

석사학위논문

차량 번호판의 실시간 인식을
위한 딥러닝 기반 OCR설계

2020년

한 성 대 학 교 대 학 원

컴 퓨 터 공 학 과

컴 퓨 터 공 학 전 공

김 바 다

석사학위논문
지도교수 허준영

차량 번호판의 실시간 인식을 위한 딥러닝 기반 OCR설계

Deep Learning Based OCR Design for Real Time
Recognition of License Plate

2019년 12월 일

한성대학교 대학원

컴퓨터공학과

컴퓨터공학전공

김바다

석사학위논문
지도교수 허준영

차량 번호판의 실시간 인식을 위한 딥러닝 기반 OCR설계

Deep Learning Based OCR Design for Real Time
Recognition of License Plate

위 논문을 공학 석사학위 논문으로 제출함

2019년 12월 일

한 성 대 학 교 대 학 원

컴 퓨 터 공 학 과

컴 퓨 터 공 학 전 공

김 바 다

김바다의 공학 석사학위 논문을 인준함

2019년 12월 일

심사위원 _____(인)
장

심사위원 _____(인)

심사위원 _____(인)

국 문 초 록

차량 번호판의 실시간 인식을 위한 딥러닝 기반 OCR설계

한 성 대 학 교 대 학 원
컴 퓨 터 공 학 과
컴 퓨 터 공 학 전 공
김 바 다

최근 차량 번호판 실시간 인식 시스템은 다양한 분야에서 상용화 되어있으며 카메라만을 이용한 저가의 임베디드 시스템을 선호한다. 이러한 시스템은 차량이외 객체의 출현이 없는 주차장 같은 환경의 경우 98%이상의 높은 인식률을 보이고 있지만 환경이 제한되지 않은 골목이나 일부 도로에서는 50~70% 사이의 인식률을 보인다. 이러한 성능의 저하는 실시간 상황에서 나타나는 조도변화나 차량이외의 물체출현 등에 의한 외부 환경변화 요인이 아웃라이어데이터가 되기 때문에 발생한다. 본 논문은 카메라 이미지 입출력이 가능한 임베디드 보드를 이용하여 딥러닝 기반 차량 번호판 OCR시스템을 설계한다. 환경이 제한되지 않은 상황에서의 OCR 인식률 향상을 위하여 준 지도학습과 시계열분석 기법을 적용한다. 준 지도학습은 일반데이터와 아웃라이어데이터를 함께 하나의 모델에 학습시킴으로써 아웃라이어를 식별하여 인식률을 향상시킨다. 시계열분석은 각각의 실시간 프레임 사이의 데이터들을 비교분석함으로써 더 신뢰성 높

은 프레임의 데이터를 선별해낸다. 성능 비교실험을 위하여 환경이 제한되지 않은 상황에서 일반적인 차량 번호판 OCR시스템과 본 논문에서 제시한 기법으로 구현된 차량 번호판 OCR시스템의 인식 정확도를 측정한다. 결과적으로 일반적인 차량 번호판 OCR시스템은 77%의 인식률을 보였으나 준 지도학습과 시계열분석이 적용된 차량 번호판 OCR시스템은 88%의 인식률을 보였다. 준 지도학습과 시계열분석 두 가지 기법을 활용한다면 인식률을 향상시킨 차량 번호판 OCR시스템의 구현이 가능하다는 것을 알 수 있었다.

【주요어】 번호판 인식, OCR, 딥러닝, 아웃라이어, 준 지도학습, 시계열분석

목 차

제 1 장 서 론	1
제 1 절 차량 번호판 OCR 동향	1
제 2 절 연구 내용	3
제 2 장 관련 연구	5
제 1 절 OCR과 번호판 인식	5
1) OCR	5
2) 번호판 인식	6
제 2 절 아웃라이어탐지	7
1) 아웃라이어탐지 개념	7
2) 아웃라이어탐지 구현기법	8
제 3 장 O C R 설 계	10
제 1 절 OCR 시스템 흐름도	10
제 2 절 딥러닝 모델 설계	11
1) 기호별 모델 구분	11
2) 준 지도학습 모델 설계	15
3) 모델 레이어 구성	19
제 3 절 시계열분석	22
1) 위치트래킹분석	22
2) 템플릿매칭분석	25
3) 해밍거리분석	27
4) 그룹스코어분석	30
제 4 장 실 험	32
제 1 절 실험환경	32

제 2 절 모델 검증평가	34
제 3 절 실시간 인식 성능평가	35
1) 분류성능평가지표	35
2) 실시간 인식 결과 분석	36
제 5 장 결 론	39
참 고 문 헌	41
ABSTRACT	44

표 목 차

[표 1-1] 라이다와 카메라	2
[표 3-1] 차종 및 용도구분등의 기호; 관할관청 기호표시	11
[표 3-2] 차량 번호판 인식 모델 라벨	17
[표 3-3] 위치 값 평균변화율과 거리 값 계산	23
[표 4-1] 딥러닝 구성 기준	32
[표 4-2] 데이터 셋 구성	33
[표 4-3] 모델 검증평가 진행 결과	34
[표 4-4] 번호판인식 OCR 시스템 성능평가.	37

그림 목 차

[그림 1-1] SoC보드 기반 임베디드 시스템 설계.	2
[그림 2-1] OCR의 지역화와 인식.	5
[그림 2-2] 번호판 인식과정.	7
[그림 2-3] 데이터 분포도 예시.	8
[그림 3-1] 번호판 인식 OCR 시스템흐름도.	11
[그림 3-2] 차량 번호판 종류.	13
[그림 3-3] 각 텍스트별 인식 모델.	13
[그림 3-4] 학습데이터.	14
[그림 3-5] 학습데이터 분포 예시.	15
[그림 3-6] 지역화 데이터.	16
[그림 3-7] 아웃라이어데이터 예시.	16
[그림 3-8] 다중클래스 분류 모델 결과출력 구조.	19
[그림 3-9] 모델 네트워크 구성.	20
[그림 3-10] 각 프레임 차량 번호판 제안 영역.	23
[그림 3-11] 위치트래킹 그래프분석.	24
[그림 3-12] 템플릿매칭 원리.	25
[그림 3-13] 이미지유사도 산출 결과 예시.	26
[그림 3-14] 해밍거리 계산을 사용한 검출 원리 예시.	27
[그림 3-15] 해밍거리 산출 결과 예시.	28
[그림 3-16] 그룹스코어분석 실시 과정 예시.	31
[그림 4-1] 오차행렬(confusion matrix).	36
[그림 4-2] 본 시스템 기준 오차행렬(confusion matrix).	36

제 1 장 서론

제 1 절 차량 번호판 OCR 동향

OCR(Optical character recognition)은 영상 내에서 기계나 사람에 의해 쓰인 문자를 컴퓨터가 인식가능한 문자로 변환하는 기술이다. 최근 딥러닝의 발전으로 인해 더욱 정확도 높은 기술로 발전해 나가고 있으며 인쇄물 판독, 신용카드 등록, 차량 번호판 인식 등 다양한 분야에서 상용화되어있다. 그중에서도 차량 번호판 인식의 경우 부가적인 센서없이 오직 카메라만을 이용한 저가의 임베디드 시스템을 선호한다. 사물을 탐지하는 센서를 사용한다면 카메라만을 사용한 제품에 비하여 훨씬 단가가 높기 때문이다. 예를 들어 사물을 탐지하는 센서의 대표적인 예인 라이다(LIDAR : Light Detecting and Ranging)는 주·야간 조도 영향에 상관없이 사물 검출이 가능하지만 카메라에 비해 가격이 상당히 비싸다(윤종호, 2017). 구체적인 라이다와 카메라의 특징 비교는 표 1-1과 같다. 반면에 카메라만을 이용한 시스템의 경우 가격이 저렴하며 부족한 성능 부분에서는 딥러닝을 활용하여 뛰어난 성능을 낼 수 있다(Lee et al., 2017). 저가의 임베디드 시스템은 SoC(System on Chip)보드 기반으로 구현된다. SoC보드는 단일 칩 하나만으로도 자기완결적 시스템을 구현할 수 있는 기술로 칩 내부에 연산장치, 주기억장치, 보조기억장치, 입출력장치를 모두 갖춘 시스템이다. 이러한 SoC보드 기반의 임베디드 시스템은 그림 1-1과 같이 카메라로 이미지 데이터의 입력을 받고 SoC보드에서 입력 이미지 분류하는 알고리즘을 처리하는 방식으로 설계된다.

[표 1-1] 라이다와 카메라

	라이다	카메라
장점	- 조도의 영향이 적음 - 간섭에 강함 - 밤낮에 무관	- 저렴함 - 해상도가 높음
단점	- 비쌈 - 해상도가 낮음	- 조도의 영향이 큼 - 간섭에 약함 - 밤낮의 영향이 큼



[그림 1-1] SoC보드 기반 임베디드 시스템 설계.

최근 상용화된 번호판 인식 시스템 중 무인주차관리시스템과 같이 차량 이외의 객체 출현이 통제되도록 환경이 제한되어있는 상황에서는 98% 이상의 높은 인식률을 보인다. 하지만 환경의 조건이 일정하지 않은 도로, 골목과 같은 상황에서는 인식률이 약 50~70% 사이로 나타나는 경우도 있다(기술이전목적 및 필요성-인식률, 2018). 이러한 낮은 인식률은 조도의 변화에 따른 잡음, 번호판과 유사한 아웃라이어데이터(outlier data)와 같은 잡음 등이 인식을 방해하기 때문에 나타난다. 전처리를 거친다고 해도 이러한 잡음들이 나타나는 부분은 상당히 다양하기 때문에 차량 번호판 인식 시스템을 구축하기 위해서는 높은 인식률과 함께 아웃라이어데이터 탐지에 대한 알고리즘이 동반되어야 한다(Daniel Lopresti, 2009).

제 2 절 연구 내용

대부분의 문자인식시스템은 입력으로 들어온 문자가 단순히 어느 클래스와 가장 유사한지를 분류하는 인식에 초점을 두었고 일반적인 번호판인식시스템 또한 마찬가지이다. 그러나 차량 이외 객체 출현이 제한되지 않은 환경에서 사용될 번호판인식시스템에서는 번호판 텍스트만 입력 값으로 들어오는 것이 아니라 그 밖의 여러 잡음 이미지가 입력으로 들어올 수 있다. 어떤 장소나 각도에서도 사용될 수도 있고 골목일 경우 차량이외에 사람이나 동물과 같은 객체가 출현할 수 있다. 예를 들면 차량 외부 부착된 전화번호광고판, 영업용차량의 상호 명, 지나가는 사람 티셔츠 내의 텍스트 성분 그리고 심지어 일정간격으로 배치된 차량 라디에이터 그릴과 같은 형태까지 인식의 입력 값 후보가 될 수 있다. 이러한 잡음이 입력 값으로 들어오게 되면 딥러닝 네트워크에서는 그나마 가장 유사한 클래스로 출력을 내놓아 오답을 만들 수밖에 없다(Yan et al., 2019).

본 논문은 환경이 제한되지 않은 상황에서의 번호판 인식 시스템에 알맞은 OCR시스템을 설계한다. 입력으로 들어오는 잡음을 식별하는 아웃라이어탐지(outlier detection)를 행하기 위해서 정상적인 데이터와 오답 데이터를 함께 학습 시키는 준 지도학습(semi-supervised learning) 기법으로 모델 학습을 진행 한다. 인식을 향상 위하여 모델 예측 전후과정에서는 실시간인식 시스템의 이점을 활용하여 시계열 데이터를 분석한다. 위치 값, 템플릿매칭 값, 해밍거리, 그룹스코어 등의 시계열 데이터를 분석하며 각 데이터에 적합한 알고리즘을 구현함으로써 분류데이터의 정확도를 높인다. 실험으로는 준 지도학습의 아웃라이어탐지 성능측정을 위하여 지도학습모델과 준 지도학습 모델의 검증데이터셋에 대한 정확도를 비교한다. 실시간 인식 성능을 평가하기 위해서는 일반 지도학습으로 구현된 OCR시스템과 본 논문에서 제시하는 준 지도학습과 시계열탐지기법이 적용된 OCR시스템의 성능을 분석한다.

본 논문은 2장에서 관련연구인 OCR과 기존 번호판인식과정 그리고 아웃라이어탐지에 대한 방법론에 대해서 알아본다. 3장에서는 본 논문에서

제시하는 번호판 OCR에 적합한 시스템 흐름도의 설명, 딥러닝 모델 구성 및 설계, 인식률을 높이기 위한 시계열분석 등을 설명한다. 4장에서는 본 논문에서 제시하는 번호판 인식 시스템에 대한 실험과 성능평가에 대한 결과를 제시하고, 마지막으로 5장에서는 결론 및 향후 연구를 논의하는 것으로 논문을 마무리 한다.

제 2 장 관련연구

제 1 절 OCR과 번호판 인식

1) OCR(Optical character recognition)

OCR은 그림 2-1과 같이 크게 두 가지 과정의 종합체이다(Mitthe at al., 2013). 먼저 첫 번째는 이미지 내에서 대략적인 문자들의 위치가 어디 있는지를 찾는 것을 의미하는 지역화(Localization)과정이다. 지역화 과정은 입력으로 들어온 이미지에 대해서 문자들의 위치 값을 출력 값으로 내놓는다. 두 번째는 찾은 문자들이 어떤 클래스 내에 속하는 문자인지를 분류하는 인식(Recognition)과정이다. 인식과정은 지역화 과정에서 얻은 문자의 위치 값과 이미지를 입력으로 받고 각각의 문자열이 속한 클래스를 출력 값으로 내놓는다. 즉 정리하자면 OCR은 문자의 위치를 찾는 지역화과정과 문자를 분류하는 인식과정이 합쳐진 기술이라고 할 수 있다 (Neumann and Matas, 2010).



(a) 지역화(Localization)



(b) 인식(Recognition)

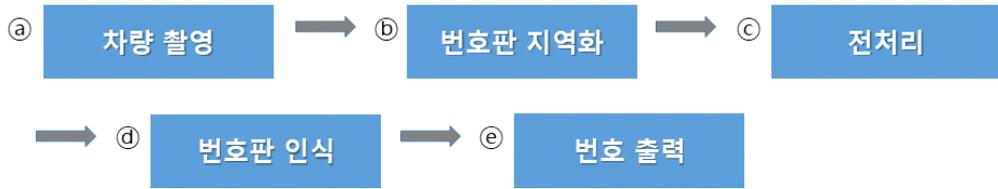
[그림 2-1] OCR의 지역화와 인식.

전통적인 OCR은 주로 Tesseract를 기반으로 구현되어 왔다. Tesseract OCR은 2006년 이후 구글이 주도하여 개발하고 있는 라이브러리이다. 100개 이상의 언어를 인식할 수 있으며 다양한 출력 포맷을 지원

하기 때문에 많은 용도의 OCR 개발에서 사용되었다. 2015년도 이후부터는 딥러닝 기반의 CNN을 사용한 OCR방식이 컴퓨터 비전 기반의 Tesseract OCR 보다 더 높은 정확도를 보이는 사례(Acharya at al., 2015; Zhong at al., 2015)가 늘면서 주로 CNN을 OCR엔진으로 사용하기 시작했다. 이에 대응하기 위하여 Tesseract도 4.x버전부터는 딥러닝 기반의 LSTM 인식 엔진을 구현하였다(Balci at al., 2017). 따라서 현재는 문자열의 연속적인 정보를 담는 특징을 인식하기 위한 OCR은 LSTM기반 엔진의 Tesseract4.x를 주로 이용하고 있으며, 문자열이 비연속적인 정보이며 단일문자 인식의 성격이라면 CNN을 주로 이용한다. 본 논문의 차량 번호판 인식 문자열은 비연속적인 단일문자 성격을 가졌으므로 CNN기반의 OCR을 구현한다.

2) 번호판 인식

번호판인식을 위한 OCR도 역시 앞 단락에서 언급한 과정과 상당히 유사하며 세부적으로 과정을 나누었을 때 총 5단계의 과정(그림 2-2)을 가진다(송영준 등, 2006; Sonavane at al., 2015; Atiwadkar at al., 2015). 먼저 첫 번째는 차량 촬영(그림 2-2 (a)) 과정으로 번호판이 보이는 정면 또는 후면의 번호판 텍스트가 담긴 영상을 확보한다. 두 번째는 번호판 지역화(그림 2-2 (b))과정으로 이미지 내에서 문자가 있는 위치 즉 번호판의 위치를 추출 과정이다. 세 번째는 추출된 번호판 위치 내에 있는 문자열을 개별적인 문자로 나누고 기울기와 음영을 보정, 잡음을 제거하는 전처리(그림 2-2 (c)) 과정이다(박성실과 박세호, 2017). 네 번째는 개별 문자가 구체적으로 어떤 문자인지는 분류하는 번호판 인식(그림 2-2 (d))과정이다. 마지막으로 다섯 번째 번호 출력(그림 2-2 (e))과정은 인식된 문자들을 하나의 문자열로 종합하여 전체 번호판 텍스트를 출력하는 과정이다. 본 논문에서는 그림 2-2의 인식과정을 딥러닝 기반으로 구현하여 본 논문에서 제시한 기법을 적용한 시스템과 비교하는 실험을 4장에서 진행한다.



[그림 2-2] 번호판 인식과정.

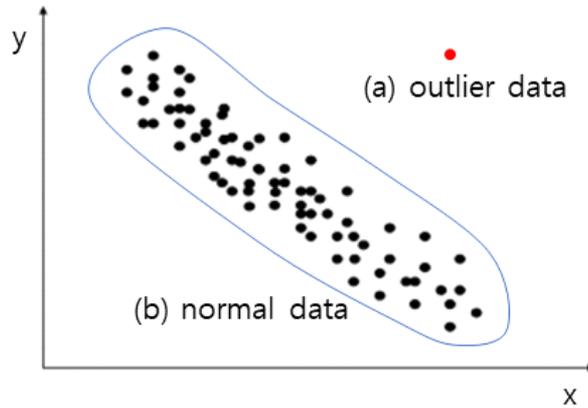
제 2 절 아웃라이어탐지

1) 아웃라이어탐지 개념

아웃라이어탐지(outlier detection)란 데이터 분석 과정에서 보편적이지 않은 데이터를 식별하는 것을 의미한다(Ahmed at al., 2016). 데이터 분포도 상에서 보편적이지 않은 데이터는 그림2-3 (a)와 같은 데이터를 말한다. 일반적인 데이터의 관측치는 그림 2-3 (b)와 같이 군집을 이룰 수 있고 일련의 상관관계가 있지만 그림 2-3 (a)와 같은 이상치는 군집에서 벗어나 상당히 다른 양상의 분포를 지닌다. 따라서 아웃라이어탐지의 기본 원리는 바로 이러한 이상점을 식별하는 것이다. 아웃라이어탐지는 이상탐지(anomaly detection), 편차탐지(deviation detection), 예외마이닝(exception mining) 등 다양한 용어로도 불리며 이러한 모든 용어들은 이상점을 식별한다는 동일한 의미를 갖는다.

딥러닝의 예측기능에서 아웃라이어탐지는 상당히 중요하다. 일반적인 딥러닝의 예측기능은 주어진 입력 값을 기준에 학습되어있는 라벨들 중에서 가장 유사하다고 판단한 라벨을 출력 값으로 내놓는 방식이다. 만약 아웃라이어탐지가 제대로 되어있지 않을 때 아웃라이어데이터가 입력으로 들어온다면 딥러닝 모델은 이미 존재하는 클래스 라벨 중 그나마 유사한 라벨로 출력 값을 내놓을 수밖에 없다. 이러한 출력 값은 무조건 오답일

수밖에 없으므로 아웃라이어데이터가 많아지는 것에 비례하여 인식정확도가 떨어진다. 실제 상용화 되는 제품에서는 어떤 아웃라이어데이터가 입력될지 알 수 없으므로 얼마나 아웃라이어탐지 정확도가 높은가에 따라 성능이 결정될 수 있다.



[그림 2-3] 데이터 분포도 예시.

2) 아웃라이어탐지 구현기법

아웃라이어탐지는 여러 가지 기법으로 구현될 수 있다. 대표적으로 크게 세 가지 기법을 통하여 구현 가능하다(Chandola at al., 2009; Domingues at al., 2018). 첫 번째는 지도학습(Supervised learning)을 통한 방법이다. 원래의 학습데이터 이외의 아웃라이어데이터를 수집한 후 그 데이터들을 라벨링하고 학습 시켜 별도의 아웃라이어데이터의 예측을 목적으로 한 모델을 따로 만드는 접근 방식이다. 하지만 지도학습 기반 아웃라이어탐지의 경우 아웃라이어데이터의 수집여부와 클래스 구분이 확실한 상황에만 사용 가능하기 때문에 제한적인 상황에만 국한하여 구현가능하다.

두 번째 방법은 비 지도학습(Unsupervised learning)이다. 비 지도학습은 데이터 내부의 패턴과 관계를 발견하는 알고리즘으로 정답이 주어지지

않더라도 데이터간의 분류가 가능하다. 이러한 원리로 학습데이터와 아웃라이어데이터 사이의 관계를 파악하고 분류할 수 있다. 그러나 아웃라이어데이터가 다양한 양상으로 분포되어있는 경우 일반적인 지도학습 방법으로는 데이터를 세밀하게 군집화 할 수 없기 때문에 다른 특별한 기법 없이는 아웃라이어탐지 성능에 제한이 있다.

세 번째 방법으로는 통계적 방법을 이용하는 방식이 있다. 통계적 방법은 확률 모델, 시계열분석(Anderson, 2011) 등 여러 알고리즘이 있다. 예를 들면 확률 모델 같은 경우 아웃라이어데이터가 논리적으로 함께 발생하거나 조합될 수 없는 데이터에 대한 패턴 분석 또는 확률적인 계산을 이용한 탐지방법이다. 시계열분석과 같은 경우 시간 흐름에 따라 값이 변화하는 데이터를 분석하여 이상치를 식별하는 방법이다. 이와 같은 통계적 방법은 아웃라이어탐지에 유용한 방법이 될 수 있지만 대부분의 문제를 해결할 수 있는 일반화된 알고리즘이 딱히 존재하지 않는다는 단점이 있다.

본 논문에서는 지도학습을 응용한 방식인 준 지도학습을 기반으로 아웃라이어를 탐지한다. 준 지도학습은 별도의 아웃라이어 예측 모델을 두는 지도학습 방법과 달리 하나의 모델 내에 일반 데이터와 아웃라이어데이터를 함께 학습하여 데이터를 예측하는 방식이다. 또한 실시간 인식이 목적인 본 논문에서는 시계열데이터를 분석하는 것으로 더 신뢰성 있는 데이터를 얻어낼 수 있다. 비록 시계열분석은 하나의 일반화된 아웃라이어탐지 알고리즘을 구현할 수는 없지만 아웃라이어 탐지 기능을 보조하면서 신뢰성 있는 데이터를 얻어내어 인식률을 향상시킬 수 있다.

제 3 장 OCR 설계

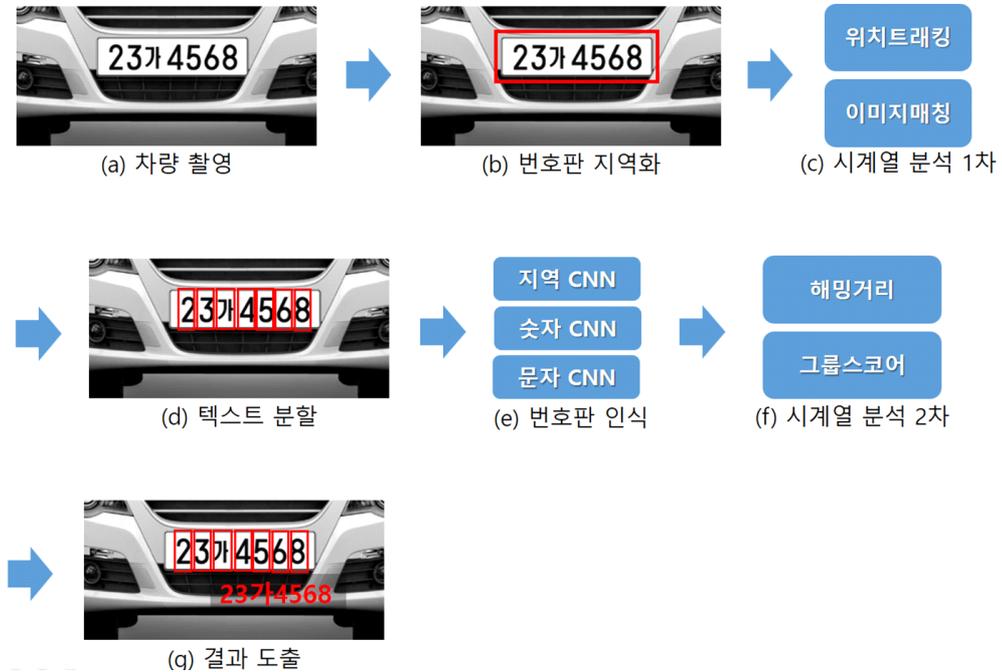
제 1 절 OCR 시스템 흐름도

번호판 인식을 위한 OCR을 설계하기 위하여 이번 장에서는 CNN 학습 모델을 구성하고 시계열분석을 적용하여 인식률을 향상시키는 방법에 대하여 설명한다. 먼저 CNN 학습 모델은 번호판 자리에 따른 기호별 모델들을 구성하여 인식하도록 설계하며, 모델학습과정에서 아웃라이어데이터를 탐지하기 위한 준 지도학습기법을 이번 장 2절에서 제시한다. 준 지도학습은 지도학습 기법을 응용한 방식으로 하나의 모델에 일반데이터와 아웃라이어데이터를 모두 학습시키는 방식이다.

시계열분석은 2장 2절에서 언급했듯이 아웃라이어탐지만을 위하여 일반화된 알고리즘이 존재하지는 않기 때문에 하나의 완전한 아웃라이어탐지방법이 되기는 어렵다. 하지만 실시간 변화량을 갖는 데이터들을 분석하는 것으로 더 신뢰도 높은 데이터를 선별해내고 나아가 아웃라이어탐지를 보조하는 역할을 수행할 수 있다. 이번 장 3절에서는 본 시스템에 적용시킨 시계열분석기법 알고리즘 4가지를 제시한다. 번호판 데이터 인식 전 지역화 데이터를 기반으로 분석하는 위치트래킹과 템플릿매칭 분석이 있고, 번호판 데이터의 딥러닝 인식 후 얻는 데이터를 기반으로 해밍거리와 그룹스코어 분석이 있다.

2장 그림2-2에서 언급한 번호판 인식과정에서 추가적으로 본 논문의 제시 기법을 적용한 시스템흐름도는 그림 3-1과 같다. 번호판 인식은 가장 먼저 카메라를 이용하여 차량 이미지를 촬영(그림3-1 (a))한 후 지역화 알고리즘을 통하여 번호판 영역을 추출(그림3-1 (b))한다. 그 후 해당 번호판의 올바른 추출 여부를 확인하기 위한 1차 시계열분석인 위치트래킹과 템플릿매칭을 실시(그림3-1 (c))한다. 1차 시계열분석이 끝나면 번호판 영역내의 각각의 텍스트들을 분할(그림3-1 (d))한다. 이러한 분할된 텍스트는 각 텍스트에 맞는 CNN모델에서 입력 값이 되어 인식(그림 3-1 (e))된다. 분할된 텍스트들을 기반으로 2차 시계열분석인 해밍거리

분석과 그룹스코어분석을 실시(그림3-1 (f))하고 해당 과정을 마치면 전체의 번호판 텍스트 결과를 도출(그림3-1 (g))하게 되는 구조이다.



[그림 3-1] 번호판 인식 OCR 시스템흐름도.

제 2 절 답러닝 모델 설계

1) 기호별 모델 구분

표3-1(차종 및 용도구분등의 기호; 관할관청 기호표시, 2018)에 따르면 국내 차량 번호판의 텍스트는 차종별, 용도별, 지역별 기호가 각각 다르다. 차종별 기호는 0~9로 이루어진 숫자 기호이며, 용도별 기호는 총 40종류의 단일 문자기호 그리고 지역별 기호는 총 17종류의 두 글자 지역명을 사용한다.

[표 3-1] 차종 및 용도구분등의 기호; 관할관청 기호표시

구 분		분 류	기 호	
차종별	승용 자동차	비사업용	100-699	
		대여사업용		
		일반사업용	01-69	
			승합자동차	70-79
			화물자동차	80-97
			특수자동차	98, 99
용도 별	비사업용 (SOFA 자동차 포함)	자가용 (관용포함)	가,나,다,라,마,거,너,더,러,머,버, 서,어,저,고,노,도,로,모,보,소, 오,조,구,누,두,루,무,부,수,우,주	
	자동차운수 사업용	일반용	바, 사, 아, 자, 배	
		대여사업용	허, 하, 호	
관할 관청 지역 별	자동차운수 사업용	일반용	강원, 경기, 경남, 경북, 광주, 대구, 대전, 부산, 서울, 세종, 울산, 인천, 전남, 전북, 제주, 충남, 충북	

이러한 기호들을 조합하여 만들어진 차량번호판은 그림3-2와 같이 크게 비사업용과 사업용(일반), 사업용(대형)으로 나뉜다. 비사업용은 차종별, 용도별 기호를 종합하여 7자리 텍스트로 구성된 번호판이다. 사업용은 지역별, 차종별 그리고 용도별 기호를 종합하여 총 9자리 텍스트로 구성된 번호판이다. 사업용의 일반과 대형은 지역명이 수직인지 수평인지의

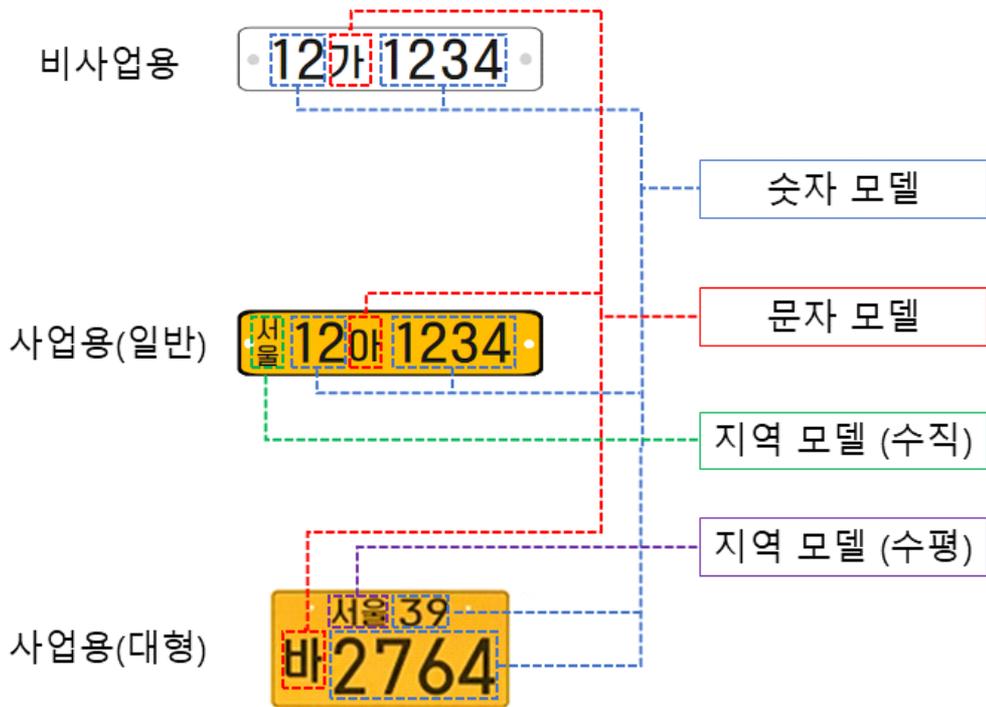
차이점이 있다.



[그림 3-2] 차량 번호판 종류.

차량 번호판들의 각 텍스트자리는 일정한 종류로 기호가 정해져있다. 숫자, 문자, 지역명(수직) 그리고 지역명(수평)으로 총 4가지 기호로 분류할 수 있다. 이렇게 분류된 기호들을 이용하여 서로 다른 4가지 기호별 CNN 모델을 학습시킨다. 기호별 인식 모델을 각각 사용하는 방법은 각 텍스트 자리별로 인식에 필요한 클래스만 학습을 시킨 모델을 사용 할 수 있게 해준다. 따라서 불필요한 클래스의 출력 값을 내놓는 상황을 배제시킬 수 있기 때문에 더 높은 정확도의 예측이 가능하다. 그림 3-3은 번호판 텍스트를 각각의 자리에 적합한 모델을 사용하여 인식하는 개념을 설명한 그림이다. 예를 들면 비사업용의 경우 3번째 텍스트 자리만 단일 문자기호를 학습시킨 모델을 인식에 사용하고 나머지 텍스트 자리들은 숫자기호를 학습한 모델을 인식에 사용한다. 사업용 번호판의 경우도 제일 첫 앞 글자는 지역별 기호를 학습한 모델을 사용하여 인식하고 그 이후의 텍스트는 비사업용 번호판의 인식과 동일한 원리로 인식한다.

그림 3-4는 각 모델들의 학습으로 사용되는 차량 번호판 텍스트 데이터들이다. 숫자모델은 그림 3-4 (a)의 차종별 기호 데이터인 0~9까지의 숫자데이터들로 학습된다. 문자모델은 그림 3-4 (b)의 용도별 기호 데이터인 총 40종류의 단일 문자데이터들로 학습된다. 지역명모델은 수직과 수평타입이 있으며 각각 그림 3-4 (c)와 그림 3-4 (d)의 지역명 데이터들로 학습된다.



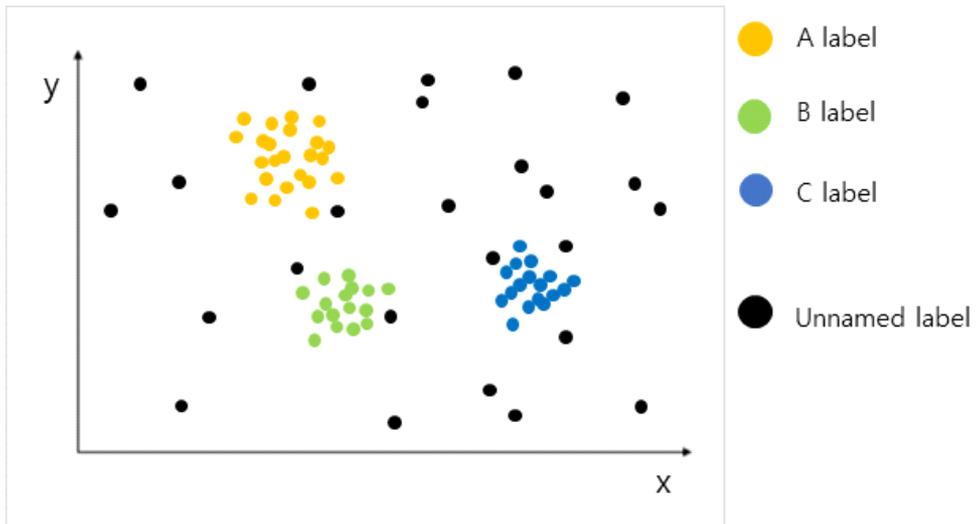
[그림 3-3] 각 텍스트별 인식 모델.



[그림 3-4] 학습데이터.

2) 준 지도학습 모델 설계

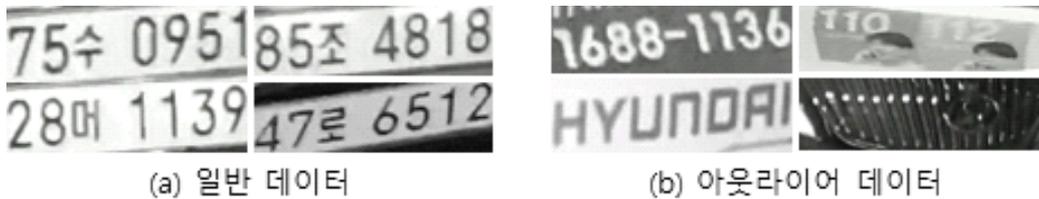
일반적인 지도학습은 라벨링된 데이터를 학습하여 모델을 만드는 방법이다. 하지만 준 지도학습(semi-supervised learning)은 라벨링된 데이터와 라벨링이 되지 않은 데이터를 모두 학습 모델의 훈련데이터로 사용하는 방법을 말한다(Görnitz at al., 2013). 예를 들면 그림3-5에서 지도학습을 행할 경우 A, B, C라벨을 이용하여 학습을 진행하지만 준 지도학습은 A, B, C 그리고 Unnamed label을 이용하여 학습을 진행한다. 이러한 방법은 입력 값에 대한 데이터 예측 시 라벨링이 되지 않은 부류의 아웃라이어데이터를 구분해내는 것에 목적이 있다.



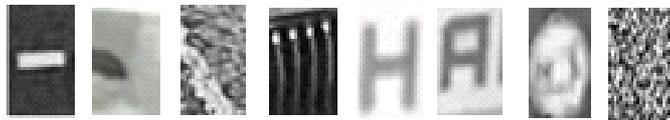
[그림 3-5] 학습데이터 분포 예시.

차량 번호판의 실시간 인식을 목적으로 하는 본 논문의 경우 실시간 인식의 특징상 수많은 아웃라이어데이터가 입력 값으로 동반된다. 차량 이외의 객체 출현이 제한된 주차장 번호판 인식 같은 경우 기후변화, 조도 변화 그리고 차량 이외의 객체 출현과 같은 외부 환경요인이 거의 없지만 상황이 제한되지 않은 골목이나 일부 도로에서는 외부 환경 요인의 변화

가 많다. 이러한 모든 요인은 아웃라이어데이터가 되어 딥러닝 모델 예측 기능 시 입력데이터가 되어 성능을 저하하는 원인이 된다. 그림3-6은 지역화 과정에서 추출된 번호판 영역데이터들이다. 그림 3-6 (a)같은 경우 일반적인 차량 번호판데이터지만 그림 3-6 (b)는 차량 번호판과 유사한 특징을 갖는 아웃라이어데이터이다. 텍스트 성격을 갖는 전화번호, 로고뿐만 아니라 텍스트의 에지변화와 유사한 특징을 갖는 차량 라디에이터그릴과 같은 물체들 또한 아웃라이어데이터가 될 수 있다. 이러한 수많은 양상의 아웃라이어데이터를 분류하기 위하여 본 논문에서는 그림 3-7과 같이 도로상황에서 자주 접할만한 아웃라이어데이터인 전화번호 광고판의 ‘-’ 기호, 번호판에 포함되지 않은 텍스트 종류들, 차량의 일부분 그리고 랜덤 노이즈 이미지 등을 모델 내의 하나의 클래스로 학습시킨다.



[그림 3-6] 지역화 데이터.



[그림 3-7] 아웃라이어데이터 예시.

본 논문의 4가지 딥러닝모델은 각각 모델학습에 필요한 일반데이터들과 unnamed 클래스 내에 아웃라이어데이터를 학습시킨다. 구체적인 모델별 학습 라벨은 표 3-2와 같다. 숫자모델의 경우 0~9까지의 숫자와 unnamed클래스를 합하여 총 11개의 클래스를 가진 모델로 학습된다. 문자모델은 40가지 번호판 문자와 unnamed클래스를 합하여 41개의 클래스

그리고 지역명모델 수직과 수평은 17개의 지역명과 unnamed 클래스를 합하여 총 18개의 클래스를 가진 모델로 학습된다. unnamed 클래스로 학습될 아웃라이어데이터는 번호판과 유사한 특징의 데이터뿐만 아니라 타 모델들의 일부 일반데이터들 또한 unnamed 클래스로 학습시킨다. 그림3-1(d)와 같은 텍스트 분할 과정이 잘못되어 다른 모델로 입력 값이 들어가는 경우라도 아웃라이어데이터로 판단 할 수 있도록 만들기 위해서이다. 예를 들면 숫자모델의 unnamed 클래스에는 아웃라이어데이터와 문자모델 데이터, 지역명모델 데이터의 일부도 함께 학습하는 것을 의미한다. 이러한 아웃라이어데이터 학습과정을 거친 각 모델의 unnamed 클래스는 다른 클래스에 비해서 일반화된 특징 없이 학습된 클래스라고 할 수 있다. 이러한 특징은 아웃라이어데이터가 발생했을 경우 일반화되어있는 클래스보다는 다양한 양상의 일반화되어있지 않은 unnamed 클래스로 아웃라이어데이터를 분류할 확률이 높게 나타난다. 즉 unnamed 클래스로 분류된 데이터를 아웃라이어데이터로 판단하는 방식으로 준 지도학습 모델을 활용 할 수 있다.

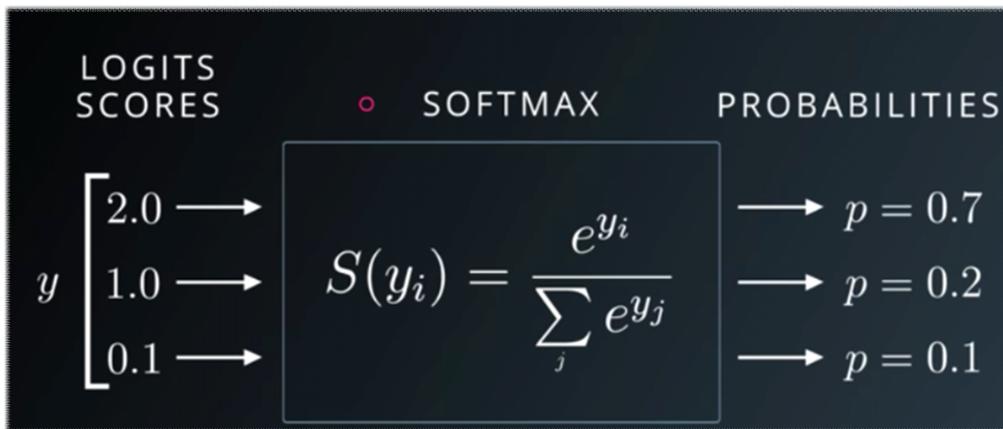
[표 3-2] 차량 번호판 인식 모델 라벨

구분	숫자모델	문자모델	수직 지역명모델	수평 지역명모델
라벨	0	가	강원	강원
	1	나	경기	경기
	2	다	경남	경남
	3	라	경북	경북
	4	마	광주	광주
	5	거	대구	대구
	6	너	대전	대전
	7	더	부산	부산
	8	러	서울	서울
	9	머	세종	세종
	unnamed	버	울산	울산
	서	인천	인천	

		어	전남	전남
		저	전북	전북
		고	제주	제주
		노	충남	충남
		도	충북	충북
		로	unnamed	unnamed
		모		
		보		
		소		
		오		
		조		
		구		
		누		
		두		
		루		
		무		
		부		
		수		
		우		
		주		
		바		
		사		
		아		
		자		
		배		
		히		
		하		
		호		
		unnamed		
계	11	41	18	18

3) 모델 레이어 구성

일반적인 딥러닝의 다중클래스 분류 모델은 마지막 레이어에서 softmax함수를 이용하여 그림 3-8과 같이 로짓 스코어를 정규화된 확률 값으로 바꾸어 출력하는 구조이다(Jang at al., 2016). 이러한 확률 값은 해당 예측이 어떤 클래스와 얼마나 가까운지를 공정하게 평가하여 하나의 예측 결과를 도출해낸다는 점에서 많이 활용된다.

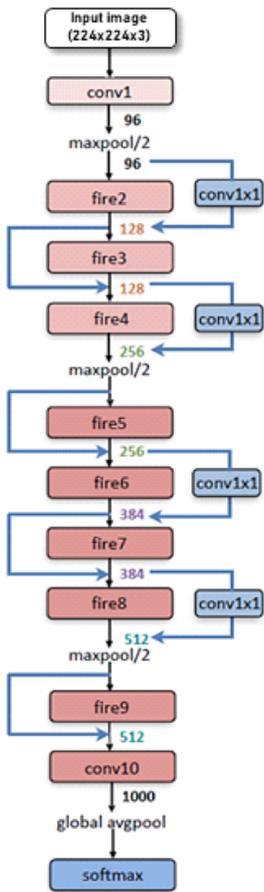


[그림 3-8] 다중클래스 분류 모델 결과출력 구조.

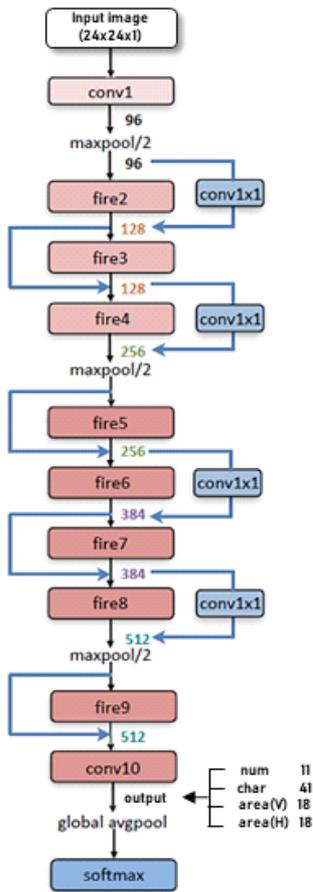
본 논문의 딥러닝 학습모델 역시 softmax를 사용하지만, 클래스 예측 기능 사용 시 softmax를 사용하지 않는 별도의 예측모델을 구성한다. 학습모델과 예측모델을 분리하는 이유는 예측기능의 결과 값으로 나온 스코어를 타 프레임 예측 스코어와 비교분석하기 위해서이다. 스코어분석에 대한 자세한 내용은 3장 3절 4)에서 추가적으로 다룬다. 타 예측결과와의 상대적인 비교분석은 softmax의 결과 값인 확률보다는 로짓 스코어를 통하여 비교분석하는 것이 공정하다. 왜냐하면 softmax를 통해 산출된 확률은 로짓 스코어의 크기에 관계없이 모든 벡터의 합이 1로 정규화된 값인데, 이러한 값은 해당 예측 내에서 결과를 도출하는 과정에는 적합할지라도 타 예측 결과와 상대적인 비교척도로 사용되기는 적합하지 않기 때문

이다. 뿐만 아니라 softmax 함수의 지수함수계산 특성상 스코어 값 단위가 크고 각 입력 값의 격차가 심한 경우 실질적인 확률 값을 출력할 수 없다. 입력 값이 크거나 격차가 심한 정도에 따라 지수함수 기울기가 증가하여 더 큰 차이를 발생시키기 때문에 가장 스코어가 높았던 벡터는 1이 되고 나머지 벡터들은 0으로 확률 값을 출력되기 때문이다.

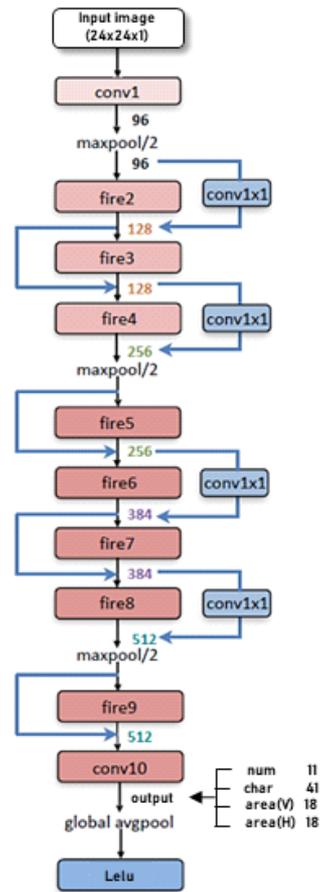
본 논문의 모델들은 임베디드 SoC보드에서 동작시키기 위하여 경량 설계방식을 적용한 그림 3-9 (a)와 같은 SqueezeNet v1.1 모델을 기반으로 한다. SqueezeNet v1.1은 전체 컨볼루션과 입력채널의 파라미터 수를 최대한 줄이고 다운샘플링을 통하여 이미지 정보를 압축시키는 경량화 전략을 사용하였으며 정확도를 최대한 보존시키는 것이 가능하다는 장점이 있다(Iandola et al., 2016). 본 논문 모델들의 입력 이미지는 24x24 크기의 흑백영상을 사용하므로 해당 크기에 맞도록 입력레이어를 구성한다. 출력레이어는 각 구분모델인 숫자, 문자, 수직지역명 그리고 수평지역명의 클래스 개수에 맞게 각각 11, 41, 18, 18로 구성해준다. 구체적인 학습모델의 네트워크 구성은 그림 3-9 (b)와 같으며, 예측모델의 네트워크 구성은 그림 3-9 (c)와 같다. 예측모델은 가장 마지막 레이어를 softmax가 아닌 relu로 변경함으로써 스코어 형태로 출력 값을 얻을 수 있다.



(a) squeezeNet v1.1



(b) 학습모델



(b) 예측모델

[그림 3-9] 모델 네트워크 구성.

제 3 절 시계열분석

1) 위치트래킹분석

위치트래킹(Yilmaz at al., 2006)은 차량 이동에 따른 각 프레임사이 번호판 영역 위치의 평균변화율 식 (1)과 거리 값 식 (2) 변화치를 이용하여 데이터를 분석한다. 만약 특정 프레임에서 아웃라이어데이터가 발생한다면 평균변화율과 거리 값 변화치가 다른 주변 값에 비하여 적절하지 못하기 때문에 특이치로 관측될 것이고 해당 값을 아웃라이어데이터로 판단하는 방식이다. 딥러닝 기반 인식 이전에 실시하기 때문에 아웃라이어 데이터를 사전에 식별하고 연속적인 인식에 있어서 하나의 그룹에 해당하는 올바른 데이터를 식별하여 신뢰성 있는 데이터 그룹을 구성할 수 있다.

$$f'(a) = \lim_{h \rightarrow 0} \frac{f(a+h) - f(a)}{h} \quad (1)$$

$$d_{distance} = |\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}| \quad (2)$$

그림3-10은 총 9개 프레임에서 차량 번호판으로 추정되는 지역화 영역이 발생한 이미지이고, 표3-3은 그림 3-10에서 발생한 데이터들의 중심점 x, y 값을 이용하여 평균변화율과 거리 값을 계산한 도표이다. 그림과 도표를 보면 알 수 있듯이 Frame 7에서 아웃라이어데이터가 발생했다. 해당 자료에서 일반적인 평균변화율은 10~20사이의 양수값이지만 아웃라이어데이터가 발생한 프레임7과 계산된 평균변화율은 음수의 값을 갖는다. 또한 해당 자료에서 일반적인 거리 값은 50~90사이의 값이지만 프레임 7에서 계산된 값은 한자리수의 값을 가졌다. 구체적인 데이터의 변화폭은 그림 3-11과 같다. 위치 값 (a)만으로는 변화폭이 크지 않은 데이

터의 특이치 발생을 관측하기 어렵지만 평균 변화율 (b)과 거리 값 (c)을 계산한다면 특이치 변화폭을 관측하기 용이하다. 이러한 데이터 분석을 이용하여 평균변화율과 거리 값의 연속 데이터 사이에서 발생 가능한 수치의 임계값을 설정하는 방식으로 아웃라이어데이터에 대한 식별이 가능하다.

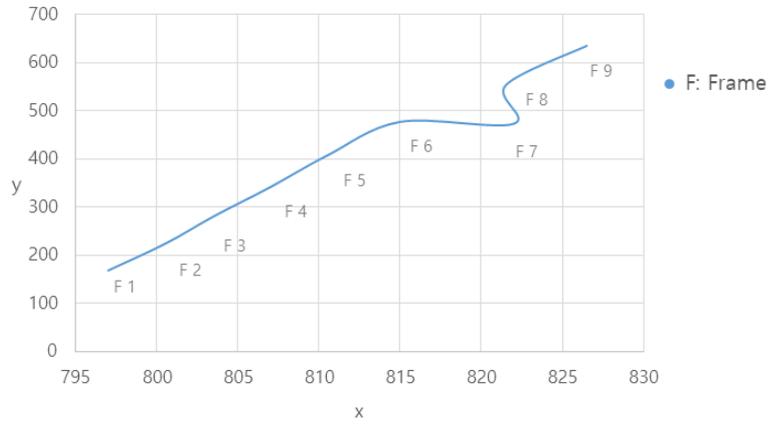


[그림 3-10] 각 프레임 차량 번호판 제안 영역.

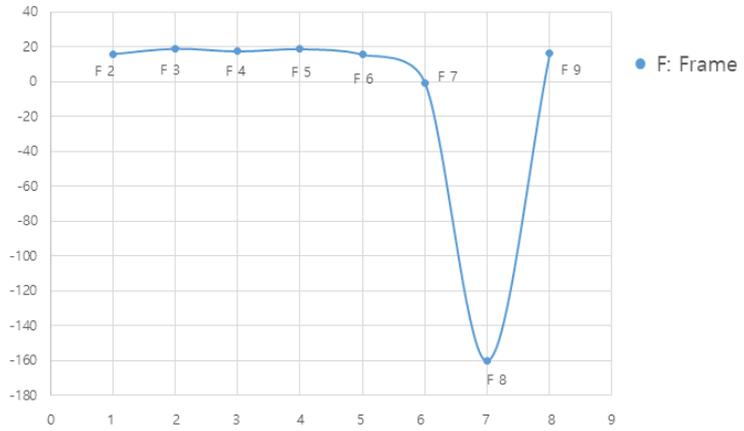
[표 3-3] 위치 값 평균변화율과 거리 값 계산

프레임	x	y	이전프레임 대비 평균 변화율	이전프레임 대비 거리 값
Frame 1	797	168	-	-
Frame 2	800.5	223.5	15.8571	55.6102
Frame 3	803.5	280.5	19	57.0788
Frame 4	807	341.5	17.4285	61.1003
Frame 5	810.5	407	18.7142	65.5934
Frame 6	815	477	15.5555	70.1444
Frame 7	822	473	-0.5714	8.0622
Frame 8	821.5	553	-160	80.0015
Frame 9	826.5	635	16.4	82.1523

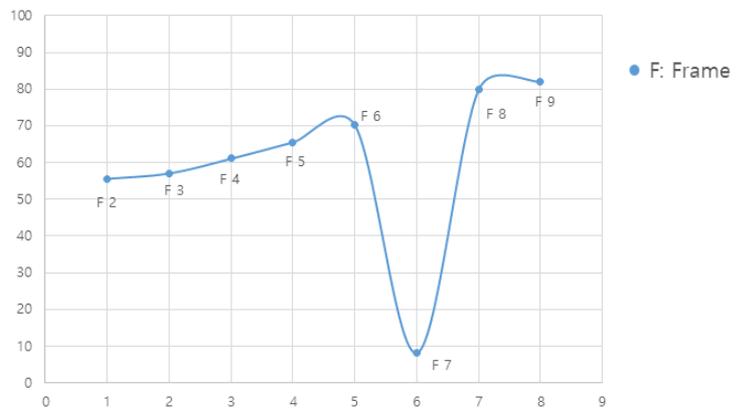
(a) 위치 값



(b) 평균 변화율



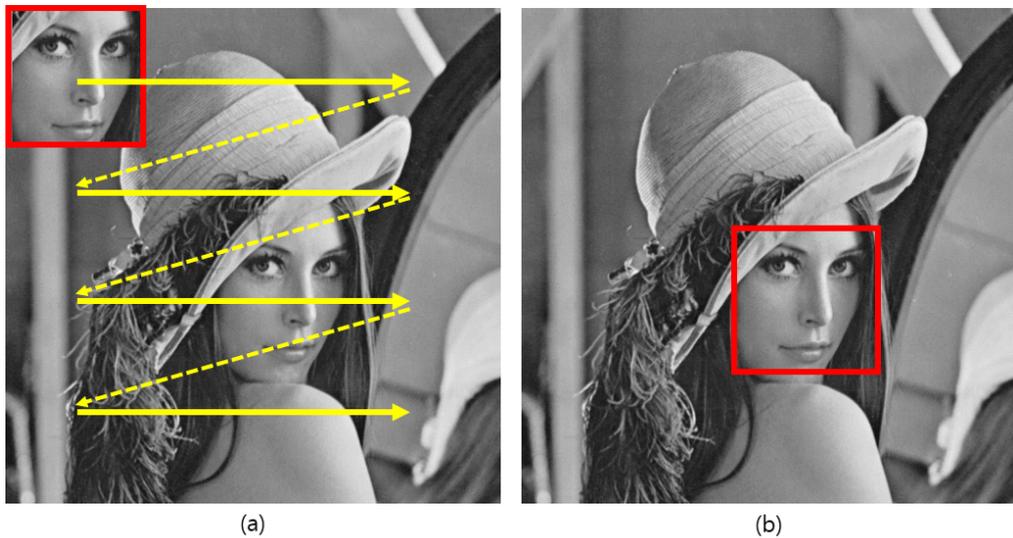
(c) 거리 값



[그림 3-11] 위치트래킹 그래프분석.

2) 템플릿매칭분석

템플릿매칭은 참조 이미지와 템플릿 이미지 사이의 픽셀 유사도를 비교하여 매칭이미지를 찾아내는 전통적인 이미지 비교방법이다. 그림 3-12 (a)와 같이 바운딩박스 영역인 템플릿 이미지를 참조이미지 내에서 픽셀을 옮겨 다니며 그림3-12 (b)와 같이 가장 높은 수치의 유사도 픽셀을 출력하는 원리이다(황선규, 2019).



[그림 3-12] 템플릿매칭 원리.

본 논문에서 템플릿매칭은 위치트래킹과 마찬가지로 딥러닝 기반 인식 이전에 실시한다. 만약 위치트래킹분석에서 전송지연 등에 의해 프레임간 데이터 손실이나 불연속이 발생하여 위치 추정이 불가능한 경우 템플릿매칭을 이용하여 하나의 그룹에 해당하는 데이터인지를 추정할 수 있다. 또한 본 논문의 템플릿 매칭은 어느 한 프레임을 기준으로 이전 프레임의 번호판 영역 데이터와 이후 프레임의 번호판 영역 데이터의 픽셀을 일대일 비교하는 것에 목적이 있다. 그러므로 픽셀이동비교 기능은 사용

하지 않고 동일한 크기의 두 이미지 픽셀 유사도 값을 출력해내는 기능만을 사용한다. 해당 비교 이미지들의 크기를 똑같은 크기로 재조정 해주기 위해 픽셀손실이 적은 보간법을 이용하여 크기를 재조정 해주는 과정이 필요하다. 템플릿이미지를 확대할 경우는 INTER_CUBIC 메소드를 이용하고, 이미지를 축소할 경우는 INTER_AREA를 사용한다. 매칭 메소드는 TM_SQDIFF_NORMED을 이용한다. 템플릿 T를 탐색영역 I에서 이동시켜가며 차이 제곱의 합계를 계산하여 매칭 값을 찾고 그 값을 정규화 계수로 나누는 원리로 식(3)과 같이 구성되어있다. 해당 정규화 방법은 조도의 차이가 존재하는 이미지의 영향을 크게 줄여주는 방식으로 차량 번호판 이미지 비교에 적합하다.

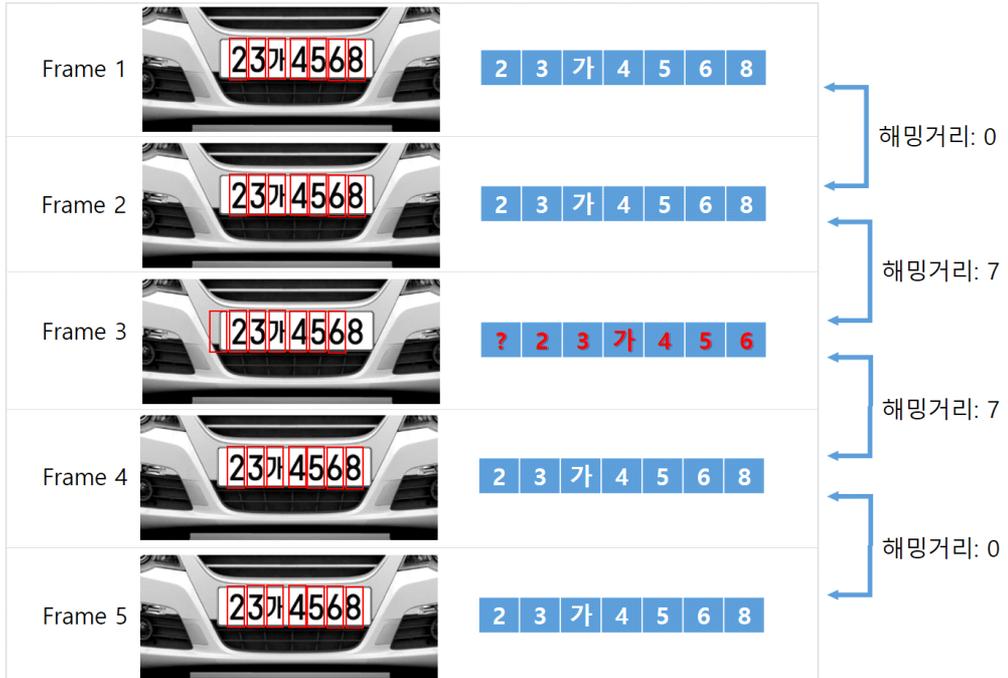
$$R(x, y) = \frac{\sum_{x',y'} (T(x',y') - I(x+x',y+y'))^2}{\sqrt{\sum_{x',y'} T(x',y')^2 \cdot \sum_{x',y'} I(x+x',y+y')^2}} \quad (3)$$

이미지 유사도는 0~1사이의 백분율형식 수치로 출력되며 그림 3-13과 같이 같은 그룹의 번호판끼리는 대부분 0.8 이상의 유사도를 보이고 서로 다른 그룹의 번호판과는 0.5 미만의 유사도를 보이는 경향이 있다.

Frame 1		↪ 유사도: 0.8993
Frame 2		
Frame 3		↪ 유사도: 0.4339
Frame 4		

[그림 3-13] 이미지유사도 산출 결과 예시.

위치트래킹과 템플릿매칭을 거쳤다고 해서 번호판 텍스트들이 전부 올바르게 CNN 입력 값으로 들어가지는 않는다. 번호판의 영역 제안은 문제가 없더라도 번호판 내 개별 텍스트영역 분할 과정이 잘못되어 아웃라이어데이터를 입력 값으로 받는 경우가 생기기 때문이다. 따라서 딥러닝 인식 이후에 사용되는 후처리 개념의 해밍거리분석은 번호판 내의 개별 텍스트들의 영역 분할이 올바르게 되었는지를 검증하고 잘못되었을 경우 이를 조정하여 원래의 입력 값이 되어야 하는 데이터를 식별 해낼 수 있다. 그림 3-15는 각 프레임 인근을 기준으로 차량 번호판 텍스트 영역에 대한 허밍거리를 계산한 예시이다. Farne 3은 텍스트 영역분할이 잘못 제안된 경우이며 인근 프레임과 해밍거리를 비교했을 시 해밍거리가 7로 계산되어 모든 자릿수가 인근 프레임 결과와는 다르다는 사실을 알 수 있다. 오류추정뿐만 아니라 그림3-15와 같은 상황으로 제안 영역이 잘못된 경우 바운딩박스를 좌측 시프트나 우측 시프트를 시도함으로써 올바른 결과를 얻어 낼 수도 있다. 차량 속도가 빠른 경우나 프레임이 손실된 경우 등 인식 데이터가 적은 상황일 경우 분석이 제한될 수 있지만 이러한 해밍거리분석을 통하여 최대한 많은 프레임들의 인식 데이터를 확보할 수 있다. 때문에 해밍거리분석의 오류추정 및 수정은 양적인 데이터를 확보하는 것에 도움이 된다.



[그림 3-15] 해밍거리 산출 결과 예시.

4) 그룹스코어분석

그룹스코어분석은 딥러닝 데이터 예측과정에서 마지막 뉴런의 출력 값을 이용하여 데이터를 분석하는 것을 의미한다. 번호판의 전체 텍스트를 더 신뢰성 있도록 구성하기 위한 과정이며 시계열변화에 따른 개별 텍스트들의 스코어를 분석한다. 먼저 각 프레임별 인식된 번호판 텍스트의 같은 자리 영역별로 예측된 결과그룹들을 나누고 그룹스코어(G-score)를 계산한다. 그 후 각 자리 영역마다 가장 높은 그룹스코어로 계산된 예측결과를 종합하여 전체 텍스트를 도출해내는 방식이다. 그룹은 같은 클래스로 분류된 예측 값들을 하나의 그룹으로 간주하고 해당 그룹 내 스코어들의 평균점수와 가중치를 합하는 식 (4)으로 그룹스코어를 계산한다. 즉 예측 빈도수가 많으면서 높은 스코어 평균으로 인식된 그룹일수록 가중치가 크며, 반대로 예측 빈도수가 적으면서 낮은 스코어 평균으로 인식된 결과 그룹일수록 가중치가 작다.

$$G\text{-score}(S_i) = \frac{\sum_{k=0}^{n_i} S_{ik}}{n_i} + \sum_{k=1}^{n_i} S_{ik} \cdot w \quad (4)$$

S : 각 예측그룹별 스코어집합
 w : 가중치 ($0 < w < 1$)

그림 3-16은 번호판 텍스트 첫 번째 자리(a)에 대한 그룹스코어분석을 실시한 예시이다. (b)는 프레임별로 인식된 값들이고 예측 값을 기준으로 (c)와 같이 4종류의 그룹으로 나눈다. 나뉜 그룹마다 그룹스코어를 계산하고 이를 비교하여 가장 높은 그룹스코어의 예측결과인 (d)를 도출한다. 이와 같이 각 프레임마다 인식된 값들의 스코어를 분석한다면 예측이 잘못된 오답 데이터들과 아웃라이어데이터들을 배제시키고 올바른 예측 결과를 도출에 기여하여 전체 번호판 텍스트를 출력할 수 있다.



	Region Proposal	prediction	score		Region Proposal	prediction	score	G-score	
Frame 1	2	2	1635	➔	A group	2	2	1635	
Frame 2	2	2	1810			2	2	1810	3890
Frame 3	4	4	2063			2	2	1758	
Frame 4	4	4	503			2	2	1870	
Frame 5	2	2	1758		B group	4	4	2063	
Frame 6	7	7	262			4	4	503	
Frame 7	2	2	1870		C group	7	7	262	262
Frame 8	1	1	2320		D group	1	1	2320	2320

(b) (c)



(d) Result : 2

[그림 3-16] 그룹스코어분석 실시 과정 예시.

제 4 장 실험

제 1 절 실험환경

본 논문은 표 4-1과 같은 기준으로 평가실험을 진행한다. 모델은 squeezeNet v1.1 모델을 기반으로 본 시스템에 맞게 변형시킨 모델을 사용한다. 이미지 채널은 흑백으로 입출력되며, 손실함수는 다중클래스분류에 적합한 categorical crossentropy를 사용하며 그리고 최적화함수는 adam을 사용한다. 코드는 python3.6으로 작성되었으며 딥러닝 프레임워크는 keras2.2.4기반으로 동작한다.

[표 4-1] 딥러닝 구성 기준

항목	구성 내역
Model	squeezeNet v1.1
Image channel	Gray scale
Loss function	categorical crossentropy
Optimizer	adam
Language	python3.6
Deep learning framework	keras2.2.4

딥러닝 모델은 3장 2절에서 언급했듯이 기호별로 예측모델을 따로 사용하기 때문에 숫자, 문자 수직 지역명 그리고 수평 지역명으로 구성된 총 4가지 종류의 모델이 있고 표 4-2와 같은 데이터셋 구성으로 학습시킨다. 비교 실험을 위하여 4가지 종류의 모델들을 각각 지도학습기법과 준지도학습기법으로 학습시킨다. 지도학습기법의 경우 일반적인 학습데이터로만 학습을 진행하고 준 지도학습기법의 경우 일반적인 학습데이터와 아웃라이어데이터를 함께 학습시킨다.

[표 4-2] 데이터 셋 구성

모델	학습 데이터	아웃라이어 데이터	검증 데이터 A타입	검증 데이터 B타입 (아웃라이어 데이터 포함)
수직 지역명모델	9421	576	972	1029
수평 지역명모델	9376	537	931	984
숫자모델	24278	925	2366	2458
문자모델	41597	1227	4098	4220

평가는 두 가지 종류의 평가를 진행한다. 첫 번째는 모델 검증평가로써 지도학습기법으로 학습된 모델들과 준 지도학습기법으로 학습된 모델들의 정확도를 두 종류의 검증데이터 셋을 이용하며 측정한다. 검증데이터 A타입은 아웃라이어데이터가 포함되지 않은 검증데이터이고, 검증데이터 B타입은 아웃라이어데이터가 포함된 검증데이터이다. 학습으로 사용된 전체 아웃라이어데이터의 약 10%의 비율이 검증데이터 B타입 내의 아웃라이어 데이터로 포함되어있다. 측정된 결과를 바탕으로 해당 모델들의 신뢰성 여부와 아웃라이어데이터를 분류성능을 검토한다. 두 번째는 실시간 인식 성능평가로써 학습된 모델들을 이용하여 구현시킨 본 논문에서 제시하는 번호판 인식 OCR시스템의 통합성능 테스트를 진행한다. 통합성능 테스트에서는 네 가지 시스템을 비교한다. 지도학습 기반의 시계열분석이 적용되지 않은 시스템과 적용된 시스템, 준 지도학습 기반의 시계열분석이 적용되지 않은 시스템과 적용된 시스템으로 각각의 성능을 평가한다. 평가 데이터는 실제 도로에서 주행하는 차량들을 대상으로 촬영된 실시간 영상을 이용한다. 주간 150대, 야간 181대의 차량 번호판 인식 여부를 기준으로 분류성능평가지표인 정밀도, 재현율, F1-Score, 정확도를 측정한다.

제 2 절 모델 검증평가

표 4-3은 4장 1절 실험 환경을 기준으로 모델 검증 평가를 진행한 결과이다. 아웃라이어데이터가 포함되지 않은 검증데이터 A타입의 평가결과 지도학습기법으로 학습된 모델은 숫자, 문자, 수직 지역명 그리고 수평 지역명모델 순으로 각각 99%, 97%, 97%, 96%의 정확도를 보였고, 준 지도 학습 기법으로 학습된 모델은 숫자, 문자, 수직 지역명 그리고 수평 지역명모델 순으로 각각 98%, 96%, 96%, 95%의 정확도를 보였다. 두 학습 기법은 아웃라이어데이터가 포함되지 않은 환경에서 성능적인 차이가 약 1%내외로 큰 수치를 보이지는 않았다.

[표 4-3] 모델 검증평가 진행 결과

학습 기법	모델	검증데이터 타입	정확도
지도학습	숫자모델	A 타입	0.9979
	문자모델	A 타입	0.9768
	수직 지역명모델	A 타입	0.9712
	수평 지역명모델	A 타입	0.9667
준 지도학습	숫자모델	A 타입	0.9831
	문자모델	A 타입	0.9675
	수직 지역명모델	A 타입	0.9619
	수평 지역명모델	A 타입	0.9549
지도학습	숫자모델	B 타입	0.9479
	문자모델	B 타입	0.9358
	수직 지역명모델	B 타입	0.9203
	수평 지역명모델	B 타입	0.9187
준 지도학습	숫자모델	B 타입	0.9805
	문자모델	B 타입	0.9630
	수직 지역명모델	B 타입	0.9553
	수평 지역명모델	B 타입	0.9502

아웃라이어데이터가 포함된 검증데이터 B타입 평가결과는 조금 달랐다. 지도학습기법으로 학습된 모델은 숫자, 문자, 수직 지역명 그리고 수

평 지역명모델 순으로 각각 94%, 93%, 92%, 91%의 정확도를 보였고, 준 지도학습 기법으로 학습된 모델은 숫자, 문자, 수직 지역명 그리고 수평 지역명모델 순으로 각각 98%, 96%, 95%, 95%의 정확도를 보였다. 지도 학습 기반 모델들은 대부분 크게 5%까지 성능이 하락된 반면에 준 지도 학습 기반 모델들은 대부분 원래의 성능을 유지했다.

학습으로 사용된 아웃라이어데이터 비율의 약 10%가 검증데이터 B타입에 포함된 것을 감안한다면, 아웃라이어데이터 비율이 높아질수록 더욱 지도학습 기반 모델들의 정확도가 하락할 것으로 예상된다. 지도학습 기반 모델들은 거의 아웃라이어데이터를 분류하지 못하였고 준 지도학습 기반 모델들은 대부분 아웃라이어데이터 분류가 가능하다는 사실을 본 검증평가 결과를 통하여 알 수 있었다.

제 3 절 실시간 인식 성능평가

1) 분류성능평가지표

실시간 인식 성능평가를 위하여 그림 4-1과 같은 오차행렬(confusion matrix)을 이용하여 분류성능평가지표인 정밀도, 재현율, F1-score, 정확도를 식(5)-(8)과 같이 측정하고 성능을 비교 분석한다. 본 논문의 측정을 위한 오차행렬 기준은 그림 4-2와 같다. 실제 번호판 텍스트를 올바르게 인식한 경우는 TP(True Positive)로 분류하고, 실제 번호판 텍스트를 올바르게 인식하지 못하거나 미인식한 경우는 FN(False Negative)으로 분류한다. 아웃라이어데이터를 번호판이라고 오인식한 경우는 FP(False Positive)로 분류하고, 아웃라이어데이터를 아웃라이어데이터라고 올바르게 인식한 경우는 TN(True Negative)으로 분류한다. 정리하자면 TP와 TN은 정답에 해당하며 양성데이터와 음성데이터를 올바르게 분류한 경우이고, FN과 FP는 오답에 해당하며 양성데이터와 음성데이터를 올바르게 분류해내지 못하는 경우이다.

		실제 정답	
		True	False
분류 결과	True	TP (True Positive)	FP (False Positive)
	False	FN (False Negative)	TN (True Negative)

[그림 4-1] 오차행렬(confusion matrix).

$$\text{정밀도(Precision)} = \frac{TP}{TP + FP} \quad (5)$$

$$\text{재현률(Recall)} = \frac{TP}{TP + FN} \quad (6)$$

$$F1\ score = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (7)$$

$$\text{정확도(Accuracy)} = \frac{TP + TN}{TP + FN + FP + TN} \quad (8)$$



[그림 4-2] 본 시스템 기준 오차행렬(confusion matrix).

2) 실시간 인식 결과 분석

실시간 인식 성능을 분석한 결과는 표 4-4와 같다. S(Supervised learning)기반 시스템은 아웃라이어데이터 분류에 취약하므로 네 가지 시스템 중 가장 낮은 정확도인 77%로 나타났다. S&TA(Supervised learning & Time series Analysis)기반 시스템에서는 아웃라이어데이터를 어느 정도 필터 할 수 있기 때문에 82%의 정확도를 보였다. SS(Semi-Supervised learning)기반 시스템은 딥러닝 인식과정에서 아웃라이어데이터를 분류가 가능하므로 85%의 비교적 높은 정확도를 보였다. SS&TA(Semi-Supervised learning & Time series Analysis)기반 시스템은 네 가지 시스템 중 가장 높은 정확도인 88%를 보였다.

[표 4-4] 번호판인식 OCR 시스템 성능평가.

S(Supervised learning): 지도학습;

TA(Time series Analysis): 시계열분석;

SS(Semi-Supervised learning):준 지도학습

시스템	정밀도	재현율	F1 score	정확도
S기반	0.9014	0.8504	0.8752	0.7794
S&TA기반	0.9278	0.8766	0.9015	0.8217
SS기반	0.9666	0.8671	0.9141	0.8519
SS&TA기반	0.9726	0.9018	0.9356	0.8821

아웃라이어데이터 분류성능을 확인하려면 TP와 FP 비율로 계산되는 정밀도를 확인한다. 예상대로 아웃라이어데이터 분류 기능이 취약한 S기반 시스템이 가장 낮은 정밀도인 90% 보였다. STA기반 시스템은 92%, SS기반 시스템은 96%의 정밀도를 보였다. 준 지도학습기반과 시계열분석이 함께 적용된 SSTA기반 시스템은 정밀도가 97%이상으로 나타났으며, 두 기법을 함께 적용된다면 서로의 부족했던 기능을 보완하여 거의 대부분의 아웃라이어데이터 구분과 인식을 향상시킬 수 있다는 것을 알 수 있었다.

정밀도와 비교했을 때 재현율이 낮은 이유는 오답 중 FP로 분류된 오답보다 FN으로 분류된 오답이 훨씬 많기 때문이다. 따라서 F1 score와 정확도를 더욱 향상시키기 위해서는 FN으로 분류된 오답을 줄이는 방안의 연구가 필요하다.

제 5 장 결론

본 논문은 차량 번호판의 실시간인식을 위한 OCR 시스템을 구현하였다. 기존 차량 번호판 인식 OCR시스템은 차량이외 객체가 지나가는 등의 촬영 상황이 제한되지 않은 실시간 상황에서 낮은 인식률을 보였다. 이러한 낮은 인식률은 아웃라이어데이터로 인하여 발생한 문제이므로 본 논문에서는 아웃라이어데이터를 억제하고 인식률을 높이기 위한 딥러닝 모델과 데이터 분석기법을 설계하였다.

딥러닝 모델은 차량 번호판 내 각각의 기호마다 인식하는 모델을 별도로 구성함으로써 타 클래스 기호를 출력 값으로 내놓는 상황을 배제하여 정확도를 높였다. 모델은 경량 모델인 squeezeNet v1.1을 기반으로 본 논문 상황에 맞게 입출력 레이어를 변경하였다. 데이터 학습은 일반데이터와 아웃라이어데이터를 함께 학습시키는 준 지도학습기반으로 학습되며 해당 학습 기법을 통하여 아웃라이어데이터 발생 유무를 구분한다. 딥러닝 인식데이터 클래스 예측기능 사용 시 학습모델과 별도로 구성된 예측 모델을 사용하는 것으로 출력 값을 스코어 형태로 받고 이를 이용하여 데이터 분석을 실시한다.

인식률을 높이고 아웃라이어탐지를 보조하기 위하여 각 프레임의 시간에 따른 데이터변화를 비교 분석하는 형태의 시계열분석을 실시한다. 총 네 가지 시계열분석기법인 위치트래킹, 템플릿매칭, 해밍거리, 그룹스코어 분석을 제안하였다. 해당 분석들을 통하여 더 신뢰성 높은 데이터를 사용하고 아웃라이어데이터를 배제시키는 역할을 보조함으로써 시스템의 정확도를 높이는데 기여한다.

본 논문에서 제시한 내용들을 기반으로 모델검증 평가와 실시간 인식 성능평가 실험을 진행하였다. 먼저 모델 검증평가에서는 지도학습과 준 지도학습 모델의 성능을 비교하였다. 실험 결과 아웃라이어데이터가 없는 검증 셋에서는 지도학습과 준 지도학습 모델들 모두 95%이상의 정확도를 보였다. 그러나 아웃라이어데이터가 포함된 검증셋에서 지도학습 모델들

은 모두 95%미만으로 정확도가 떨어졌지만 준 지도학습 모델들은 여전히 95%이상의 인식률을 유지했다. 검증셋의 아웃라이어데이터는 학습으로 사용된 아웃라이어데이터의 10%비율로 구성되어있는 것을 감안했을 때, 검증셋 내의 아웃라이어데이터 비율이 늘어날수록 지도학습모델의 인식률은 더 떨어질 것으로 예상된다. 또한 모델 검증평가 실험을 통하여 준 지도학습이 충분히 아웃라이어데이터 판별에 용이하다는 사실을 알 수 있었다.

두 번째 실험인 실시간 인식 성능 평가에서는 구현한 기법들을 적용할 경우 성능이 어떻게 변화하는지를 확인 할 수 있었다. 일반적인 방식과 동일하게 설계된 S(Supervised learning)기반의 차량번호판 OCR시스템에서 약 77%의 인식 정확도를 보였다. 그러나 본 논문에서 제시하는 기법들로 구현된 S&TA(Supervised learning & Time series Analysis) 기반 시스템, SS(Semi-Supervised learning) 기반 시스템, SS&TA(Semi-Supervised learning & Time series Analysis)기반 시스템은 각각 82%, 85%, 88%의 인식 정확도를 보였다. SS&TA기반 시스템이 가장 높은 인식 정확도를 보였기 때문에 두 가지 기법을 함께 사용하는 것이 가장 높은 정확도를 얻을 수 있다는 것을 알 수 있었다. 또한 해당 시스템의 아웃라이어데이터 판별성능을 증명 할 수 있는 수치인 정밀도가 97%인 것을 감안하면 대부분의 아웃라이어데이터 판별이 가능하다는 것을 알 수 있었다.

본 논문의 한계로는 가장 높은 인식률은 보인 SS&TA기반 시스템의 재현률은 정밀도에 비하여 비교적 낮은 수치인 90%를 보였다는 점이다. 재현률이 낮은 이유는 FN 수치가 높기 때문인데, 이는 텍스트를 오인식하거나 또는 아예 지역화 영역을 제안하지 못하는 미인식의 경우이다. 본 논문은 오답의 경우중 주로 아웃라이어데이터를 구분하는 FP의 판별 성능 개선을 위주로 내용을 전개하였고 FN의 판별 성능개선문제는 다루지 않았다. 향후 연구로 FN의 판별성능을 개선하기 위하여 텍스트의 인식률을 개선하는 전처리과정 또는 지역화 영역제안 성능을 높이는 기법 등을 추가적으로 연구한다면 더 높은 정확도의 차량 번호판 OCR시스템 구현이 가능할 것으로 판단된다.

참 고 문 헌

1. 국내문헌

- 기술이전목적 및 필요성. (2018). 『[ETRI] 딥러닝기반 다중 자동차번호판 인식 기술(다차로 LPR2.0, 전기차/신규번호판 수용)』. 0-0
- 박성실, 박세호. (2017). 카메라를 사용한 차량 번호판 인식을 위한 국내 번호판 종류 구별 방안. 『한국통신학회 학술대회논문집』, 317-318.
- 송영준, 안재형, 김남. (2006). 자동차 번호판 인식 기술 동향. 『IITA] 정보통신연구진흥원 학술정보』, 0-0.
- 윤종호. (2017). 『지능형 교통 서비스를 위한 다중 번호판 인식 영역 추출 알고리즘 연구』. 한양대학교.
- 차종 및 용도구분등의 기호; 관할관청 기호표시. (2018). 『자동차 등록번호판 등의 기준에 관한 고시』. (5). 국토교통부.
- 황선규. (2019). 템플릿매칭. 수록처: 『OpenCV 4로 배우는 컴퓨터 비전과 머신 러닝』 (13.1). 서울: (주)도서출판길벗.
- Lee, S., Son, K., Kim, H., & Park, J. (2017). *Car plate recognition based on CNN using embedded system with GPU*. In 2017 10th International Conference on Human System Interactions (HSI), 239-241.

2. 국외문헌

- Acharya, S., Pant, A. K., & Gyawali, P. K. (2015). *Deep learning based large scale handwritten Devanagari character recognition*, In 2015 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), IEEE, 1–6.
- Ahmed, M., Mahmood, A. N., & Hu, J. (2016). A survey of network anomaly detection techniques, *Journal of Network and Computer Applications*, 60, 19–31.
- Anderson, T. W. (2011). *The statistical analysis of time series*, Vol. 19. John Wiley & Sons.
- Atiwadkar, A., Mahajan, S., Lande, T., & Patil, K. (2015). Vehicle license plate detection: A survey, *International Research Journal of Engineering and Technology (IRJET)*, 2(8), 354–360.
- Balci, B., Saadati, D., & Shiferaw, D. (2017). *Handwritten text recognition using deep learning*, *CS231n: Convolutional Neural Networks for Visual Recognition*, Stanford University, Course Project Report, Spring.
- Chandola, Varun, Arindam Banerjee, and Vipin Kumar. (2009). Anomaly detection: A survey, *ACM computing surveys (CSUR)*, 41(15), 3.
- Domingues, R., Filippone, M., Michiardi, P., & Zouaoui, J. (2018). A comparative evaluation of outlier detection algorithms: Experiments and analyses, *Pattern Recognition*, 74, 406–421.
- Görnitz, N., Kloft, M., Rieck, K., & Brefeld, U. (2013). Toward supervised anomaly detection, *Journal of Artificial Intelligence Research*, 46, 235–262.
- He, M. X., Petoukhov, S. V., & Ricci, P. E. (2004). Genetic code, Hamming distance and stochastic matrices, *Bulletin of mathematical biology*, 66(5), 1405–1421.
- Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., & Keutzer, K. (2016). *SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size*, arXiv preprint arXiv:1602.07360.

- Jang, E., Gu, S., & Poole, B. (2016). Categorical reparameterization with gumbel-softmax, arXiv preprint arXiv:1611.01144.
- Lopresti, D. (2009). Optical character recognition errors and their effects on natural language processing, *International Journal on Document Analysis and Recognition (IJ DAR)*, 12(3), 141–151.
- Mithe, R., Indalkar, S., & Divekar, N. (2013). Optical character recognition. *International journal of recent technology and engineering (IJ RTE)*, 2(1), 72–75.
- Neumann, L., & Matas, J. (2010). *A method for text localization and recognition in real-world images*, In Asian Conference on Computer Vision, Springer, Berlin, Heidelberg, 770–783.
- Sonavane, K., Soni, B., & Majhi, U. (2015). Survey on Automatic Number Plate Recognition (ANR), *International Journal of Computer Applications*, 125(6).
- Yan Y. Madden S. Rundensteiner E. Cao, L. (2019). *Outlier Detection from ImageData*, The International Conference on Learning Representations, 18.
- Yilmaz, A., Javed, O., & Shah, M. (2006). Object tracking: A survey. *Acm computing surveys (CSUR)*, 38(4), 13.
- Zhong, Z., Jin, L., & Xie, Z. (2015). *High performance offline handwritten chinese character recognition using googlenet and directional feature maps*, In 2015 13th International Conference on Document Analysis and Recognition (ICDAR), IEEE, 846–850.

ABSTRACT

Deep Learning Based OCR Design for Real Time Recognition of License Plate

Kim, Ba-Da

Major in Computer Engineering

Dept. of Computer Engineering

The Graduate School

Hansung University

(Recently, the real-time recognition system of car license plates has been commercialized in a variety of fields, and it prefers low-cost embedded systems using only cameras. These systems have a higher recognition rate of over 98% for the environments such as parking lots where non-vehicle objects do not appear. However, there is a 50% to 70% recognition rate on some roads or alleys where the environments are not restricted. The decrease in performance occurs when the external environmental factor, such as the occurrence of non-vehicle objects or the changes in illumination, become the outlier data. This paper designs the license plate OCR system based on deep learning and the system uses the embedded board which enables the input and output of the images of camera. To improve OCR recognition rate in situations where the environments are not restricted, semi-supervised learning and time-series

analysis techniques can be applied. Normal data and outlier data are learned together in one model by semi-supervised learning, and it identifies the outlier and improves the recognition rate. Time series analysis identifies more reliable frame data by comparing real-time frame data. In order to perform the experiment for the performance comparison, measure the recognition accuracy of general licence plate OCR system and this paper's proposal technique of licence plate OCR system in a situation where the environment is not restricted. As a result, the general license plate OCR system had a recognition rate of 77%, whereas semi-supervised learning and time-series analysis based on license plate OCR system had a recognition rate of 88%. Using the two techniques, semi-supervised learning and time-series analysis, we could see that the OCR system with an improved recognition rate can be implemented.

KEYWORD: Licence plate recognition, OCR, Deep learning, outlier, semi-supervised learning, time series analysis