 & 2 & 2 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{bmatrix} / 8.0 \quad (6)$$

Filter는 수식(7)에서 보는 것과 같이, Kernel을 이동하면서 입력 이미지 영역과 곱하여 그 결과값을 이미지의 해당 위치의 값으로 하는 선명한 이미지를 만드는 연산을 하게 된다. 기호 '*'는 Convolution을 의미하며, 원본 이미지의 (x,y) 위치의 명도를 f(x,y), kernel 이미지를 h(x,y) , 필터링된 결과를 g(x,y)라고 정의하였다. 그리고 본 식에서 'K'는 필터 크기의 절반을 의미한다. 도출된 결과 이미지는 크기를 절반으로 Resize하였다. 그 이유는 학습 모델에 사용되는 입력 이미지의 크기를 줄임으로써 알고리즘의 수행 시간을 크게 감소시키기 위함이다[8].

$$\begin{aligned} g(x,\ y) &= f * h \\ &= \sum_{u=-K/2}^{K} \sum_{v=-K/2}^{K} f(x+u,\ y+v) \times h(u,\ v) \end{aligned} \quad (7)$$

총 5단계의 전처리 과정을 수행하면 저조도 이미지에서 학습 모델이 Object를 추가적으로 검출하는 것을 Fig. 7(b)의 그림에서 확인할 수가 있다. 본 연구에서 저조도 이미지의 판단은 입력 이미지의 전체 Brightness 값이 평균 40 Level이하일 때 저조도 입력 이미지라고 판단하였다. 40 Level은 Heuristic Value로 실험에 의해 결정된 값이다. Fig. 7(a)는 전처리 과정을 수행하지 않고 다 객체 검출 알고리즘을 수행한 결과이다. 그리고 Fig. 7(b)는 전처리 과정 5단계를 모두 수행한 후 다 객체 검출 알고리즘을 수행했을 때의 결과이다. Fig. 7(a)와 (b)를 비교해 보면 전처리 과정을 수행한 알고리즘에서 저 조도로 인해 검출되지 않았던 차량들을 추가적으로 검출한 것을 확인할 수가 있다. 그리고 차량에 대한 Object의 Bounding Box를 더욱 정확하게 검출하는 것을 확인할 수가 있다. Fig. 7의 그림들은 본 문서 안에서 전체적으로 밝기 (Brightness) 40% 상승 시키고 대비 (Contrast)를 20% 상승 시켜서 표현하였다. 그 이유는 검출된 ROI를 그림에서 잘 확인할 수 있도록 하기 위함이다. 따라서 본 문서에서 보이는 이미지 보다 더 어두운 환경에서 알고리즘이 수행되었다.



(a)

(b)

Fig. 7 Object detection result after low light image correction. (a) YOLOv5 (b) Vehicle and traffic lights was detected additionally in dark environment after low light image correction using YOLOv5 and pre-processing

4. 결 론

본 연구에서는 자율주행차량과 지능형 자동차에서 필수적인 카메라 센서를 이용하여 이미지 처리를 진행하였으며 추후에는 저조도에서 좀 더 정확한 결과 값을 얻을 수 있는 방안에 대해서도 검토가 필요해 보인다. LiDAR 센서는 자체 방출된 레이저의 ToF를 측정하기 때문에 조도와 무관한 결과값을 얻을 수 있다. 추후 검토에서는 카메라와 LiDAR 센서 모두에서 움직임 정보를 갭쳐하여 조도가 낮은 조건에서 동적 객체 인식을 위한 실시간 알고리즘을 검토해 보겠다.

참고문헌

[1] 한국산업융합학회, 자율주행을 위한 딥러닝 기반의 차선 검출 방법에 관한 연구. https://doi.org/10.21289/KSIC.2020.23.6.979

[2] YOLOv5, https://github.com/ultralytics/yolov5/wiki/Train-Custom-Data

[3] Wikipedia,"Color space", https://en.wikipedia.org/wiki/Color_space

[4] R. W. G. Hunt, The Reproduction of Colour in Photography, Printing & Television, 5th Ed. Fountain Press, England, 1995. ISBN 0-86343-381-2

[5] Mark D. Fairchild, Color Appearance Models, Addison-Wesley, Reading, MA (1998). ISBN 0-201-63464-3

[6] Wikipedia, "Pinhole camera model", http://en.wikipedia.org/wiki/Pinhole_camera_model

[7] 카메라 캘리브레이션 (Camera Calibration)", 영상처리 2013. 2. 1, https://darkpgmr.tistory.com/32

[8] SanBae Park. (2020). Development of Artificial Intelligence base Multi-Object Recognition System for Urban Autonomous Driving. phD . from Graduate School of Automotive Engineering, KOOKMIN University, Seoul, Korea.

(접수: 2021.08.11. 수정: 2021.08.30. 게재확정: 2021.09.03.)