



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

碩士學位論文

딥러닝 기반 기상 악화 대응 보행자 검출
시스템 설계 및 구현

嘉泉大學校 大學院
모바일소프트웨어學科
모바일소프트웨어학 專攻
咸 旻 廷

碩士學位論文

딥러닝 기반 기상 악화 대응 보행자 검출
시스템 설계 및 구현

Design and Implementation of
Pedestrian Detection System for Weather Deterioration
Based on Deep Learning

嘉泉大學校 大學院
모바일소프트웨어學科
모바일소프트웨어학 專攻
咸 旻 廷

碩士學位論文
指導教授 朴石千

딥러닝 기반 기상 악화 대응 보행자 검출 시스템 설계 및 구현

Design and Implementation of
Pedestrian Detection System for Weather Deterioration
Based on Deep Learning

위 論文을 모바일소프트웨어학 碩士學位 論文으로 제출함.

2017年 12月 日

嘉泉大學校 大學院
모바일소프트웨어學科
모바일소프트웨어학 專攻
咸 旻 廷

이 論文을 咸旻廷의
工學碩士 學位論文으로 認准함

2017年 12月 日

審査委員長 _____ (印)

審査委員 _____ (印)

審査委員 _____ (印)

요 약

최근 자율주행 자동차 시스템에 대한 관심이 증가하면서 자율주행 자동차의 안전을 책임지는 ADAS(Advanced Driver Assistance System)는 각종 위협이나 사고에 대비해야 한다. 특히 자율주행 시 보행자 사고는 치명적으로 생명과 직결되어 큰 사고로 이어질 수 있다. 최근 5년간 교통안전공단이 분석한 결과 안개 낀 날에 보행자 교통사고 치사율이 맑은 날에 비해 5.3배 더 높으며 이는 안개 낀 날 및 흐린 날의 기상상태에 따른 시야 미확보가 안전운행에 가장 큰 위험요인으로 작용한 것으로 분석되어진다. 이에 본 논문에서는 자율주행중 기상악화로 인한 교통사고 치사율을 감소하기 위한 딥러닝 기반 기상 악화대응 보행자 검출 시스템을 제안한다.

본 논문에서 제안하는 시스템을 설계하기 위하여 먼저 영상 안개 유무 확인 후 안개 노이즈를 보정하였다. 이후 YOLO(You Only Look Once) 알고리즘을 수정하여 입력받는 영상 데이터에서 보행자 영역 특징을 추출하고 보행자 이미지 선행학습 데이터를 기반으로 보행자 분류하는 알고리즘을 통해 응답시간은 줄이고 검출률을 높이는 시스템을 설계했다.

본 논문에서 제안하는 시스템을 구현하기 위해 Windows 10에서 C, C++, Python2를 이용하였고 보행자 영역 추출에 OpenCV, CUDA SDK8.0 오픈소스 라이브러리를 활용하여 구축하였다. 먼저 보행자 학습용 Dataset을 CNN(Convolutional Neural Network)으로 선행학습을 시켰으며, 보행자 검출용 영상을 선택하고 업로드 할 수 있는 웹페이지를 구현하였다. 또한, 윈도우 기반의 보행자 검출 서버로 스트리밍 받아 Dark Channel Prior(DCP)를 활용하여 안개의 유무를 판단하고 영상 화소별 보

정을 하여 기상 악화에 대응한 영상 전처리를 한다. 그리고 영상은 딥러닝 기반 검출 시스템 모듈로 전송하여 영상 특징을 추출하고 기존의 선행 정보를 이용하여 클래스를 분류하고 보행자를 검출하도록 시스템을 구현하였다.

본 논문에서 제안하는 시스템을 평가하기 위해 기존의 기상 악화 대응 보행자 검출 시스템과 비교 평가 수행하였다. 테스트 결과 제안하는 시스템은 기존의 연구에 비하여 검출률은 15.52% 높고 미검출률은 15.06% 감소하였으며 응답시간은 240.12ms 단축시켰음을 확인하였다.

약 어 표

ADAS	Advanced Driver Assistance System
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
DCP	Dark Channel Prior
DBN	Deep Belief Network
DR	Detection Rate
DFD	Data Flow Diagram
DNN	Deep Neural Network
HOG	Histograms of Oriented Gradients
HTTP	Hyper Text Transfer Protocol
MR	Missing Rate
R-CNN	Region with Convolutional Neural Network
RNN	Recurrent Neural Network
RoI	Region of Interest
SVM	Support Vector Machine
YOLO	You Only Look Once

목 차

I. 서 론	1
II. 관련연구	4
2.1 안개 영상 노이즈 처리	4
2.1.1 안개 영상의 물리적 모델	4
2.1.2 Dark channel prior를 이용한 안개 제거	5
2.1.3 Dark channel prior기반 안개 유무 판단	9
2.2 딥러닝	10
2.2.1 인공신경망	10
2.2.2 딥러닝의 정의	13
2.2.3 딥러닝의 알고리즘	16
2.2.4 합성곱 신경망	17
2.3 보행자 검출 시스템	21
2.3.1 보행자 인식 시스템 구조	22
2.3.2 보행자 검출 알고리즘	23
2.4 기상 악화 대응 보행자 검출 시스템 제안	27
III. 기상 악화 대응 보행자 검출 시스템 설계	30
3.1 기상 악화 대응 보행자 검출 시스템 개요	30
3.2 기상 악화 대응 보행자 검출 시스템 요구사항	32
3.3 기상 악화 대응 보행자 검출 시스템 시퀀스 다이어그램	33

3.4 기상 악화 대응 보행자 검출 시스템 데이터 흐름도	34
3.5 기상 악화 대응 보행자 검출 시스템 알고리즘 설계	35
3.5.1 기존 시스템 알고리즘	35
3.5.2 딥러닝 기반 시스템 알고리즘	36
3.5.3 제안하는 안개 제거 기법	38
3.5.4 제안하는 YOLO 기반 보행자 검출 방법	40
 IV. 기상악화 대응 보행자 검출 시스템 구현 및 평가	42
4.1 구현 환경	42
4.2 기상 악화 대응 보행자 검출 시스템 구성	42
4.3 기상 악화 대응 보행자 검출 시스템 구현	43
4.4 기상 악화 대응 보행자 검출 테스트	48
4.4.1 테스트 환경	48
4.4.2 기존 기상 악화 보행자 검출 테스트	48
4.4.3 테스트 시나리오	50
4.4.4 테스트 절차	51
4.5 테스트 결과 및 분석	52
4.5.1 검출률 테스트 결과	52
4.5.2 시스템 실시간 처리 테스트 결과	54
 V. 결론	56
참 고 문 헌	59

표 목 차

표 2.1 딥러닝 알고리즘 구조 및 활용	17
표 2.2 보행자 검출 알고리즘 종류	23
표 4.1 시스템 구현 환경	42
표 4.2 시스템 테스트 환경	48
표 4.3 제안하는 시스템 모듈 별 처리 시간 비교 및 결과	55

그 립 목 차

그림 2.1 DCP 기반 안개 제거 알고리즘	6
그림 2.2 DCP 안개 제거	8
그림 2.3 안개 유무에 의한 다크 채널과 누적 히스토그램	9
그림 2.4 단층 퍼셉트론	12
그림 2.5 다층 퍼셉트론	12
그림 2.6 딥러닝 학습 모델의 필요성	14
그림 2.7 딥러닝 고수준 특징 추출 과정	15
그림 2.8 딥러닝 알고리즘 분류 및 특징	16
그림 2.9 CNN 구조	18
그림 2.10 LeNet5(LeCun et al., 1998)	19
그림 2.11 AlexNet(Krizhevsky et al., 2012)	20
그림 2.12 GoogLeNet(Christian, Szegedy, et al., 2015)	20
그림 2.13 보행자 검출 순서도	22
그림 2.14 HOG 기반 보행자 검출 순서도	24
그림 2.15 HOG 기반 보행자 특징 추출	24
그림 2.16 R-CNN 구조	25
그림 2.17 Fast R-CNN 구조	26
그림 2.18 YOLO 구조도	26
그림 2.19 YOLO 최종 출력단 기능	27
그림 3.1 기상 악화 대응 보행자 검출 시스템 개요도	30
그림 3.2 기상 악화 대응 보행자 검출 시스템 구조도	31

그림 3.3 기상 악화 대응 보행자 검출 시스템 시퀀스 다이어그램	33
그림 3.4 기상 악화 대응 보행자 검출 시스템 DFD	34
그림 3.5 기존 시스템 알고리즘	36
그림 3.6 제안하는 딥러닝 기반 시스템 알고리즘	37
그림 3.7 블랙박스 화면 설치 비율	39
그림 3.8 제안하는 그리드 분할	41
그림 4.1 구현 시스템 구성도	43
그림 4.2 INRIA Person Dataset 사람 이미지 추가 학습	44
그림 4.3 보행자 검출 영상 업로드 페이지	45
그림 4.4 제안하는 안개 제거 전처리 결과	46
그림 4.5 제안하는 기상 악화 대응 보행자 검출 시스템 최종 결과물	47
그림 4.6 기존 연구 보행자 검출 결과	49
그림 4.7 제안하는 시스템 DR 비교 및 측정	53
그림 4.8 제안하는 시스템 MR 비교 및 측정	53

I. 서 론

최근 자율주행 자동차 시스템에 대한 관심이 증가하고 있다. 자율주행 자동차는 ‘인지-판단-제어’ 순으로 동작하기 때문에 인지기술의 발전은 자율주행 시스템의 가장 기본이라고 볼 수 있다. 따라서 지능형 운전자 보조 시스템(ADAS)과 같은 차량 운행 환경과 인식 기술의 연구 발전이 활성화되고 있다.

ADAS는 자율주행 자동차의 안전을 책임지는 기술로 각종 위협이나 사고에 대비해야 하며 특히 주행 시 보행자 사고는 치명적으로 생명과 직결되어 큰 사고로 이어질 수 있는 문제이기 때문에 그 무엇보다도 중요한 분야이다. 따라서 자율주행 자동차 영상에서 보행자 검출률 개선을 위한 연구는 꼭 필요하다.

기존 보행자 검출 연구에서 가장 많이 사용되는 기법은 Hand-crafted Model의 방식으로 대표적으로 HOG(Histograms of Oriented Gradients) 알고리즘을 활용한다. 하지만 HOG 알고리즘은 엣지(edge)에 기반을 둔 특징 추출 기법이기 때문에 보행자의 크기가 작게 보이거나 기상 악화와 같이 조도가 낮거나 분별력이 떨어지는 상황에서 검출이 힘들다는 제약 사항이 있다.

최근 5년(2010년~2014년)간 교통안전공단이 봄철(3~5월) 기상상태에 따른 교통사고 발생현황을 분석한 결과에 따르면 기상상태 별 보행자 교통사고의 치사율은 안개(12.7%), 흐림(4.4%), 비(3.6%), 눈(2.5%), 맑음(2.4%), 기타(0.1%) 순으로 안개 낀 날 보행자 교통사고 치사율이 맑은 날에 비해 5.3배 더 높게 나타났다[1]. 이는 안개 낀 날 혹은 흐린 날 운전

시 기상상태에 따른 시야 미확보가 안전운행에 가장 큰 방해요인으로 작용하여 보행자 교통사고 치사율을 높인 것으로 분석된다.

따라서 기상 악화로 인한 시야 분별력이 좋지 않은 환경변수에 대응할 수 있는 보행자 검출 연구가 꼭 필요하다.

이와 같은 문제점을 보완하고 오검출률을 줄이기 위한 대안으로 최근에는 인공지능의 핵심기술인 딥러닝을 기반으로 보행자를 검출하는 연구가 활발히 이루어지고 있다.

대표적인 연구로 AlexNet 구조는 컨볼루션 신경망을 이용하여 각 뉴런의 공간배치를 3차원으로 만듦으로써 기존에 문제로 여겨지던 너무 많은 파라미터로 인한 Overfitting을 최소화시켰고 뉴럴 네트워크를 영상처리에 적용시키기 용이하게 하였다[2]. 이와 같이 딥러닝을 이미지 분류에 활용하여 검출하는 연구는 뛰어난 검출성능을 보여주고 있으나 기존의 방식보다 학습시간이 오래 걸리고 상대적으로 실시간 처리에 취약한 면이 있다.

따라서 본 논문에서는 CNN의 구조를 개선하여 응답시간은 줄이고 검출률을 높이는 시스템을 제안하고자한다.

또한, 주행 중 시간이나 날씨 등의 변수로 인하여 만들어진 다양한 환경으로 생긴 노이즈로 인해 인식률이 떨어지는 상황에 대비하여 원본 영상에서 일정 영역에 대한 DCP 알고리즘을 이용하여 대기광을 산출하고 각 화소에 따른 빛의 전달량을 보정하여 딥러닝 학습을 시킴으로써 기상 악화 시에도 미검출률을 감소시켰다.

본 논문에서 제안하는 시스템을 설계하기 위해 입력받는 차량 영상 데이터를 안개제거 전처리하고 기존의 딥러닝을 통해 만든 특징 분류 모델을 바탕으로 보행자 영역을 추출하여 보행자 인식 후 식별 결과를 전송하는

시스템을 설계하였다.

본 논문은 총 5장으로 구성된다. 1장 서론에서는 본 논문의 필요성 및 목적에 대해 기술하였고, 2장에서는 안개 제거 기술과 딥러닝 기술에 대해 분석하였다. 또한, 해당 시스템에 보행자 검출을 적용하기 위해 기존 보행자 검출 시스템을 분석하였다. 3장에서는 2장에서 분석한 내용을 토대로 기상 악화 대응 보행자 검출 시스템을 설계하였다. 4장에서는 3장에서 설계한 기상 악화 대응 보행자 검출 시스템을 구현하고 테스트하였으며 이를 통해 성능을 평가하였다. 마지막으로 5장 결론에서는 본 연구의 결론을 맺는다.

II. 관련연구

본 장에서는 딥러닝 기반 기상 악화 대응 보행자 검출을 설계하는 데 필요한 관련 연구에 대해 분석하였다. 우선 기상 악화 대응에 필요한 영상의 안개 노이즈 처리에 대하여 분석 후 인공지능의 핵심 기술인 딥러닝에 대해 분석한다. 또한, 보행자 검출을 위하여 기존에 활용되고 있는 기술 및 딥러닝 검출 기술에 대해서도 분석한다.

2.1 안개 영상 노이즈 처리

실외의 환경에서 촬영되는 차량 영상은 악천후 환경에 영향을 받기 쉬우며 이는 영상 화질의 저하를 발생시켜 보행자와 같은 물체 인식 알고리즘의 성능저하를 일으킨다. 그런데 안개 낀 날 주행 중인 차량의 전방 카메라로부터 얻은 영상은 비나 눈이 오는 환경에서 얻은 영상과 비슷하다[3]. 따라서 자율주행 자동차의 주행 중 기상 악화 대응에 안개 제거 기법을 적용하여 악천후의 상황에서도 보행자 인식 성능을 개선할 수 있다.

2.1.1 안개 영상의 물리적 모델

안개나 스모그가 낀 날씨의 영상은 색과 엣지 정보가 불분명해지기 때문에 영상인식이 어렵다. 따라서 악천후의 상황에서도 영상인식의 정확성을 높이기 위해서는 전처리 과정으로 안개를 제거하는 과정이 필요하다[4].

안개 제거를 통한 영상 보정을 위해서 최근 기법으로 안개 물리적 모델링

을 제안하고 있다.

안개 물리적 모델링은 영상에서 거리가 멀수록 대기 중 안개 입자로 인하여 물체로부터 반사되어 오는 빛의 산란도가 높아진다는 성질을 이용하였다. 식 (2-1)은 안개 제거 알고리즘에서 사용하는 안개 물리 식이다.

$$I(x) = J(x)e^{-\beta d} + A(1 - e^{-\beta d}) \quad (2-1)$$

위 식에서 $I(x)$ 는 카메라로부터 획득한 안개가 포함된 영상이고, $J(x)$ 는 최종적으로 얻어야 하는 안개가 없는 깨끗한 영상이며, A 는 대기광으로 안개의 밝기 정도를 나타낸다. d 는 카메라와의 거리이며, β 는 대기 감광계수이다. $e^{-\beta d}$ 는 안개의 농도를 나타내어 거리가 멀고 안개가 짙을수록 즉, β 와 d 의 값이 클수록 $e^{-\beta d}$ 가 작아져 깨끗한 영상인 $J(x)$ 의 정보에 대한 비율 값이 작아지기 때문에 본래의 색 정보를 손실하게 된다. 따라서 안개를 제거하기 위해서는 카메라로 획득한 안개 영상 $I(x)$ 에서 영상의 각 지점의 안개 전달량을 찾고 $J(x)$ 에 관한 식으로 변환 후 안개를 제거해야 한다.

2.1.2 Dark channel prior를 이용한 안개 제거

기존의 영상에서 안개를 제거할 경우에 대비나 반사율을 이용하여 제거하였다. 하지만 대비를 이용한 안개제거는 채도가 높아지고, 반사율을 이용하면 심한 안개에 제거가 잘 안 되는 취약점이 있다.

He[5]등이 안개 없는 깨끗한 영상의 일정 구간 내의 한 픽셀 중 어느 한 컬러 채널은 0에 가까운 어두운 값을 갖는 픽셀이 존재한다는 통계적 관측

을 발견하였다[6]. 이것을 Dark Chanel Prior(DCP)라 부르고 이와 같은 특징을 이용해 거리를 측정하기 어려운 이미지 안에서 각각의 피사체에 대해 안개의 농도를 예상하는 방식을 제안했다.

안개 제거 방식은 다크 채널을 이용하여 대기광을 검출하고 각 화소 별 빛의 전달량을 산출 후 Cross Bilateral Filter를 이용하여 엣지성분을 포함한 전달량을 보정하여 후광 효과를 제거한다. 보정된 전달량은 화소별 RGB 채널을 이용하여 안개 제거를 한다. 그림 2.1은 이와 같은 알고리즘 과정을 도식화하였다[7].

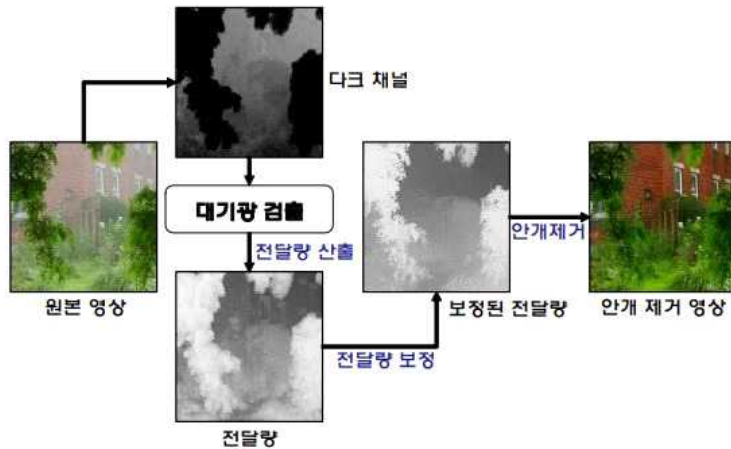


그림 2.1 DCP 기반 안개 제거 알고리즘

DCP 식은 (2-2)와 같이 나타낸다.

$$\mathcal{J}^{dark}(x) = \min_{y \in \Omega(x)} (\min_{c \in rgb} \mathcal{J}^c(y)) \quad (2-2)$$

여기서 J^c 는 영상의 각 채널을 의미하고, $\Omega(x)$ 는 픽셀 x 를 중심으로 하는 일정영역이며 y 는 $\Omega(x)$ 내부에 있는 픽셀의 위치를 의미한다.

DCP 에 의하여 식 (2-2)의 대부분의 값은 0에 근접한 값을 가지게 되며 이는 식 (2-3)과 같이 나타낸다.

$$J^{dark}(x) \simeq 0 \quad (2-3)$$

단일 영상에서는 영상에서 거리정보를 얻기 어렵다. 이 때문에 색상 정보를 바탕으로 안개 전달량을 구한다. 식(2-1)의 안개의 농도를 나타내는 $e^{-\beta d}$ 를 전달량 $t(x)$ 로 바꾸면 식 (2-4)와 같다.

$$I(x) = J(x)t(x) + A(1-t(x)) \quad (2-4)$$

식 (2-4)의 양변에 일정 구간과 각 RGB신호의 각 채널에 대한 최소값을 계산한 후 대기광 A로 나누어 주면 식 (2-5)와 같다. DCP에서 대기광은 다크 채널에서 밝기 값이 상위 0.01%이상인 픽셀들을 후보군으로 정한다.

$$\begin{aligned} & \min_{y \in \Omega(x)} (\min_{c \in rgb} (\frac{I^c(y)}{A^c})) \\ &= t(x) \min_{y \in \Omega(x)} (\min_{c \in rgb} (\frac{J^c(y)}{A^c})) + 1 - t(x) \end{aligned} \quad (2.5)$$

식 (2-3)과 같이 안개가 없는 영상에서는 $J(x)$ 를 0으로 가정할 수 있으

므로 이를 전제하에 식 (2.5)를 정리해 보면 식 (2.6)이 나온다.

$$t(x) = 1 - \min_{y \in \Omega(x)} \left(\min_{c \in rgb} \frac{I^c(y)}{A^c} \right) \quad (2.6)$$

여기에 안개 제거 양을 조절하기 위해서 가중치 $w(0 < w \leq 1)$ 를 취한다. 최종적으로 DCP 이론 기반 전달량 식 (2.7)을 구할 수 있으며 He는 실험적 결과로 의해 가중치를 0.95로 제안하고 있다.

$$t(x) = 1 - w \times \min_{y \in \Omega(x)} \left(\min_{c \in rgb} \frac{I^c(y)}{A^c} \right) \quad (2.7)$$

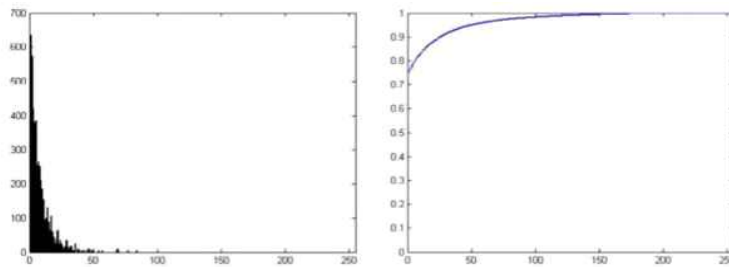
DCP로 안개 제거한 영상은 그림 2.2와 같이 나타나며 (a)는 안개가 있는 원본 영상이며 (b)는 위의 식으로 계산되어 진 전달량을 정렬한 결과이며 (C)는 안개 제거 된 복원 영상이다[8].



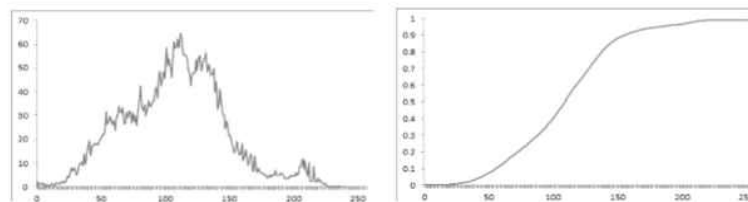
그림 2.2 DCP 안개 제거

2.1.3 Dark channel prior기반 안개 유무 판단

그림 2.3은 안개의 유무에 의한 영상에 대한 다크 채널과 누적 히스토그램을 나타낸다.



(a)안개가 없는 영상에 대한 dark channel과 누적 히스토그램



(b)안개 영상에 대한 dark channel과 누적 히스토그램

그림 2.3 안개 유무에 의한 다크 채널과 누적 히스토그램

안개가 없는 영상에서는 대부분의 다크 채널이 0 부근에 집중되어 있지만 안개 영상의 경우는 전체적으로 다크 채널의 값이 매우 커짐을 알 수 있다. 또한, 누적 히스토그램에서 통해 거리에 비례하여 다크 채널 값이 지수 함수의 모양으로 증가하는 특성을 확인할 수 있다. 이는 안개가 객체에 미치는 영향을 나타낸다.

이와 같은 통계적 데이터를 근거로 데이터를 분석으로 안개를 제거하는

식은 식(2-8)과 (2-9)와 같다.

$$D(i) = \begin{cases} 1, & \text{if } J^{dark}(i) < Th_{dark} \\ 0, & \text{otherwise} \end{cases} \quad (2-8)$$

$$Fog\ rate = \frac{1}{N} \sum^i D(i) \quad (2-9)$$

여기서 $J^{dark}(i)$ 는 국부 영역 내의 i 번째 화소에 대한 다크 채널 값이며 Th_{dark} 는 다크 채널에 대한 임계값이다. N 은 국부 영역내의 화소수를 의미하고 만약 $Fog\ rate$ 가 Th_{fog} 보다 작을 때, 안개가 있는 것으로 판단한다.

2.2 딥러닝

인공신경망(Neural Network, NN)은 가상의 뉴런을 수학적으로 모델링한 것으로 인간의 뇌와 같은 학습 능력을 지닌 인공지능 기계학습 기술 중 신경모형 패러다임에 속한다[9].

인공신경망에서도 깊게 구성하는 것을 심층신경망이라고 지칭하며 새로운 패러다임을 맞으면서 딥러닝(Deep Learning)이라는 신조어로 불리고 있다. 따라서 인공신경망의 핵심기술인 딥러닝을 대해 자세히 분석하였다.

2.2.1 인공신경망

인공신경망은 과거에서부터 많이 연구가 되어오고 있는 오래된 알고리즘으로 몇 가지 단점으로 인해 정체되어 왔던 연구가 2000년대 이후 딥러닝의 출현으로 다시 활성화 되고 있다.

인공신경망의 모티브는 생물학의 신경망으로 시냅스의 결합으로 네트워크를 형성한 인공 뉴런(노드)이 학습을 통해 시냅스의 결합 세기를 변화시켜 문제 해결을 하는 능력을 가진 수학적 모델 전반을 가리킨다[10].

인공신경망의 구조는 일반적으로 입력층(Input Layer), 은닉층(Hidden Layer), 출력층(Output Layer)로 구성되어 있으며 각 층의 뉴런들이 가중치를 통해 연결되어 있다.

인공 신경망은 크게 단층 퍼셉트론과 다층 퍼셉트론으로 나뉠수 있는데 Frank Rosenblatt은 퍼셉트론을 인지 가능한 뉴런이라는 의미로 정의하였으며 초기의 모델은 학습데이터를 통해서 가중치를 스스로 학습할 수 없어서 사람이 수동으로 조율해야 했다[11]. 하지만 인공 신경망 학습이 스스로 가능해 지면서 출력층에서 계산된 출력과 실제 출력의 값 차이를 최소화시키는 가중치를 찾는 것을 목적으로 연구가 진행되어오고 있다. 최적의 가중치를 학습하기 위해서는 인공 신경망에서 목적 함수(Object Function or Loss Function)를 정의한다. 이는 가중치의 변화에 따른 오차의 변화를 계산할 수 있게 하기 때문에 오차와 가중치의 관계를 연쇄법칙(Chain Rule)을 이용한 편미분으로 구할 수 있다.

단층 퍼셉트론(Single-layer Perceptron)은 인공지능의 가장 간단한 구조로 그림 2.4와 같이 입력을 받는 입력층과 신경망의 결과가 출력되는 출력층으로 구성되어 있다.

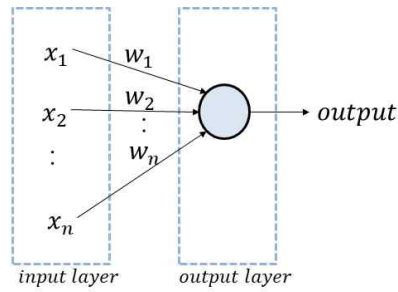


그림 2.4 단층 퍼셉트론

단층 퍼셉트론은 선형분리가 가능한 문제에만 적용 가능한 방식이기에 XOR(선형분리의 단순처리)과 같은 형태의 문제에서 한계를 가지고 있다 [12]. 이를 극복하는 방법으로 만들어진 다층 퍼셉트론(Multi-layer Perceptron)은 1985년에 Geoffrey Hinton에 의해 개발되었다[13]. 다층 퍼셉트론은 그림 2.5와 같이 여러 층을 가진 신경망으로 입력을 받는 입력층, 신경망의 결과가 출력되는 출력층, 입력층과 출력층 사이에 외부에 드러나지 않는 은닉층의 3계층으로 구성되며 두 개 이상의 은닉층이 있을 경우에 심층신경망(Deep Neural Network, DNN)으로 정의한다[14].

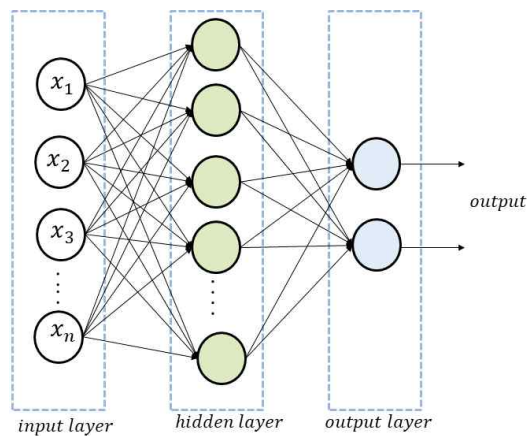


그림 2.5 다층 퍼셉트론

1980년대 중반에 PDP그룹에 의해 다층 퍼셉트론을 역전파 알고리즘에 의해 학습시킬 수 있음이 알려졌다. 이로 인하여 은닉층에 연결된 연결강도들을 조정할 수 있는 학습 방법의 어려움을 해결하게 되었고 다층 퍼셉트론은 가장 널리 사용되는 인공신경망 모형이 되었다. 일반적으로 다층 퍼셉트론의 경우, 경사 하강법(Gradient Descent)에 기반하는 오차 역전파 학습 알고리즘(Error Back-propagation Learning Algorithm)에 의해 학습되며 오차 역전파 학습 알고리즘은 먼저 은닉층에서 출력층 사이의 연결 가중치가 갱신되고, 이 결과를 토대로 입력층에서 은닉층 사이의 연결 가중치가 갱신되는 방향으로 진행된다[15].

2.2.2 딥러닝의 정의

딥러닝(Deep Learning)은 인공신경망기반의 아주 많은 은닉층을 가진 심층 신경망 혹은 심층망(Deep Networks)을 학습 시키는 방식으로 상위층으로 갈수록 복잡한 정보를 추출하는 구조를 가진 머신러닝 모델이다[16,17].

신경망은 뉴런의 층을 증가 시킬수록 그림 2.6과 같이 복잡한 패턴 분류 경계면을 생성하기 한다. 이와 같은 이유로 다층 신경망을 학습시키는 딥러닝은 패턴인식(Pattern Recognition)이나 추론(Inference)등에 효율적으로 활용할 수 있다.

최근에는 음성인식, 언어번역, 영상인식을 비롯한 다양한 패턴인식 분야에서 두각을 나타내고 이미지, 비디오 분류, 이미지 분할, 물체 인식, 특징점 인식, 물체 추적 등 다양한 분야에 적용 가능 하게 되면서 많은 연구가 활성화되고 있다.


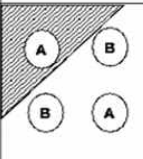
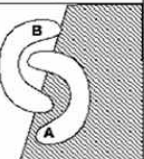

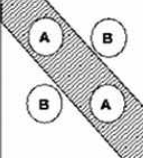
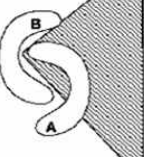
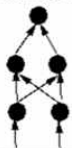
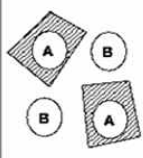
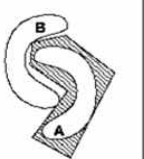
Structure	Regions	XOR	Meshed regions
single layer 	Half plane bounded by hyper-plane		
two layer 	Convex open or closed regions		
three layer 	Arbitrary (limited by # of nodes)		

그림 2.6 딥러닝 학습 모델의 필요성

이와 같이 다양한 분야에 많은 연구가 이뤄질 수 있는 이유는 컴퓨터 파워 성능이 좋아져 기존의 신경망에 비해 더 많은 중간 계층을 사용함으로써 데이터에 대한 표현 능력을 크게 증가시킬 수 있기 때문이다. 그리고 무엇보다도 차원수의 저주 문제, 과다학습 문제, Vanishing Gradient 문제, Non-convex 최적화 문제, 느린 학습 속도 등의 딥러닝 기존의 문제점들을 학습구조와 알고리즘 기술의 발전으로 해결 가능해진 점이 매우 크다.

딥러닝은 연구자나 개발자의 복잡한 네트워크 설계 없이도 빅데이터와 GPU 병렬처리 기술을 활용하여 주요 특징점을 자동으로 추출해 개선된 학습 성능을 보여주는 강점을 가지고 있다[18].

기존에는 데이터 전처리 및 가공을 통해 문제 해결에 적합한 특징을 추출하고 이를 패턴 분류기에 훈련시키기 적합한 학습 데이터로 이용하는 두 단계로 문제를 구분하여 해결 하였다. 하지만 딥러닝은 특징 추출을 위한 전

처리 단계를 전체 학습 프로세스에 포함시킴으로써 가공되지 않은 데이터를 직접 학습하는 통합된 문제 해결 방식을 가지고 있다. 이와 같은 방식은 전처리나 Feature Engineering을 통해 배제된 해의 영역에서 손실 될 수도 있는 정보조차 활용하기 때문에 차원수가 아주 크고 복잡한 영상데이터와 같은 경우에 유용하게 활용할 수 있다.

딥러닝은 복잡한 기능을 학습하기 위해서는 데이터로부터 고수준특징 (High-level Feature)을 학습할 수 있는 능력이 요구된다[19]. 예를 들면, 사람의 눈, 코, 입의 형태와 위치 같은 특징을 이용하는 것으로 화소, 엣지, 명도 변화의 방향, 좁은 영역의 텍스처와 같은 저수준특징을 학습하는 것보다 환경변화에 적응하기 좋고 정확도 높고 안정적인 성능을 얻을 수 있다. 딥러닝의 각 신경망의 계층들은 그림 2.7과 같이 하위계층에서부터 정보를 입력받아 추상화함으로서 상위계층에 고수준 정보를 전달해줄 수 있다.

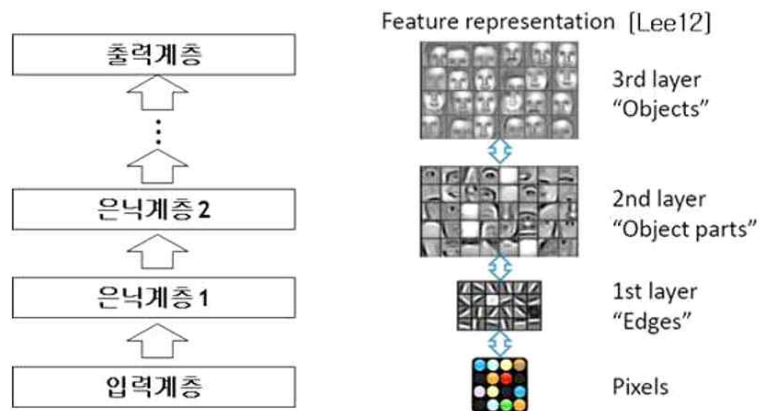


그림 2.7 딥러닝 고수준 특징 추출 과정

이와 같은 특징으로 인하여 신경망의 계층이 많을수록 높은 수준의 특징을 추출할 수 있다.

2.2.3 딥러닝의 알고리즘

딥러닝은 머신러닝의 한 종류로 머신러닝 알고리즘을 분류하는 기본 규칙과 동일하며 딥러닝 알고리즘을 학습 방식으로 분류하면 그림 2.8과 같이 지도 학습, 비지도 학습, 강화 학습으로 분류할 수 있다[20].

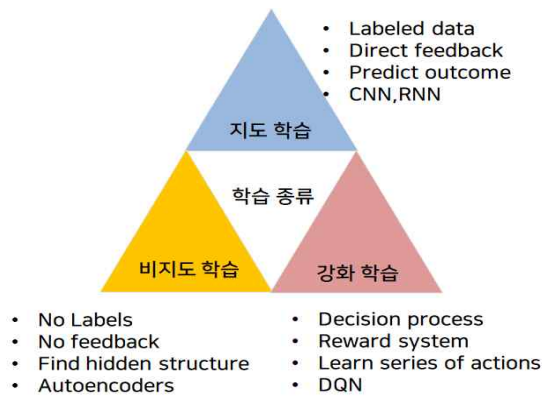


그림 2.8 딥러닝 알고리즘 분류 및 특징

지도학습은 입력 값과 출력 값으로 이루어진 훈련 데이터를 활용하는 학습하는 방법으로 도출된 함수를 통해 컴퓨터가 새로운 데이터에 대한 결과를 찾도록 하며 예측과 분류를 하는데 사용한다. 딥러닝에서 지도학습과 관련된 알고리즘은 대표적으로 CNN, RNN(Recurrent Neural Network) 등이 있다.

비지도 학습은 지도 학습과는 달리 데이터에 대한 레이블이 없고 정확한 함수를 도출할 수 없기 때문에 컴퓨터가 스스로 새로운 데이터에 대하여 분류하고 숨겨진 특징이나 구조를 발견하는데 사용한다. 딥러닝에서 주로 사용되는 구조는 Autoencoders이다.

마지막으로 강화 학습은 입력된 데이터가 가지고 있는 특징과 주어진 환경에 따라서 행동을 결정하여 실행하고 평가를 한 후 평가에 따른 보상이 주어지는 방식으로 딥러닝 알고리즘으로 DQN(Deep-Q-Network)가 있다. 표 2.1은 학습방식에 따라서 딥러닝 알고리즘의 구조 및 특징을 비교 분석하였으며 실제로 활용되는 분야에 관해서도 기술해놓았다[21].

본 논문은 딥러닝 알고리즘 중에서 CNN 알고리즘을 활용하였기 때문에 다음 장에서는 합성곱 신경망에 대하여 자세히 다루었다.

표 2.1 딥러닝 알고리즘 구조 및 활용

방식	알고리즘	구조 및 특징	활용 분야
지도 학습	CNN (Convolutional Neural Network)	Convolutional Layer, pooling Layer, Fully Connected Layer로 이루어진 합성곱 신경망	영상 분석
	RNN (Recurrent Neural Network)	인공신경망을 구성하는 유닛 사이의 연결이 Directed Cycle을 구성하는 신경망	자연어처리, 필기체 인식
비지도 학습	DBN (Deep Belief Network)	기계학습에서 사용되는 그래프 생성 모형으로 잠재변수의 다중계층으로 이루어진 심층 신경망	영상, 음성, 자연어처리 등 전 분야
	Autoencoders	출력값과 입력값의 개수가 동일한 심층 신경망	비정상 거래 검출
강화 학습	DQN (Deep Q-Network)	심층 강화학습을 통해 인간수준의 제어를 가능하게 하는 신경망	실제 인간이 하는 행동의 상당부분 예상

2.2.4 합성곱 신경망

합성곱 신경망(Convolutional Neural Network, CNN)은 기계학습 모델 중에서 인공신경망 (Artificial Neural Network, ANN)의 딥러닝 영역으로 CNN의 기본 구조는 보통 convolution layer, pooling layer, fully connected layer의 3가지 계층의 신경망을 쌓아 성능을 높인다.

Convolution 단계는 필터를 활용하여 특징을 추출하는 과정으로 필터를 적용할 시 건너뛸 픽셀의 수를 Stride라 한다. 따라서, Stride가 클수록 연산 속도는 빨라지며 적중률은 낮아진다[22]. Pooling 단계는 Max Pooling 기법을 사용하여 Convolution 단계를 통해 얻은 Feature Map의 크기를 압축시킨다. Convolution 단계와 Pooling 단계를 반복 진행하다 Fully Connected 단계의 과정을 수행하게 되면 딥러닝의 출력부와 이어져 입력받은 영상 특징 추출이 가능해진다. 그림 2.9는 다음과 같은 CNN의 일반적인 구조를 나타낸다.

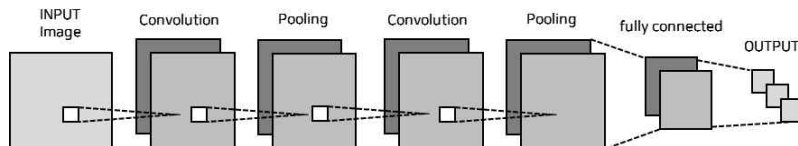


그림 2.9 CNN 구조

CNN은 영상처리를 위해 고안된 특수한 다층신경망으로 패턴의 크기나 위치가 바뀌어도 인식을 할 수 있다는 장점이 있어 최근 영상처리 분야에서 다양한 모델들이 개발되어오고 있으며 대표적인 CNN모델로는 LeNet5, AlexNet, GoogLeNet이 있다.

LeNet5는 최초의 합성곱 신경망의 기초 모델이며 다층신경망인 Neocognitron(Fukushima, 1980)의 발전을 토대로 만들어졌다[23].

LeCun et al.(1998)가 제안하는 CNN 구조인 LeNet5는 그림 2.10과 같으며 Local Receptive Field, Shared Weight, Sub-sampling의 3가지 특징을 갖고 있다[24].

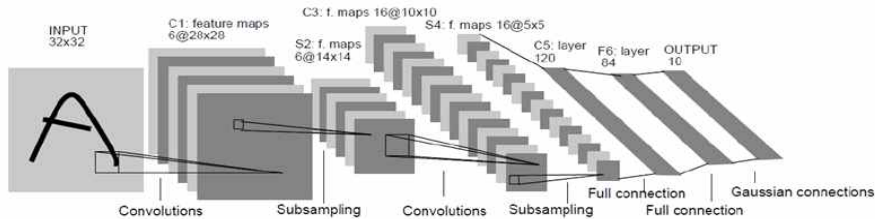


그림 2.10 LeNet5(LeCun et al., 1998)

Local Receptive Field는 기존의 모형이 하위계층의 각 노드가 모든 상위 계층의 노드와 연결되어 있는 것과 달리, 상위계층의 일부의 노드에만 가치로 연결되는 것을 의미한다. Shared Weight는 가중치를 공유한다는 뜻이다. 한 개의 패널에 있는 노드와 입력패턴이나 하위계층에 이용되는 필터는 동일하다는 의미이다. Sub-sampling은 Convolution 레이어의 크기를 압축시키며 여기서 필터크기는 2x2, Stride는 1이다.

LeNet5는 Convolution 레이어 3개와 Fully connected 레이어 1개로 구성되며 필터크기는 5x5이고 Stride는 1이다. 모든 층에는 Sigmoid활성함수가 사용되었고 출력층에는 Softmax활성함수가 사용되었다[25].

다음으로 소개할 구조는 AlexNet으로 그림 2.11은 Krizhevsky et al.(2012)에 의해 발표된 AlexNet의 구조를 나타낸 그림이다[26].

AlexNet는 컴퓨터 분야에서 합성곱 신경망이 널리 적용될 수 있는 계기가 된 구조로써 2012년 ImageNet 대회에서 우승을 한 모델이다. 총 5개의

딥러닝의 활성화로 신경망 모델의 망이 깊어지면서 더 복잡한 문제를 해결할 수 있게 되었지만, 망을 깊게 하면서 발생하는 문제들도 생겼다.

문제는 크게 3가지로 나누고 있다. 첫 번째로 모수(Parameter)가 늘어나면서 과적합 문제가 생길 수 있고, 두 번째로는 이에 따라 연산속도가 느려진다. 마지막으로 신경망이 깊어져 입력층과 근접한 부분의 모수 업데이트가 잘 안 되는 Vanishing Gradient 문제가 발생한다.

GoogLeNet은 Christian, Szegedy, et al.(2015)에 의해 개발되었으며 Inception Module, Auxiliary Classifier를 활용하여 구조적인 방식을 바꿈으로 인해 망을 깊게 하는 동시에 위와 같은 문제들을 해결하였다.

Inception Module는 1x1 필터 층을 사용하여 입력에 대한 차원을 크게 줄였고 여러 개의 필터를 하나의 합성곱 층에서 사용하여 다양한 크기의 Feature를 뽑아내는 방법으로 이를 도입함으로써 과적합 문제를 해결하고 망을 깊게 하여도 연산속도가 느려지는 것을 막을 수 있다.

Vanishing Gradient 문제를 해결하기 위해서는 망 중간에 학습을 위해 출력층 두는 Auxiliary Classifier이라는 방법을 사용하였다. 이를 사용하게 되면서 망이 깊어지면 깊어질수록 한계를 갖는 ReLU(Rectified Linear Unit)활성 함수와 달리 입력층과 인접한 모수의 기울기 변화량을 구할 수 있다.

2.3 보행자 검출 시스템

이번 장에서는 제안하는 시스템에서 핵심이 되는 보행자 검출 기술에 대해 알아보려고 한다. 이를 위해 기존 보행자 검출 시스템 구조 및 알고리즘

에 대해 분석하고 보행자 검출 시 활용되는 특징 추출 방법에 대해 다루고자 한다.

2.3.1 보행자 검출 시스템 구조

보행자 검출 기술은 크게 특징 추출, 보행자 식별, 보행자 검출 과정으로 분류할 수 있다. 일반적으로 보행자를 검출하는 과정은 그림 2.13과 같은 순서로 진행된다.

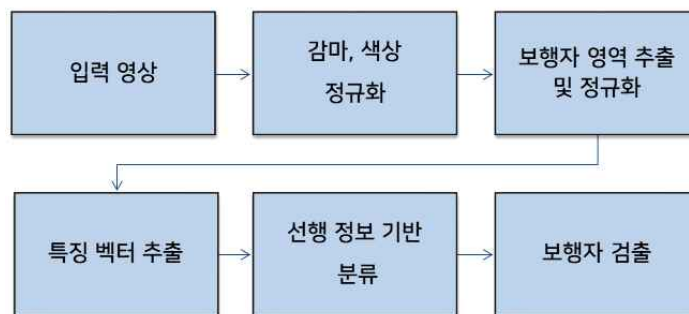


그림 2.13 보행자 검출 순서도

먼저 영상이 들어오면 영상의 감마, 색상을 보정해주고 본격적인 보행자 검출 단계로 넘어가서 보행자 영역을 추출하고 학습에 적합한 형태로 정규화하는 과정을 거친다. 이후 검출 단계에서 정규화한 보행자 이미지를 토대로 특징 벡터를 추출한다. 이로 인하여 추출된 수치화된 값은 선행 정보와 비교 및 분류하여 보행자 검출을 하게 된다.

2.3.2 보행자 검출 알고리즘

보행자 검출은 알고리즘은 크게 Hand-crafted Feature, Deep Model의 두 가지 기법으로 구분된다. 표 2.2는 다음의 특징 추출 기법으로 나눈 두 가지 기법 기준으로 나눈 보행자 검출 알고리즘의 종류를 나타내었다.

표 2.2 보행자 검출 알고리즘 종류

특징 추출 방식	알고리즘
Hand-crafted feature,	- HOG, Haar, Cascade Hog 등
Deep model	- R-CNN, SPPnet, Fast R-CNN 등

Hand-crafted Feature기법은 기존 연구에서 가장 많이 사용된 방식으로 대표적인 알고리즘으로는 HOG, Haar, Cascade Hog 등이 있다.

보행자 검출 방식은 이미지에서 Haar, Histogram of Gradient, Cascade Histogram of Gradient, Integral Channel Features 등의 특징을 추출하고 SVM(Support Vector Machine)이나 Bootstrapping을 이용하여 분류하는 방식이다.

이 중 보행자 인식의 기본적인 방법론이자 기존의 기상 악화 시스템에서 사용된 HOG 알고리즘(N.Dalal, B.Triggs, 2005)은 이미지 지역적 기울기 분포 특성을 해당영상 특징으로 사용하여 물체를 식별하는 방법이다. 보행자뿐만 아니라 자동차, 트럭, 표지판 등 다양한 물체 검출에 사용되고 있는 보편적인 방법으로 검출 순서는 그림 2.14와 같다[28].

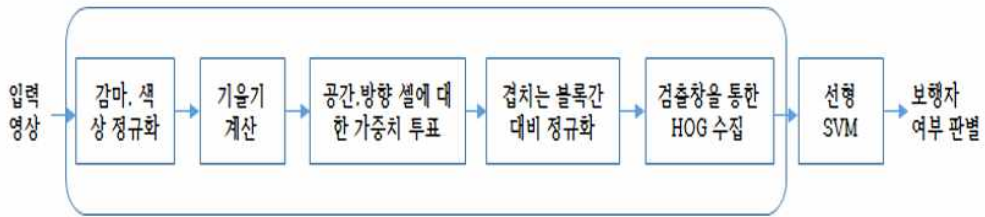


그림 2.14 HOG 기반 보행자 검출 순서도

HOG 영상 특징 추출 과정은 시각화하여 그림 2.15와 같이 나타내었다. a)는 입력 영상이며 b)는 기울기를 검출 한 결과이고 (c)는 영상 구역을 설정한 것이다. (d)는 각 구역에 대한 기울기 방향의 히스토그램을 나타냈다.

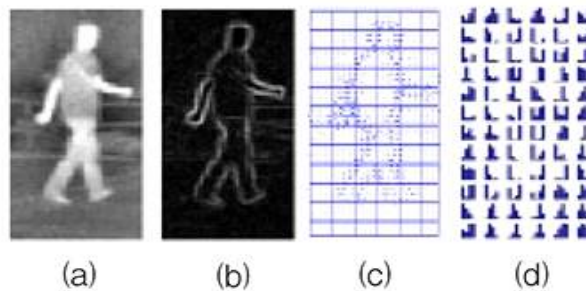


그림 2.15 HOG 기반 보행자 특징 추출

Deep Model은 최근의 딥러닝의 발전과 함께 연구가 많이 되고 있는 분야이며 물체 인식에서 좋은 성능을 보여주는 CNN 구조와 기법 연구로 성능을 향상 시켜오고 있다. 대표적인 알고리즘인 R-CNN(Region with Convolutional Neural Network)은 Selective Search를 이용해서 물체 추정영역을 모두 추출한 후 추출된 영역들을 학습된 CNN에 넣어 계산하여 결과를 분류하는 방식으로 그림 2.16과 같이 표현한다[29].

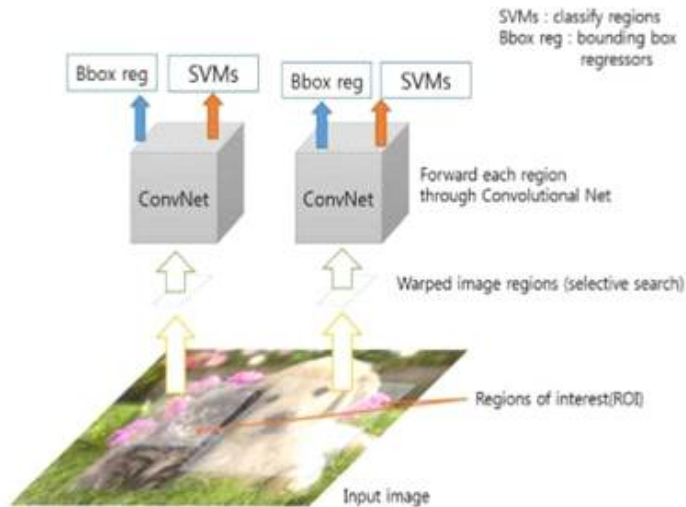


그림 2.16 R-CNN 구조

R-CNN은 높은 성능을 내지만 모든 추정영역이 CNN계산을 거치다 보니 처리 속도가 상당히 저하된다. 또한, R-CNN은 고정인 크기의 Feature Maps을 필요로 해서 입력 층에서 고정된 크기의 이미지만을 입력해야 한다. 그 후에 원본이미지에서 물체 추정영역을 워핑(Warp)과 크롭(Crop)을 하기 때문에 Resize되면서 이미지 정보에 손실 및 변형이 올 수 있다.

Fast R-CNN은 영역 기반 심화학습 중 하나로 Image Segmentation을 사용하지 않고 그림 2.17과 같이 입력 영상과 학습 영역(Region of Interest, RoI)을 완전 연결층을 제외한 Convolution Network를 이용하여 특징 맵을 생성한다. 그 후에는 RoI Pooling layer를 통해 특징을 추출 후 완전 연결층을 통해 학습한다. 학습 후 객체 검출 부분인 분류기와 검출 영역(Bounding Box Regression)에 대하여 소프트맥스(Softmax)와 크로스 엔트로피(Cross Entropy)를 이용하여 분류하고 검출 영역을 예측한다.

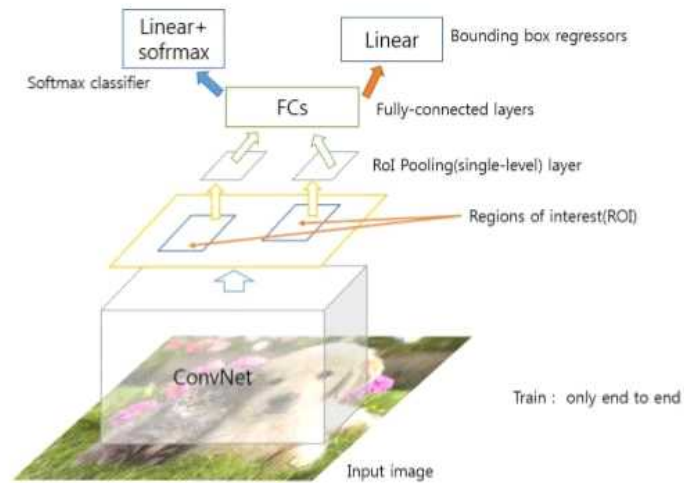


그림 2.17 Fast R-CNN 구조

Fast R-CNN에서는 Multi Task Loss를 이용하여 R-CNN에서 CNN 특징들이 선형 독립적인 SVM과 객체 영역 좌표들에 대응하여 업데이트가 안 되는 문제를 해결하였고 End-to-End 학습 구조를 이용하여 복잡한 학습 구조를 가지는 문제들에 대해 빠르게 처리할 수 있는 방안을 찾았다.

하지만 이 또한 HOG알고리즘에 비하여 느린 처리 속도이기에 본 논문에서는 실시간 처리 속도가 향상된 Deep Model 방식의 검출 알고리즘인 YOLO(Joseph Redmon et al., 2016)알고리즘을 활용하였다. CNN 구조는 GoogleNet을 기반으로 하여 만들어진 방식이며 그림 2.18과 같다[30].

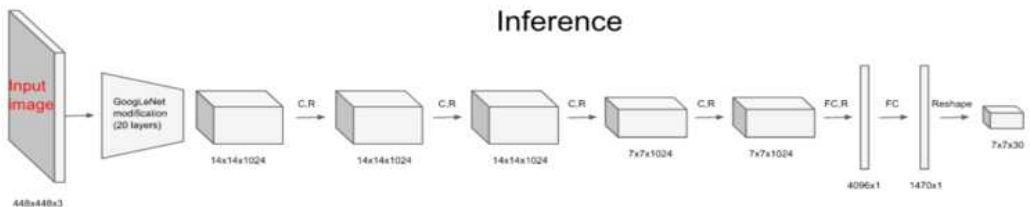


그림 2.18 YOLO 구조도

YOLO(You Only Look Once)는 이름 뜻 그대로 한 번의 학습과정으로 특징 추출 및 분류를 한 번에 하는 특징을 갖고 있어 매우 빠르다.

관심 영역을 지정하여 학습하였던 기존의 딥 모델 검출 방식과는 달리 YOLO는 이미지 전체를 학습하는 Grids 방식의 간단한 구조로 구성되어 있어 학습 시간이 짧다. 하지만 그 보다는 YOLO의 핵심 기술 영역은 최종 출력단(7*7*30)이라 할 수 있는데 그 이유는 딥러닝 학습을 통해 만들어진 특징 데이터가 마지막 출력단에 저장되어지며 이를 기반으로 경계박스 위치 찾기, 클래스 분류와 같은 모든 기능을 한 번에 수행하기 때문이다.

그림 2.19는 YOLO의 최종 출력단을 나타내며 경계박스와 신뢰지수를 찾는 영역과 클래스를 분류하는 영역으로 나뉘며 이를 합쳐서 객체를 검출하는 과정을 나타낸다.

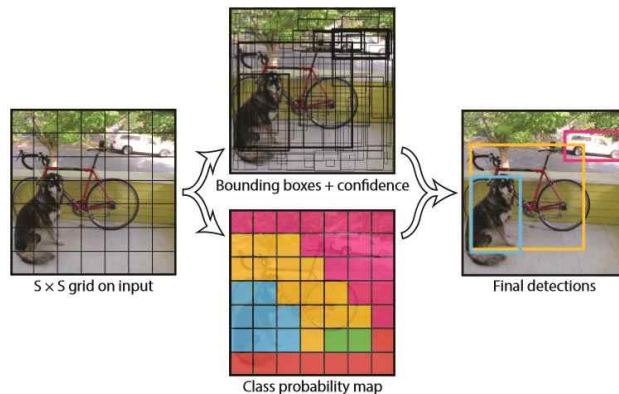


그림 2.19 YOLO 최종 출력단 기능

2.4 기상 악화 대응 보행자 검출 시스템 제안

최근 자율주행 자동차 시스템에 대한 관심이 증가함에 따라서 안전은 중

요한 화두가 되었으며 각종 위협이나 사고에 대비하는 기술인 지능형 운전자 보조 시스템(ADAS)과 같은 차량 운행 환경과 인식 기술의 연구 발전이 활성화되고 있다. 그 중에서도 보행자 사고는 생명과 직결되는 치명적인 문제이기 때문에 자율주행 자동차에서 보행자 검출은 무엇보다도 빠르고 정확하게 수행되어야 한다.

최근 5년간 수행한 교통안전공단의 연구에 따르면 안개 낀 날 보행자 교통사고 치사율이 평상시의 맑은 날보다 5.3배 더 높게 나타났는데 이는 안개 낀 날과 같은 흐린 날 운전 시 기상상태에 따른 시야 미확보가 가장 큰 방해요인으로 작용하여 사고율을 높인 것으로 분석되어진다.

따라서 외부 환경의 영향이 큰 자율주행 자동차의 경우 기상 악화의 환경 변화에도 대응하여 영상속의 보행자를 검출해야하며 이에 시야 분별력이 좋지 않은 환경변수에 대응할 수 있는 영상 객체 검출 연구가 필요하다.

안개 낀 날의 영상은 흐린 날, 비오는 날, 눈 오는 날, 스모그와 같은 악천후에서의 영상의 상태와 비슷하기 때문에 영상에서 안개 노이즈를 제거하는 방식으로 기상 악화 영상화면 전처리를 적용할 수 있다.

기존 논문을 분석해 본 결과 안개영상에서 보행자를 검출할 때, DCP 알고리즘을 활용하여 안개를 제거하였다. 검출 영역에서는 기존 보행자 검출 연구에서 가장 많이 사용되는 기법인 Hand-crafted Model의 HOG 알고리즘을 이용하여 보행자를 검출하였다.

하지만 HOG 알고리즘은 엣지에 기반을 둔 특징 추출 기법이기 때문에 보행자의 크기가 작게 보이거나 기상 악화와 같이 조도가 낮거나 분별력이 떨어지는 상황에서 검출이 힘들다는 제약 사항이 있다.

또한, 안개가 적거나 없는 이미지에 DCP 알고리즘을 적용할 경우 영상

왜곡이 생기며 안개가 아닌 밝은 흰색의 영역이 있을 경우에도 잘못 빛의 전달량을 잘못 계산해서 결과적으로 선명하지 않은 영상을 얻는다.

본 논문에서는 이러한 문제점을 해결하고자 안개의 유무를 DCP 전처리 단계를 거치기 전에 먼저 확인하고 DCP 대기광 검출영역을 재정의하여 대기광 오검출률을 줄였다. 이와 같은 기술을 사용하여 주행 중 다양한 상황에서도 적합한 빛 전달량을 구하게 됨으로써 선명한 영상에서 보행자 검출을 할 수 있다.

또한 보행자 검출방식을 빠르고 다양한 환경변수에 대응할 수 있는 YOLO 알고리즘을 활용한 구조를 제안하였다. 이로 인해 보행자 검출률은 높이고 알고리즘 구조의 변형을 통해 기존의 보행자 검출 Deep Model이 가지고 있는 처리속도 문제를 개선하였다.

따라서 본 논문에서는 차량용 블랙박스의 기상 악화 대응을 위한 딥러닝 기반 보행자 검출 시스템을 제안하고자 한다. 이를 위해 효율적인 안개 제거 알고리즘을 제안하고 보행자 인식을 위한 CNN 구조를 통합적으로 설계하여 악천후에도 보행자를 정확하고 빠르게 검출하고자 한다.

III. 기상 악화 대응 보행자 검출 시스템 설계

본 장에서는 기상 악화 대응 보행자 검출 시스템의 설계에 대해서 기술한다. 시스템의 개요 및 구조도를 설명하여 전체적인 이해를 한 후 시스템에 필요한 요구사항을 정의한다. 또한, 데이터 흐름도 및 시퀀스 다이어그램을 통하여 시스템의 흐름을 도식화하여 기술한다. 이후 제안하는 시스템 알고리즘을 설계하여 기존의 시스템과의 차별성을 확인할 수 있다.

3.1 기상 악화 대응 보행자 검출 시스템 개요

제안하는 기상 악화 대응 보행자 검출 시스템 개요도는 그림 3.1과 같다.

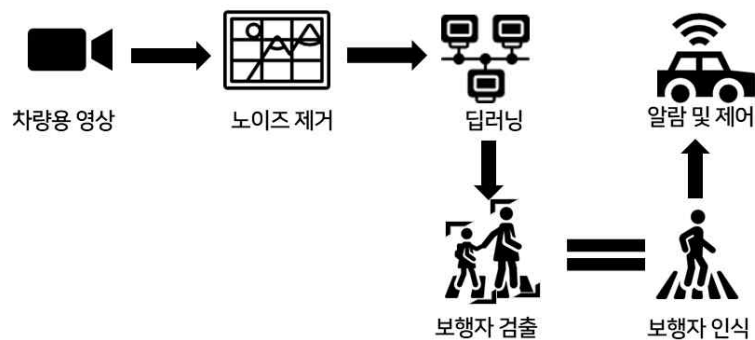


그림 3.1 기상 악화 대응 보행자 검출 시스템 개요도

먼저 블랙박스로 부터 수집한 차량용 영상 데이터가 들어오면 안개의 유무를 확인하고 제안하는 DCP 기반의 안개 노이즈를 제거방식의 전처리과정을 거친다. 그 후 제안하는 딥러닝 알고리즘으로 보행자를 검출하며 보행

자 인식 결과를 자동차 서버에 전송해주면 정보를 바탕으로 알람 및 기능 제어를 할 수 있다. 다음과 같이 제안하는 시스템을 구현하기 위해 설계한 구조도를 도식화하면 그림 3.2와 같다.

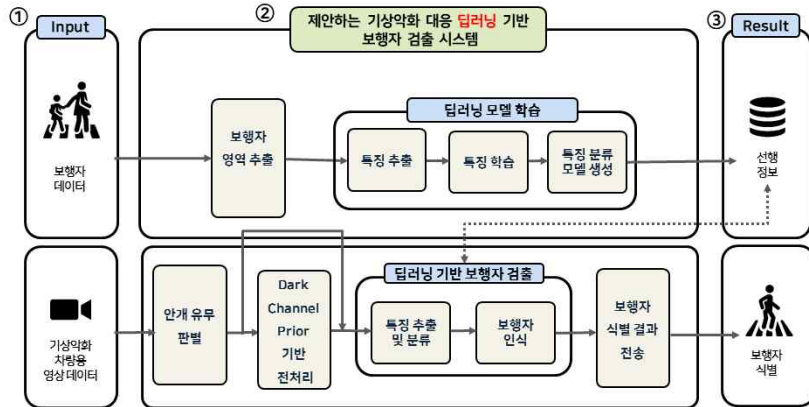


그림 3.2 기상 악화 대응 보행자 검출 시스템 구조도

제안하는 시스템은 보행자 데이터를 학습하는 영역과 실제 차량용 블랙박스 영상에서 보행자 검출을 적용하는 영역으로 나눌 수 있다.

먼저 학습하는 영역에서는 보행자가 포함된 이미지 데이터를 입력받는다. 이후 보행자 영역을 추출하고 정규화한 데이터를 바탕으로 딥러닝 학습을 통해 특징 벡터를 추출한다. 추출한 특징을 바탕으로 학습하여 특징 분류 모델을 생성 후 저장하여 향후 특징 분류 선행정보로써 활용한다.

실제 활용되는 영역에서는 블랙박스를 통해 기상 악화 영상 데이터를 입력받는다. 이후 안개 유무를 판별한 후에 안개가 있을 때는 DCP 기반의 안개 제거 전처리를 통해 보정된 영상을 만들고 안개가 없을 때는 안개 제거 전처리 다음 단계인 딥러닝 기반 보행자 검출 모듈로 간다.

보행자 검출 모듈에서는 CNN 제안 알고리즘으로 앞서 학습한 선행정보

를 활용하여 특징 추출 및 분류를 동시에 진행한다. 분류된 보행자를 인식하고 보행자 식별 결과를 확인 할 수 있다.

3.2 기상 악화 대응 보행자 검출 시스템 요구사항

본 논문에서 제안하는 기상 악화 대응 보행자 검출 시스템을 구현하기 위해서 고려되어야 할 요구사항은 다음과 같다.

- 학습에 사용할 대규모의 이미지 빅데이터가 준비되어야 한다.
- 딥러닝 모델을 학습해야 한다.
- 특징 벡터를 분류할 분류 모델을 학습해야 한다.
- 제안하는 시스템에서 사용하는 차량용 블랙박스 영상은 낮 시간에 촬영되었으며 보행자를 식별할 수 있는 정도의 화질이다.
- 제안하는 시스템에서 사용하는 차량용 블랙박스는 권장하는 정상적인 설치 각도를 준수한다.
- 제안하는 시스템은 기존 시스템보다 검출률, 응답시간이 향상되어야 한다.
- 제안하는 시스템은 실시간성이 보장되어야 한다.

3.3 기상 악화 대응 보행자 검출 시스템 시퀀스 다이어그램

제안하는 기상 악화 대응 보행자 검출 시스템의 시퀀스 다이어그램은 그림 3.3과 같다.

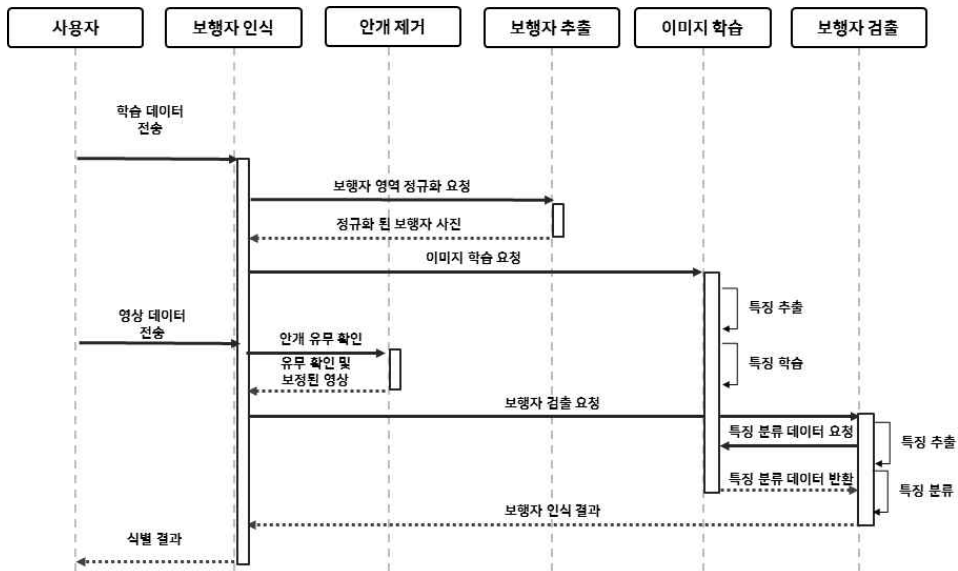


그림 3.3 기상 악화 대응 보행자 검출 시스템 시퀀스 다이어그램

사용자가 학습데이터를 보행자 인식 모듈로 전송하면 보행자 추출 모듈에다가 보행자 영역 정규화를 요청하며 그 결과로 정규화 된 보행자 영역 사진을 받아온다. 그리고 이미지 학습 모듈에 선행정보를 학습하기 위하여 학습을 요청하면 모듈 안에서 특징 추출과 특징 학습을 하게 된다.

선행 정보를 만든 후 사용자가 분석하고자 하는 영상 데이터를 전송하면 안개 제거 모듈에서 안개 유무를 확인하고 그 결과에 따라 영상처리를 시켜 보정된 영상을 반환한다. 그 후 보행자 검출 모듈에게 검출 요청을 하면 영

상의 특징을 추출하고 앞서 학습한 선행 정보를 이용하기 위하여 이미지 학습 모듈에 특징 분류 데이터를 요청하여 받아오고 보행자 분류 모델과 비교 분석하여 영상의 특징을 분류하게 되고 최종적으로 보행자 인식 결과를 얻어 보행자 인식 모듈에게 반환하면 보행자 인식 모듈이 사용자에게 식별 결과를 반환해 주어 알람이나 제어 등의 보행자 검출 결과 기반의 기능 수행을 할 수 있다.

3.4 기상 악화 대응 보행자 검출 시스템 데이터 흐름도

본 논문에서 제안하는 기상 악화 대응 보행자 검출 시스템 데이터 흐름도는 그림 3.4와 같다.

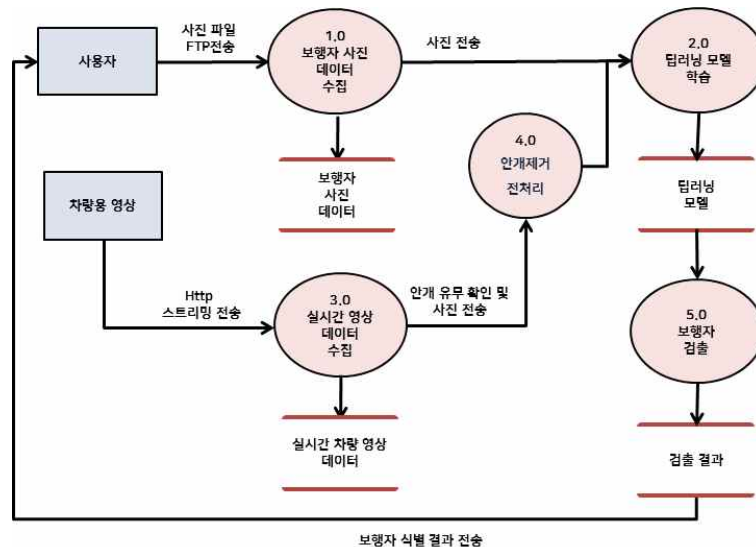


그림 3.4 기상 악화 대응 보행자 검출 시스템 DFD

시스템의 데이터 흐름의 시작은 사용자가 보행자 데이터를 학습할 것을

요구로 할 때 진행된다.

사용자가 보행자가 포함된 이미지 파일을 1.0 보행자 사진 데이터 수집 모듈로 전송하여 보행자 영역을 추출하여 보행자 사진 데이터로 수집한다. 그 다음 2.0 딥러닝 모델 학습 모듈로 사진을 전송하면 딥러닝 모델이 보행자 사진 데이터를 학습하고 딥러닝 모델 데이터를 수집하게 된다.

실제로 블랙박스 영상 속 보행자를 검출할 때 차량용 영상에서 Http 스트리밍 전송을 통해 3.0 실시간 영상 데이터 수집 모듈로 보내면 실시간 차량 영상 데이터를 수집하게 되고, 이 영상의 안개 유무를 확인 후 안개가 있을 때는 4.0 안개 제거 전처리 모듈을 통해 영상 보정을 하고 안개가 없다면 바로 다음 단계인 2.0 모듈로 가서 이미 학습된 딥러닝 모델과 특징을 비교하여 분류하고 5.0 보행자 검출 모듈로 보내준다. 여기서는 분류된 결과를 바탕으로 보행자를 검출하며 검출 결과 데이터를 갖게 된다. 그리고 최종적으로 검출 결과 데이터를 사용자에게 전송 및 출력한다.

3.5 기상 악화 대응 보행자 검출 시스템 알고리즘 설계

본 장에서는 제안하고자 하는 기상 악화 대응 보행자 검출 시스템의 알고리즘 설계에 대하여 다루고자 한다. 기존 알고리즘과 비교 및 분석을 통하여 제안하는 알고리즘의 차별성을 설명한다.

3.5.1 기존 시스템 알고리즘

기존의 기상 악화 영상에서 보행자 검출하는 알고리즘에 대하여 도식화

하면 그림 3.5와 같다.

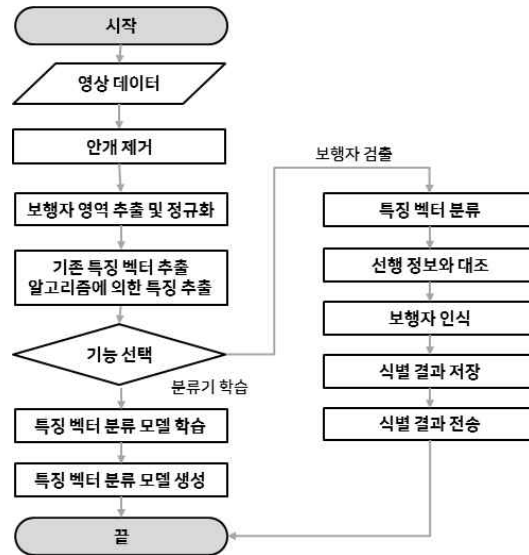


그림 3.5 기존 시스템 알고리즘

시스템이 시작하면 영상 데이터를 입력 받고 안개 제거를 한다. 그 후에 보행자 영역을 추출 및 정규화하고 기존 특징 벡터 및 알고리즘에 의한 특징을 추출한다. 그 다음 분류기 학습과 보행자 검출로 나누어져서 기능을 선택할 수 있는데 만약 선행 정보가 부족하다면 분류기 학습을 하게 된다. 분류기 학습을 하면 앞서 추출한 특징 벡터 분류 모델을 학습하고 특징 벡터 분류 모델을 생성한다.

선행 정보가 충분하다면 보행자 검출 기능을 수행하게 된다. 이는 앞서 추출한 특징 벡터를 분류하고 선행 정보와 대조함으로써 보행자를 인식하고 식별 결과를 저장 후 전송하는 과정으로 시스템을 종료하게 된다.

3.5.2 딥러닝 기반 시스템 알고리즘

제안하는 딥러닝 기반 시스템의 알고리즘은 그림 3.6과 같이 설계하였다.

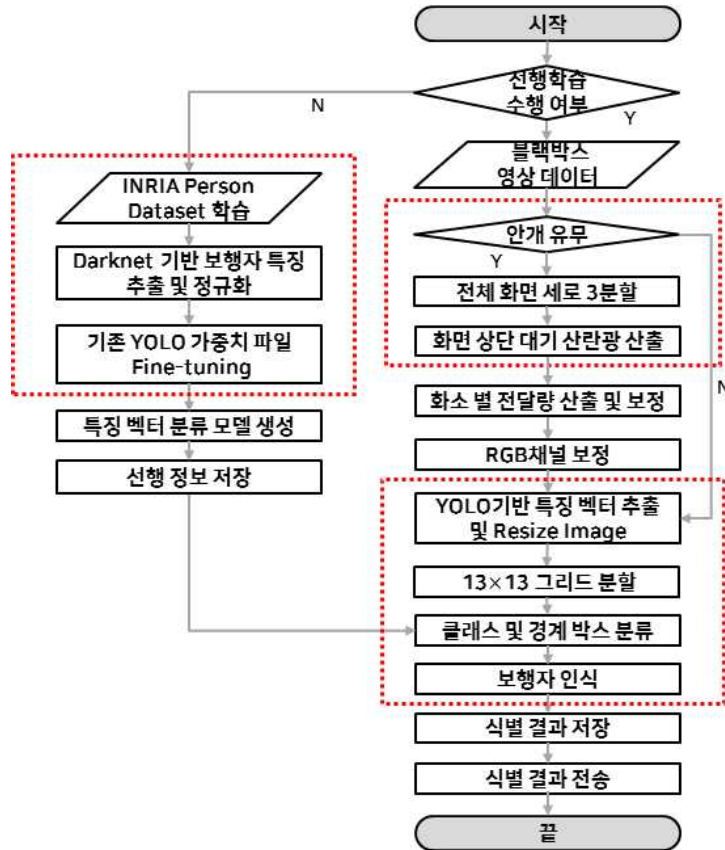


그림 3.6 제안하는 딥러닝 기반 시스템 알고리즘

시스템이 시작하면 선행학습의 수행 여부를 판단하기 위하여 가중치 파일의 여부를 판단하여 만약 학습이 필요할 경우 INRIA Person Dataset를 활용한다. 딥러닝 프레임워크인 Darknet 기반에서 보행자 특징 추출 및 정규화 과정을 거쳐 기존의 학습된 YOLO 가중치 파일을 Fine-tuning함으로써 생성된 특징 벡터 분류 모델을 선행 정보로 저장한다.

선행학습을 수행하였을 경우에는 블랙박스 영상 데이터를 입력받고 다크 채널로 안개 유무를 확인하여 안개가 있는 경우에 전체 화면의 세로 3분할 하여 상단 분할 화면에서 대기산란광을 산출한다. 이를 기반으로 화소 별 다크 채널 전달량을 산출 및 보정으로 각 RGB채널을 보정하여 안개제거를 한다. 이 후 보정된 영상을 이용하여 YOLO기반 특징 벡터 추출 및 이미지를 Resize하여 나온 최종 특징 출력단을 13×13 그리드로 분할한다. 최종 특징 출력단에서는 선행정보를 이용하여 구한 신뢰도를 기반으로 경계박스를 생성 후 추려내고 클래스를 분류하여 보행자를 인식한다. 그 후 식별 결과를 저장 후 전송하게 되면 시스템이 종료 된다.

3.5.3 제안하는 안개 제거 기법

본 논문에서 제안 하는 안개 기법은 DCP를 기반으로 활용하였다. 관련 연구에서 살펴 본 것과 같이 DCP는 안개가 없는 영상의 한 화소 값은 0에 가까운 값을 가진다는 전제에서 만들어진 기법으로 다크 채널을 만들고 거기서 대기광을 검출하는데 이 때 다크 채널에서 밝기 값이 상위 0.01% 이상인 픽셀을 후보군으로 정하게 된다. 그 이유는 대기가 아닌 밝은 빛이나 흰색 물체의 색상을 후보군으로 오인할 수 있기 때문이다.

본 논문의 시스템은 차량용 블랙박스를 기반으로 보행자를 검출하기 때문에 도로 위의 영상들이 주가 되는데 이때 도로에는 수많은 흰색 자동차와 차선이 있고, 또한 자동차의 불빛 역시 화면상에서 밝은 값을 나타내기 때문에 대기광으로 오인하게 될 확률이 높다.

본 논문은 이와 같은 문제를 해결하기 위하여 블랙박스 영상을 분할하여

대기광 검출 영역을 재정의 해주었다.

블랙박스를 설치 할 때 차량 영상의 최적화를 위해 권장 설치각도가 있다. 그림 3.7은 LG Innotek 블랙박스 사용자 설명서의 명시된 권장 설치 각도 이다.



그림 3.7 블랙박스 화면 설치 비율

다음과 같이 정상적인 블랙박스 영상 화면이라면 전체 화면의 세로 비율이 3분할되어 있다. 화면 하단의 2/3 영역은 영상을 촬영하는 차체의 일부분과 차량과 도로와 같은 다양한 객체들이 변화하여 나타나는 영역으로 이 영역에서 대기광을 검출한다면 오인할만한 요인들이 많이 등장 한다.

하지만 화면 상단의 1/3 영역은 원거리 영역으로 하늘 위주의 영상으로 대기광을 검출하기 적절한 영역이다.

따라서 본 논문에서는 대기광을 검출할 영역을 화면 상단의 1/3영역으로 제한을 두었고 이는 실시간으로 변화하는 주행 중 블랙박스 영상의 대기광의 오검출률을 줄임으로 인하여 적절한 전달량을 산출하여 효율적으로 안개 제거를 할 수 있었다.

3.5.4 제안하는 YOLO 기반 보행자 검출 방법

주행 중 블랙박스 영상에서의 보행자 검출은 무엇보다 실시간으로 빠르게 검출을 하는 것이 중요하다. 앞서 관련연구에서 살펴 본 바와 같이 YOLO는 기존의 딥 모델의 검출 방식에 비하여 학습 시간이 상당히 짧다. 따라서 본 논문에서는 보행자 검출에 YOLO알고리즘을 기반으로 활용하였다.

YOLO에서 보행자를 검출하려면 학습한 데이터의 특징을 기반으로 신뢰도를 계산하여 경계 박스를 만들어야 한다. 이 때, YOLO는 영상을 7×7 그리드로 분할하여 각 그리드 셀을 중심으로 하는 각종 크기의 객체에 대해서 경계박스 후보를 2개씩 예측 한다. 따라서 YOLO의 총 경계박스는 98개인데 이는 다른 알고리즘에 비하여 수가 적다.

또한 한 객체 주변에 여러 객체들이 있거나 셀에 비하여 작은 객체가 있을 때 검출을 잘 못한다. 예를 들면 새때처럼 조그만 객체들이 모여 있는 경우이다. 영상에서 새의 크기가 작으면 새의 크기에 비해 그리드 셀이 너무 클 수 있다.

위와 같은 특징들로 YOLO는 객체 검출 속도가 획기적으로 빠르지만 경계 박스의 부족과 그리드 셀의 크기가 크기 때문에 작거나 서로 근접한 객체에서의 검출률이 조금 떨어진다.

따라서 본 논문에서는 YOLO의 핵심 영역인 최종 출력단 그리드를 재구성화하여 이를 개선하였다. 기존의 최종 그리드 셀을 7×7 로 분할하던 것과 달리 제안하는 그리드 셀은 13×13 로 분할하여 검출하였다. 그림 3.8은 제안하는 그리드 분할 모델이다.

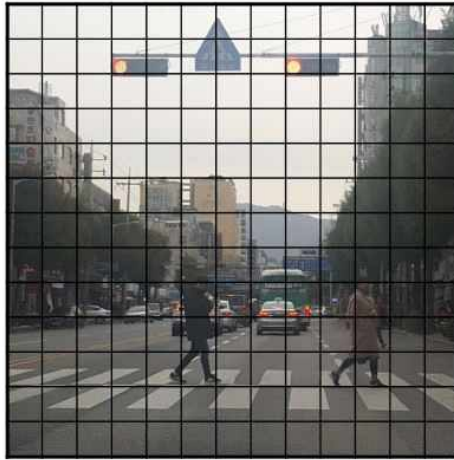


그림 3.8 제안하는 그리드 분할

제안하는 그리드 분할을 활용하면 기존에 비해서 영상의 그리드 크기에 1/4가량이 줄어 더 작은 객체들에 대한 검출률이 높아지고, 총 경계박스도 338개로 늘어 오검출률이 줄었다.

또한 영상의 특성상 가장자리보다 중앙 쪽에 객체가 등장할 가능성이 높기 때문에 홀수의 비로 분할하여 중간에 그리드 셀이 생길 수 있게 하였다.

따라서 제안하는 딥러닝 알고리즘은 보행자 검출에서 주를 이뤘던 HOG 알고리즘에서의 문제점인 환경변화의 취약점과 보행자 검출에서 딥 모델의 느린 속도를 개선하여 보행자 검출률을 높이고 검출시간을 줄일 수 있다.

IV. 기상악화 대응 보행자 검출 시스템 구현 및 평가

본 장에서는 앞에서 설계한 기상악화 대응 보행자 검출 시스템을 구축하고, 시스템 성능의 향상도를 평가하기 위하여 기존 시스템과 제안하는 시스템의 성능을 비교평가 하고자 한다. 또한, 제안하는 시스템에서 지향하는 실시간성을 보장하고자 성능을 평가한다.

4.1 구현 환경

본 논문에서 제안하는 기상 악화 시스템을 구현하기 위한 테스트 시스템의 구현 환경은 표 4.1과 같다.

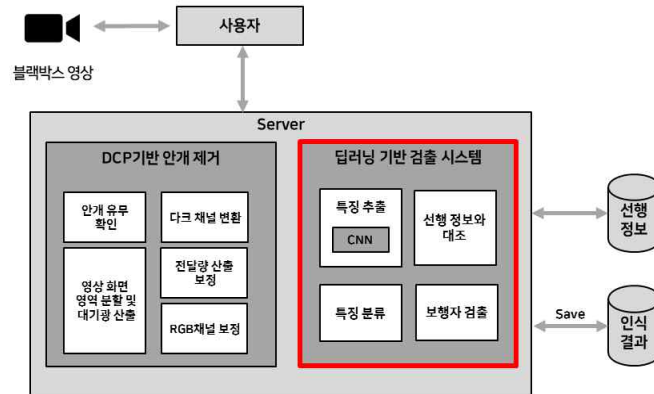
표 4.1 시스템 구현 환경

구분	구성요소	세부내용
Hardware	CPU	Intel Core i5-6200U 2.3GHz
	RAM	8G
	Graphic	NVIDIA GeForce GT 840M
Software	OS	Windows 10
	Language	C, C++, Python2
	DB	MariaDB
Library	Open Source	OpenCV, CUDA SDK8.0

4.2 기상 악화 대응 보행자 검출 시스템 구성

본 논문에서 기상 악화 대응 보행자 검출 시스템을 구현한 시스템의 구성

은 그림 4.1과 같다.



4.1 구현 시스템 구성도

제안하는 시스템은 사용자로부터 차량용 블랙박스 영상을 받아서 사용한다. 분석하고자 하는 영상을 서버에 보내면 가장 먼저 DCP 기반의 안개 제거 모듈로 보내진다. 영상을 다크 채널로 변환하여 안개의 유무를 확인 후 안개가 있을 경우 제안하는 방식으로 영상 화면 영역을 분할하여 대기광을 산출 후 화소 별 전달량을 보정하여 영상을 보정하고 안개가 없을 경우 이 과정을 거치지 않는다. 다음과 같이 전 처리 영역이 끝난 후의 영상 데이터는 제안하는 딥러닝 기반 검출 시스템 모듈로 보내진다. 본 시스템에서는 특징 추출 알고리즘으로 CNN을 사용하였으며 사전에 학습되어진 선행 정보를 이용하여 분류 및 보행자 검출을 한 번에 처리하고 데이터베이스에 인식 결과를 저장하였다. 따라서 인식 결과를 기반으로 사용자에게 필요한 보행자 검출 정보를 얻을 수 있다.

4.3 기상 악화 대응 보행자 검출 시스템 구현

기상 악화 대응 보행자 검출 시스템에서 가장 먼저 해야 할 작업은 보행자를 학습시키는 것이다. 본 논문에서는 YOLO 알고리즘을 보행자 검출에 활용할 수 있도록 개선시켜서 이미지의 특징을 추출하였으며 이는 사진 속의 객체와 클래스를 분류할 수 있는 기반의 선행정보를 제공해 준다.

기존의 YOLO 알고리즘에서는 객체 클래스 인식을 위한 표준화 이미지 Dataset인 ImageNet으로 사전학습 한 후 Pascal VOC Dataset에서 Fine-tuning 이용하여 학습시켰다. Pascal VOC에서는 사람을 포함한 클래스를 20개로 나누고 있는데 본 논문에서는 차량용 보행자 검출을 강화하기 위하여 그림 4.2와 같이 INRIA Person Dataset의 사람 사진 460장을 추가하여 초기의 가중치 파일을 재조정함으로써 보행자 특징 추출에 최적화될 수 있도록 사전학습을 시켰다.

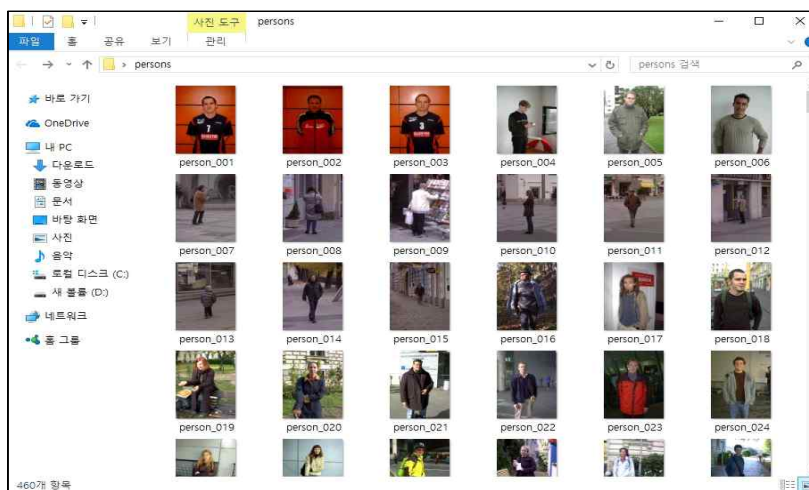


그림 4.2 INRIA Person Dataset 사람 이미지 추가 학습

선행 학습이 다 이뤄진 후에 보행자 검출을 위해서는 검출하고자하는 차

차량용 블랙박스 영상을 시스템에 업로드 해야 하는데 사용자가 직관적으로 웹에서 차량용 영상을 확인 후 선택하여 올릴 수 있도록 HTTP를 형식의 Web Client를 지원한다.

그림 4.3는 PC상에서 보행자 검출용 영상을 선택하여 업로드 할 수 있도록 하는 웹 페이지의 화면이다. 블랙박스에서 실시간으로 업로드 되는 목록들을 확인할 수 있으며 추가로 내 PC에서 동영상 영상목록을 가져오고 동영상을 시각적으로 확인 후 검출하고자 하는 영상을 선택한 후 업로드 버튼을 눌러 기상 악화 대응 보행자 검출 서버로 스트리밍 한다.

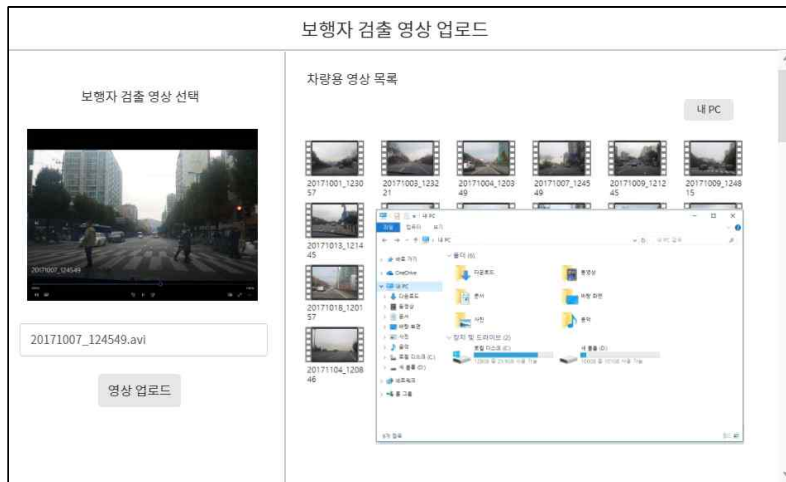


그림 4.3 보행자 검출 영상 업로드 페이지

이후 업로드 한 차량용 영상은 보행자 검출 서버로 보내지며 안개 제거 모듈에서 영상의 안개의 유무를 파악하게 된다. 안개 유무의 판단은 관련 연구에서 제시한 다크 채널 히스토그램의 통계적 분석을 기반으로 판단하며 본 논문에서의 기준 값은 다크 채널에 대한 임계값 $Th_{dark} = 50$, 안개에 대

한 임계값 $Th_{fog} = 10\%$ 로 정하였으며 만약 $Fog\ rate$ 가 Th_{fog} 보다 작을 때, 안개가 있는 것으로 판단한다.

안개가 있을 경우에는 DCP의 방식을 활용하여 영상 전처리를 하게 되며 안개가 없거나 기준 값을 넘지 않는 미미한 정도일 경우에는 전처리 없이 바로 다음 단계로 진행하게 된다.

본 논문에서는 기상 악화에 대응을 위한 보행자 검출 성능을 높이기 위하여 안개가 짙은 영상을 기준으로 시스템을 구현해보았다. 그림 4.4는 제안하는 방식의 안개 제거로 보정된 전처리 결과를 보여준다.



(a)원본 사진



(a)제안한 안개 제거 전처리 후 사진

그림 4.4 제안하는 안개 제거 전처리 결과

대기광 검출에서 오류를 줄이기 위하여 차량용 영상을 3분할 후 가장 위쪽의 영역에서 대기광을 추출하였으며 이를 기준으로 화소별 전달량을 계산 후 RGB 보정을 하여서 효율적으로 안개 노이즈를 제거 할 수 있었다.

전 처리 후 영상은 딥러닝 기반 검출 시스템 모듈로 이동하며 영상속의 특징을 추출 하고 앞서 학습한 보행자 이미지의 가중치 파일을 이용하여 클래스를 분류하고 보행자 검출을 한다. Framework는 Darknet으로 학습하였으며 그리드의 셀에서 경계박스를 제안하고, 객체 검출과 클래스 분류를 한 번에 처리하는 실시간에 처리에 능한 YOLO알고리즘의 최종 출력단의 그리드 셀을 13×13 으로 분할함으로써 기존보다 작은 보행자에도 정확하게 검출 할 수 있었다. 이와 같은 과정을 통해서 얻은 보행자 인식 결과는 데이터베이스에 저장하며 사용자의 필요의 맞게 차량의 제어, 알람 등의 목적으로 활용될 수 있다. 그림 4.5는 본 논문에서 제안한 기상 악화 대응 보행자 검출 시스템의 최종 결과물이다.



그림 4.5 제안하는 기상 악화 대응 보행자 검출 시스템 최종 결과물

4.4 기상 악화 대응 보행자 검출 테스트

본 장에서는 위에서 구현한 시스템을 기반으로 제안하는 기상 악화 대응 보행자 검출 시스템 테스트를 수행한다. 또한, 기존 연구와의 성능을 비교하여 제안하는 시스템의 향상을 제시하였다.

4.4.1 테스트 환경

본 논문에서 제안하는 기상 악화 대응 보행자 검출 시스템을 평가하기 위한 테스트 환경은 표 4.2와 같다.

표 4.2 시스템 테스트 환경

구분	구성요소	세부내용
Hardware	CPU	Intel Core i5-6200U 2.3GHz
	RAM	8G
	Graphic	NVIDIA GeForce GT 840M
Software	OS	Windows 10
	Language	C, C++, Python2
	DB	MariaDB
Library	Open Source	OpenCV, CUDA SDK8.0

4.4.2 기존 기상 악화 보행자 검출 시스템 테스트

본 논문에서 제안하는 시스템과 비교하기 위해서 기존 기상 악화 상황에

서의 보행자 검출 시스템을 조사해 보았다. 조사 결과 대다수의 안개 제거 관련 보행자 검출에서 HOG알고리즘을 사용하였다. 본 논문에는 기존 연구와의 테스트 향상도를 비교하기 위하여 관련 연구 중에서 국내에서 가장 최근인 2017년 2월 이윤석 저자 석사학위논문인 “보행자 검출을 위한 고속 안개 영상 향상 기법” 연구를 기준으로 분석하였다.

기존의 논문에서는 비가시성의 문제를 해결하기 위하여 DCP를 이용하여 전처리를 한 후 HOG 알고리즘으로 검출을 하였다. 직접 만든 Dataset으로 테스트한 결과 안개 처리 속도는 Gibson's method에 비하여 2.9배 빠르게 나왔지만 검출률은 더 낮게 나왔기 때문에 개선이 필요하다.

본 논문에서 제안한 방식과 기존의 방식을 테스트 비교하기 위하여 제안하는 시스템과 같은 환경 및 데이터로 구현 하였으며 결과는 그림 4.6과 같이 나타난다.



그림 4.6 기존 연구 보행자 검출 결과

기존 연구의 안개 제거 방식에서는 대기광 검출 영역이 전체이기 때문에 대기가 아닌 하얀 객체에서 대기광을 추출하여 전달하였기 때문에 전달량의 오류가 생겨 그림 4.6과 같이 화면이 전체적으로 어둡게 보정이 되었다.

또한 엣지 기반의 특징 추출 기법으로서 환경변수에 취약한 영향을 많이 받는 Hand-crafted Model의 HOG 알고리즘을 활용함으로써 조도가 낮고 분별력이 낮은 다음과 같은 영상 상태에서의 검출률이 떨어진다.

4.4.3 테스트 시나리오

본 장에서는 위에서 구현한 테스트 시스템을 기반으로 제안하는 기상 악화 보행자 검출 시스템 테스트를 수행한다. 제안하는 시스템의 차별성을 제시하기 위해 기존의 기상 악화 보행자 검출 시스템 성능과 비교하고자 한다. 테스트를 위한 시스템 테스트 시나리오는 다음과 같다.

<기상 악화 보행자 검출 시스템 테스트 시나리오>

- ① 선행정보를 생성하기 위해 제안하는 Dataset을 특징 추출하여 사전 학습 시킨 후 Fine-tuning하여 보행자 검출에 적합한 가중치 파일을 생성
- ② 보행자 검출 영상 업로드 Web에서 학습 시키고자 하는 차량용 보행자 영상을 확인하고 업로드
- ③ 영상 데이터를 입력 받아 다크 채널로 변환하여 안개 유무 확인 후 제

안하는 대기광 추출 방식으로 화면을 3분할하여 상단 대기광을 검출 후 화소 별 전달량 보정

④ 보행자 검출의 정확도 및 실시간 처리 테스트를 위한 것으로 블랙박스 동영상에서 보행자가 인식된 프레임을 기준으로 잡음

⑤ 제안하는 딥러닝 알고리즘 기반 보행자 검출 결과를 저장하여 100회 측정된 단위를 한 사이클로 지정하고, 동일하게 100회 반복하여 테스트 함

⑥ 정확도 및 처리 속도를 바탕으로 제안 알고리즘과 기존 알고리즘 분석 및 비교

4.4.4 테스트 절차

테스트하기 위해 제안하는 Dataset을 특징 추출하여 사전 학습 시킨 후 Fine-tuning하여 보행자 검출에 적합한 가중치 파일을 생성하고 이를 테스트의 선행정보로 이용한다.

선행 정보가 다 준비 된 이후에는 보행자 검출을 할 차량용 영상이 필요하다. 사용자가 Web Client역할을 하는 보행자 검출 영상 업로드 웹 브라우저를 통하여 검출하고자 하는 블랙박스 영상을 확인하고 업로드 한다.

이로 인하여 보행자 검출 서버에 검출용 동영상을 전달받게 되고 이를 다크 채널로 변화한 후 안개 유무를 확인하여 보정이 필요한 경우에는 제안하는 대기광 추출 방식으로 화면을 3분할하여 상단 대기광을 검출 후 화소 별

전달량을 보정 해준다.

본 시스템은 보행자 검출의 정확도 및 실시간 처리를 테스트하기 위한 것이기 때문에 블랙박스 동영상에서 보행자가 인식된 프레임의 기준으로 잡아 정보를 저장하며 제안하는 딥러닝 알고리즘 기반 보행자 검출 결과는 100회를 측정한 단위를 한 사이클로 지정하고 동일하게 100회 반복하여 테스트하였다.

위와 같은 테스트로 보행자 검출률의 정확도와 처리 속도를 그래프로 나타내고 측정할 수 있으며 테스트의 평균값을 기준으로 제안 알고리즘과 기존 알고리즘을 분석 비교하여 평가하였다.

4.5 테스트 결과 및 분석

4.5.1 검출률 테스트 결과

본 논문의 차별성을 제시하기 위해 기존 기상 악화 대응 보행자 검출 시스템과의 검출률과 처리 속도에 대해 비교 평가하고자 한다.

제안하는 시스템에서의 검출률은 영상의 전체 보행자들 중 성공적으로 검출된 보행자 비율을 의미하며 미검출률은 데이터의 보행자들 중 검출되지 못한 보행자의 비율이다. 본 논문에서는 검출률을 DR(Detection Rate)이라 지칭하며 미검출률은 MR(Missing Rate)이라 지칭한다.

그림 4.7은 기존과 제안하는 시스템의 DR을 비교 및 측정한 그래프이다.

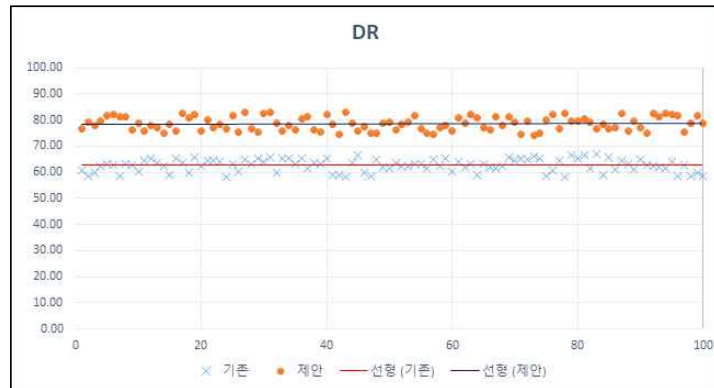


그림 4.7 제안하는 시스템 DR 비교 및 측정

그림 4.7 에서 분포되어있는 점은 1 사이클이며 이를 100회 실시하여 측정한 결과이다. 해당 그래프에서 x축은 횟수이며 y축은 DR 수치가 된다. 그래프에서 나타나는 점선은 추세선이고 100회 테스트 전체를 기준으로 제안하는 시스템 DR 평균값은 78.38%로 기존의 평균값인 62.85%보다 15.52%가량 높게 측정되었다.

그림 4.8은 제안하는 시스템의 MR을 기존의 시스템과 비교 측정한 그래프이다.

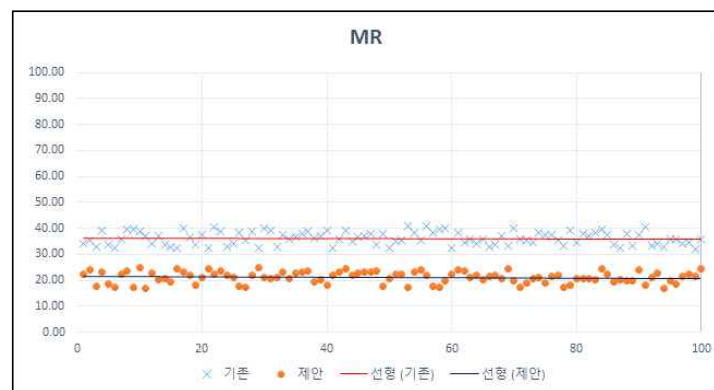


그림 4.8 제안하는 시스템 MR 비교 및 측정

그림 4.8 에서 분포되어있는 점은 1 사이클이며 이를 100회 실시하여 측정한 결과이다. 해당 그래프에서 x축은 횟수이며 y축은 MR 수치가 된다. 그래프에서 나타나는 점선은 추세선이고 100회 테스트 전체를 기준으로 제안하는 시스템의 MR 평균값은 21.14%로 기존의 평균값인 36.19%보다 15.06%가량 낮게 측정하였다.

제안하는 시스템의 결과 값의 신뢰도를 높이기 위해 표준편차를 이용하여 95%의 신뢰구간을 사용하여 평가하고자 한다. DR 수치에 대하여 100개의 표본의 평균과 표준편차는 각각 78.38, 2.64이며 MR 수치에 대하여 100개의 표본의 평균과 표준편차는 각각 21.14, 2.27이다. 이를 이용하여 DR의 평균값의 95%의 신뢰구간을 계산하면 구간 사이는 $77.86 < \mu < 78.89$ 이고, MR의 평균값의 95% 신뢰구간을 계산하면 구간 사이는 $20.69 < \mu < 21.58$ 이다.

4.5.2 시스템 실시간 처리 테스트 결과

제안하는 시스템이 실제로 적용되기 위해서는 블랙박스에서 입력받는 영상에 대해 실시간으로 처리할 수 있어야 한다.

따라서 본 장에서는 제안하는 시스템의 검출 시간을 앞서 분석한 2017년 2월 이윤석 저자의 논문인 “보행자 검출을 위한 고속 안개 영상 향상 기법” 연구와 제안하는 시스템을 본 논문의 테스트 환경에서 구현하여 비교 분석하였으며 테스트 처리 시간의 결과 값은 표 4.3와 같다. 처리 시간의 기준은 보행자가 검출이 포함 된 한 프레임을 기준으로 하였으며 총 테스트 결과의 평균값으로 비교하였다.

표 4.3 제안하는 시스템 모듈 별 처리 시간 비교 및 결과

구분	기존 시스템 (ms)	제안하는 시스템 (ms)	향상도 (%)
안개 제거	781.52	557.03	28.72
보행자 검출	40.77	25.14	38.34
전체 시스템 실행	822.29	582.17	29.20

표 4.4는 기존과 제안하는 시스템에서 사용되는 모듈 인 안개 제거, 보행자 검출영역에서의 처리 시간과 시스템 전체의 총 영상 처리 시간을 비교하였다. 제안하는 시스템의 안개 제거 모듈에서는 대기광 검출시간이 224.49ms 감소되면서 기존의 시스템 보다 28.72% 더 빨라졌으며, 보행자 검출에서는 15.63ms만큼 빠르게 측정되어 38.34% 더 빨리 검출하였다. 결과적으로 전체 시스템의 처리 속도는 240.12ms만큼 시간이 단축됨으로서 기존의 시스템에 비하여 29.20%만큼 처리 속도가 향상되었다.

V. 결론

4차 산업혁명이라는 단어가 화두가 되면서 인공지능의 대한 관심이 높아졌고 관련 기술의 소식과 제품들이 주목을 받는 요즘 인공지능은 우리의 일상에서 더 이상 낯선 기술이 아니다.

이와 더불어 최근 자율주행 자동차에 대한 관심이 증가하면서 ADAS 차량 운행 장치와 인식 기술의 연구 발전이 활성화 되고 있다.

자율주행 자동차의 기본 동작 구조는 ‘인지-판단-제어’ 순으로 본 논문의 연구는 자율주행 자동차의 ‘인지-판단’ 부분에 해당한다.

ADAS는 자율주행 자동차의 안전을 책임지는 기술로 각종 위협이나 사고에 대비해야 하는데 특히 주행 시 보행자 사고는 생명과 직결되는 치명적인 사고로 이어질 수 있으며 최근 5년간 교통안전공단이 분석한 결과 안개 낀 날에 보행자 교통사고 치사율이 맑은 날에 비해 5.3배 더 높게 나타난다. 이는 안개 낀 날과 같은 흐린 기상상태로 인한 시야 미확보가 보행자 교통사고 치사율을 높이며 이는 안전운행에 가장 큰 위험요인으로 작용한 것으로 분석된다.

따라서 주로 외부에서 날씨의 영향을 많이 받는 자율주행 자동차에서 기상 악화에 대응하여 보행자를 검출하는 연구는 꼭 필요하다.

기존 기상 악화에 환경에서 보행자 검출 연구에서 가장 많이 사용되는 검출 기법은 Hand-crafted Model의 방식으로 대표적으로 HOG 알고리즘을 활용한다. 하지만 HOG 알고리즘은 엣지에 기반을 둔 특징 추출 기법이기 때문에 보행자의 크기가 작게 보이거나 기상 악화와 같이 조도가 낮거나 분별력이 떨어지는 상황에서 검출이 힘들다는 제약 사항이 있다.

따라서 기상 악화로 인한 시야 분별력이 좋지 않은 환경변수에 대응할 수 있는 보행자 검출 연구가 꼭 필요하다고 볼 수 있다.

이와 같은 문제점을 보완하고 오검출률을 줄이기 위한 대안으로 최근에는 인공지능의 핵심기술인 딥러닝을 기반으로 하는 보행자 검출 연구가 활발히 이루어지고 있다. 최근 연구 분석으로는 Fast R-CNN과 같은 R-CNN 알고리즘이 대표적인 검출 활용한 기법들로 높은 검출률을 나타낸다. 하지만 기존의 방식보다 학습시간이 오래 걸리고 상대적으로 실시간 처리에 취약한 면이 있다.

따라서 본 논문에서는 기존의 연구에서 쓰이는 HOG알고리즘에 비하여 검출률은 높이고, R-CNN모델보다 빠른 알고리즘인 YOLO의 CNN구조를 개선하여 기존의 연구보다 응답시간을 줄이고 검출률은 높이는 시스템을 제안하였다.

또한, 주행 중 시간, 날씨에 따라 빛과 다양한 환경으로 생긴 노이즈로 인해 인식률이 떨어지는 상황에 대비하여 원본 영상에서 일정 영역에 대한 DCP 알고리즘을 이용하여 대기광을 산출하고 각 화소에 따른 빛의 전달량을 보정하여 딥러닝 학습시킴으로써 기상 악화 시에도 오검출률을 감소시켰다.

본 논문에서 제안하는 시스템을 설계하기 위해 입력받는 차량 영상 데이터를 안개제거 전처리하고 기존의 딥러닝을 통해 만든 특징 분류 모델 정보를 바탕으로 보행자 영역을 추출하여 보행자 인식 후 식별 결과를 전송하는 구조를 설계하였다. 구체적인 기능에 대해서는 시스템 구조도 및 시퀀스 다이어그램, 데이터 흐름도를 통해 시스템의 기능별 상호작용을 나타냈다.

본 논문에서 제안하는 시스템을 구현하기 위해 Windows 10에서 C,

C++, Python2를 이용하였고 보행자 영역 추출에 OpenCV, CUDA SDK8.0 오픈소스 라이브러리를 활용하여 구축하였다. 먼저 보행자 학습용 Dataset을 CNN으로 선행학습을 시켰으며, 보행자 검출용 영상을 선택하고 업로드 할 수 있는 웹페이지를 구현하였다. 또한, 윈도우 기반의 보행자 검출 서버로 스트리밍 받아 DCP를 활용하여 안개의 유무를 판단하고 영상 화소별 보정을 하여 기상 악화에 대응한 영상 전처리를 한다. 그리고 영상은 딥러닝 기반 검출 시스템 모듈로 전송하여 영상 특징을 추출하고 기존의 선행 정보를 이용하여 클래스를 분류하고 보행자를 검출하도록 시스템을 구현하였다.

본 논문에서 설계 및 구현한 보행자 검출 시스템의 테스트 결과 보행자 검출률은 78.38%, 미검출률은 21.14%, 시스템 처리 시간 582.17ms로 나타나 기존의 연구에 비하여 검출률은 15.52%가량 높이고 미검출률은 15.06% 줄였으며 응답시간은 240.12ms만큼 시간이 단축했다. 따라서 본 논문에서 제안한 딥러닝 기반 기상 악화 보행자 검출 시스템은 기존의 연구보다 검출률, 미검출률, 처리 시간 면에서 우수함을 확인하였다.

본 논문은 자율주행 자동차의 영상인식 분야에서 안개나 흐린 날과 같은 악천후로 인해 블랙박스 영상인식이 힘든 환경에 대응하여 딥러닝으로 보행자를 효율적으로 검출하고 즉각적으로 대응할 수 있는 시스템을 구축할 때 활용이 가능할 것으로 판단된다.

향후 악천후 상태에서의 흐린 영상 뿐 만이 아니라 다양한 자동차 자율주행의 방해요인에 대응할 수 있는 보행자 검출 연구가 필요할 것으로 사료된다.

참 고 문 헌

- [1] “봄철 오리무중 잦은 안개, 보행자 위험”, 교통뉴스, 2016-04-05,
<http://www.cartvnews.com/news/articleView.html?idxno=237663>.
- [2] 송수호, 현훈범, 이현, “심층 신경망을 이용한 보행자 검출 방법”, 정보
과학회논문지 제44권 제1호, 2017.
- [3] 최광연, “도로 주행 영상을 위한 안개 검출 및 제거 알고리즘”, 인하대
학교, 석사학위논문, 2017.
- [4] 안진우, “안개제거 알고리즘 개선에 관한 연구”, 숭실대학교, 석사학위
논문, 2012.
- [5] Kaiming He, Jian Sun, and Xiaoou Tang “Single Image Haze
Removal Using Dark Channel Prior,” IEEE Transactions on
Pattern Analysis and Machine Intelligence, VOL. 33, NO. 12,
2011.
- [6] 양준식, “단일 영상에서의 전달률 추정치를 이용한 실시간 안개영상 개
선방법”, 전남대학교, 석사학위논문, 2015.
- [7] 강태호, 차운주, 오웅천, 곽노은, “Dark Channel Prior 기반 안개제거
를 위한 적응형 대기광 검출”, 한국통신학회 종합 학술 발표회 논문집
(하계), 2014.
- [8] 김상균, “안개 영상에서 비가시성 개선 및 객체 추적”, 목포대학교, 박
사학위논문, 2015.
- [9] 김동훈, “인공신경망 딥러닝을 활용한 기후예측 모형 개발”, 한국기상학

- 회 가을학술대회 논문집, 2016.
- [10] 이승철, 정해동, 박승태, 김수현, “딥러닝”, 소음진동 제 27권 제3호(통권 제 147호), 2017.
- [11] Frank Rosenblatt, “Principles of neurodynamics. perceptrons and the theory of brain mechanisms,” No. VG-1196-G-8. CORNELL AERONAUTICAL LAB INC BUFFALO NY, 1961.
- [12] 마상완, “인공신경망을 이용한 압축지수 예측”, 경북대학교, 석사학위논문, 2007
- [13] 유병인, 황원준, 한승주, 이선민, 김정배, 한재준, “인간 수준에 근접한 딥러닝 기반 영상 인식의 동향”, 정보과학회, 제33권 제9호, 2015.
- [14] 이지훈, “딥러닝을 이용한 주가 예측 모델”, 숭실대학교, 석사학위논문, 2017.
- [15] 김태우, “역전파 알고리즘과 Support Vector Machine을 이용한 차량 번호판 인식 시스템 구현에 관한 연구”, 관동대학교, 박사학위논문.
- [16] 최희열, 민운홍, “Dropout 알고리즘에 대한 이해”, 한국정보과학회, 정보과학회지, 제 33권 제8호 통권 제 315호, 2015.
- [17] 장병탁, “딥하이터넷 모델”, 한국정보과학회, 정보과학회지, 제 33권 제8호 통권 제 315호, 2015.
- [18] 정경중, 최재식, “심층 순환 신경망”, 한국정보과학회, 정보과학회지, 제 33권 제8호 통권 제 315호, 2015.
- [19] 김인중, “기계학습의 발전 동향, 산업화 사례 및 활성화 정책 방향”, SPRi Issue Report(2015-017호), 2016.
- [20] 권태희, “데이터 분석 도구 성능 비교 연구:기계 학습 기반의”, 숭실대

학교, 석사학위논문.

- [21] 조경미, “인공지능-딥러닝과 CNN/RNN”, 아이리포, 아이리포 기술사회, 2017
- [22] 조나래, “딥러닝 기반 범죄자 신원 인식 시스템 설계 및 구현”, 가천대학교, 석사학위논문, 2017.
- [23] 안성만, “딥러닝의 모형과 응용사례”, 지능정보연구 제 22권 2제 2호, 2016
- [24] Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner, “Gradient-based learning applied to document recognition,” Proceedings of the IEEE, 1998.
- [25] 윤성준, “합성곱 신경망 모델을 이용한 한글 글씨체 인식”, 성균관대학교, 석사학위논문, 2017
- [26] Krizhevsky Alex, Ilya Sutskever, and Geoffrey E. Hinton. “Imagenet classification with deep convolutional neural networks,” Advances in neural information processing systems, 2012.
- [27] Szegedy, Christian, et al, “Going deeper with convolutions,” In Proceedings of the IEEE conference on computer vision and pattern recognition, 2015.
- [28] 최우석, “보행자 검출을 위한 HOG특징 추출 기법 개선”, 경희대학교, 석사학위논문, 2017.v:1409.1556, 2015.
- [29] 김병준, “Faster R-CNN 기법을 이용한 옥외 영상 차량 번호판 검출 시스템”, 전북대학교, 석사학위논문, 2016.

- [30] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, Real-Time Object Detection,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.
- [31] 이운석, “보행자 검출을 위한 고속 안개 영상 향상 기법”, 중앙대학교, 석사학위논문, 2017.

ABSTRACT

Design and Implementation of Pedestrian Detection System for Weather Deterioration Based on Deep Learning

Ham MinJeong

Advised by Prof. Seok Cheon Park

Dept. of Mobile Software

Graduate School of Gachon University

In recent years, with the increasing interest in autonomous driving systems, ADAS is responsible for the safety of autonomous vehicles, so it must be prepared for various threats and accidents. In particular, pedestrian accidents during autonomous driving can be fatal and directly linked to life, leads to serious accidents. According to Korea Transportation Safety Authority analysis for the last 5 years, the fatality rate of pedestrian's traffic accident on the foggy day was 5.3 times higher than on the clear day. It is analyzed that the uncertainty of sight according to the weather condition when driving on a misty day or in a cloudy day is the biggest obstacle to safe driving, thus this increases the fatality rate of pedestrian traffic accident. Therefore, in this paper I propose the Pedestrian Detection System for Weather Deterioration based on Deep Learning.

In order to design the system proposed in this paper, I first corrected the fog noise after checking whether there is image fog or not. After that, extract the pedestrian area features from the input image data which is given by the modified YOLO algorithm. Then, after pedestrian classification based on pedestrian image prerequisite learning, transmits the detection result to the user. As a result, I designed a system that reduces the response time and increases the detection rate.

To implement the system proposed in this paper, I used C, C++, and Python2 in Windows 10, and I use OpenCV, CUDA SDK8.0 open source libraries for pedestrian area extraction. First of all, I pre-trained pedestrian learning dataset by CNN, and then I implemented a web page that allows me to select and upload videos for pedestrian detection. And after streaming the video from a Windows-based pedestrian detection server, use Dark Channel Prior to determine whether it is a foggy day or not, and then adjust the image accordingly corresponding to deteriorating weather conditions by performing correction for each image pixel. Then, the image is moved to the deep learning based on detection system module, and the system extracts the image features, classifies the class using the existing information, and detects and proposes the pedestrian.

In order to evaluate the proposed system in this paper, I compared it with the existing pedestrian detection system for

weather deterioration. As a result of the test, the proposed system increased the detection rate by 15.52%, decreased the missing rate by 15.06%, and shortened the response time by 240.12ms compared to the previous studies. Therefore, it is confirmed that the Pedestrian Detection System for Weather Deterioration based on Deep Learning I proposed is superior in the detection rate, missing rate, and the response time to the previous studies.