

실무중심산학협력프로젝트 최종 보고서

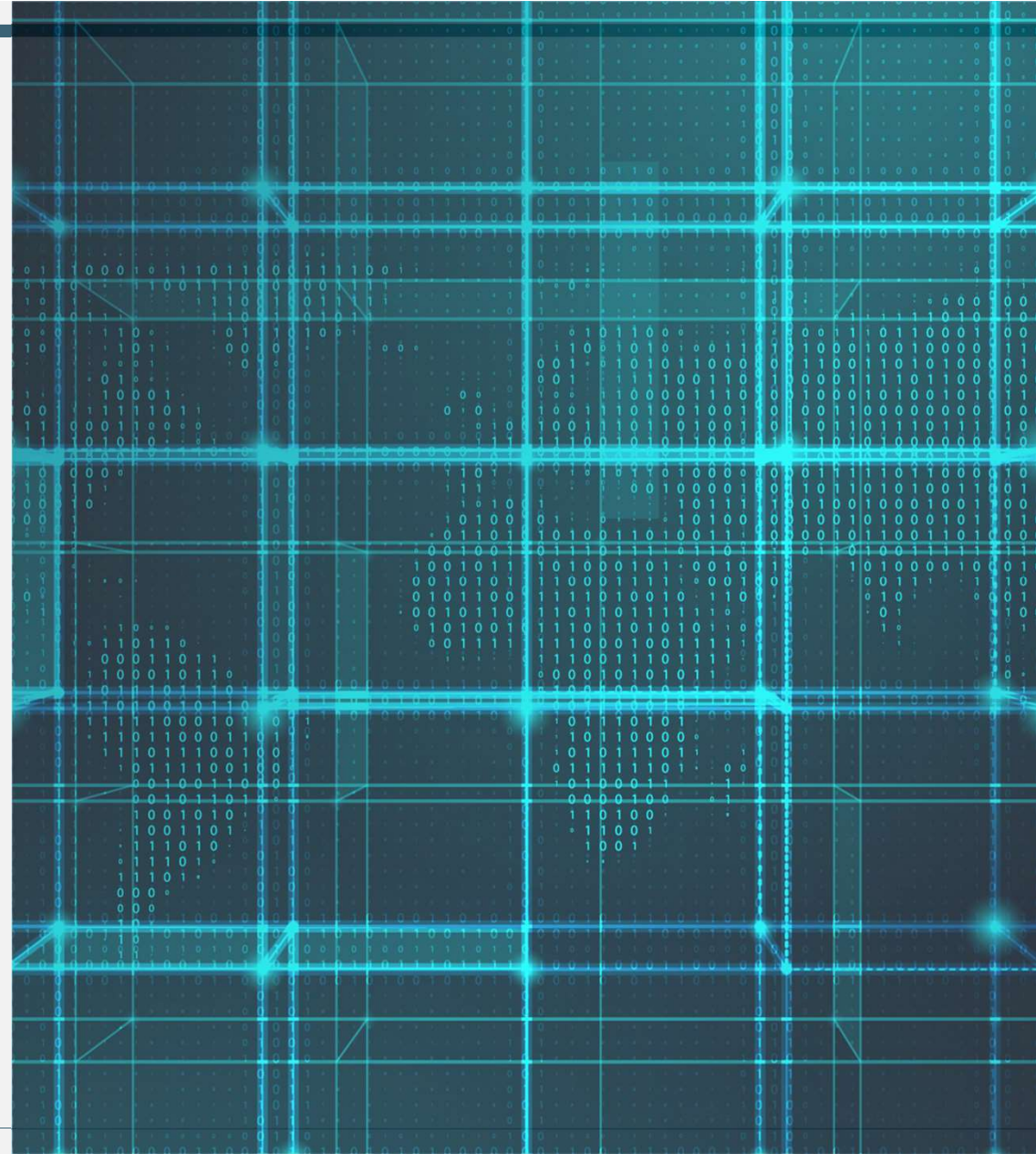
딥러닝을 이용한 CAR PLATE 인식 모델 구현

과목명	캡스톤 5분반	담당교수	황두성
학 번	32180360/ 32195083/ 32193392		
이 름	김경태/ 구희찬/ 이은송		

목차

table of contents

- 1 과제 개요
- 2 Web 구축
- 3 Localization
- 4 Segmenataion
- 5 Char recognition
- 6 구축 결과



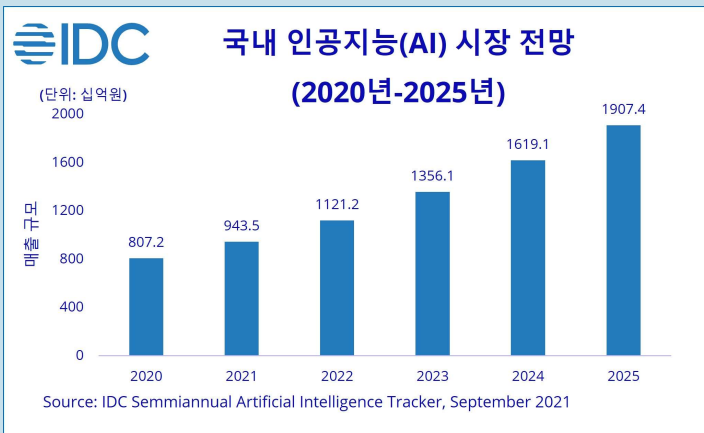
1

과제 개요

과제 개발 이유



- Chat gpt처럼 인공지능 관련된 서비스 증가
- 인공 지능 시장 전망은 지속적으로 성장할 예정
- 반면, 학부 수업 중 ai를 활용한 수업 부재
- 따라서, 학과 수업, 전공 서적 등 과거 기술과 더불어 딥러닝 기술을 사용하여 기존 지식과 융합하는 주도적인 학습 필요



자동차 번호판 인식 선정 이유

기존 자동차
번호판 인식 방법



딥러닝 기반
자동차 번호판 인식

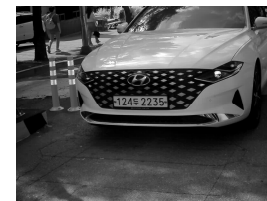
외곽선
추출

차량
번호용
문자 대조

고정된 형식의
번호판이므로
학습 용이



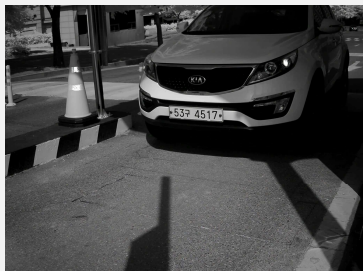
학습을 위한
데이터 확보 용이



데이터 2000개 확보

과제 목표

<프로그램에 사진을 넣기>



<자동차 번호판의 문자 인식하기>



1. 자동차 번호판 인식 프로그램을 딥러닝을 통해서 구현하기
2. 서버 구축하여 사용자가 쉽게 이용할 수 있는 서비스 제공하기

UI design 구상

이미지 업로드

차량 번호판 인식



ABC-123



여기로 이미지를 드래그하거나
파일을 업로드 하세요!

또는

이미지 링크 붙여넣기

검색

인식 결과

차량 번호판 인식



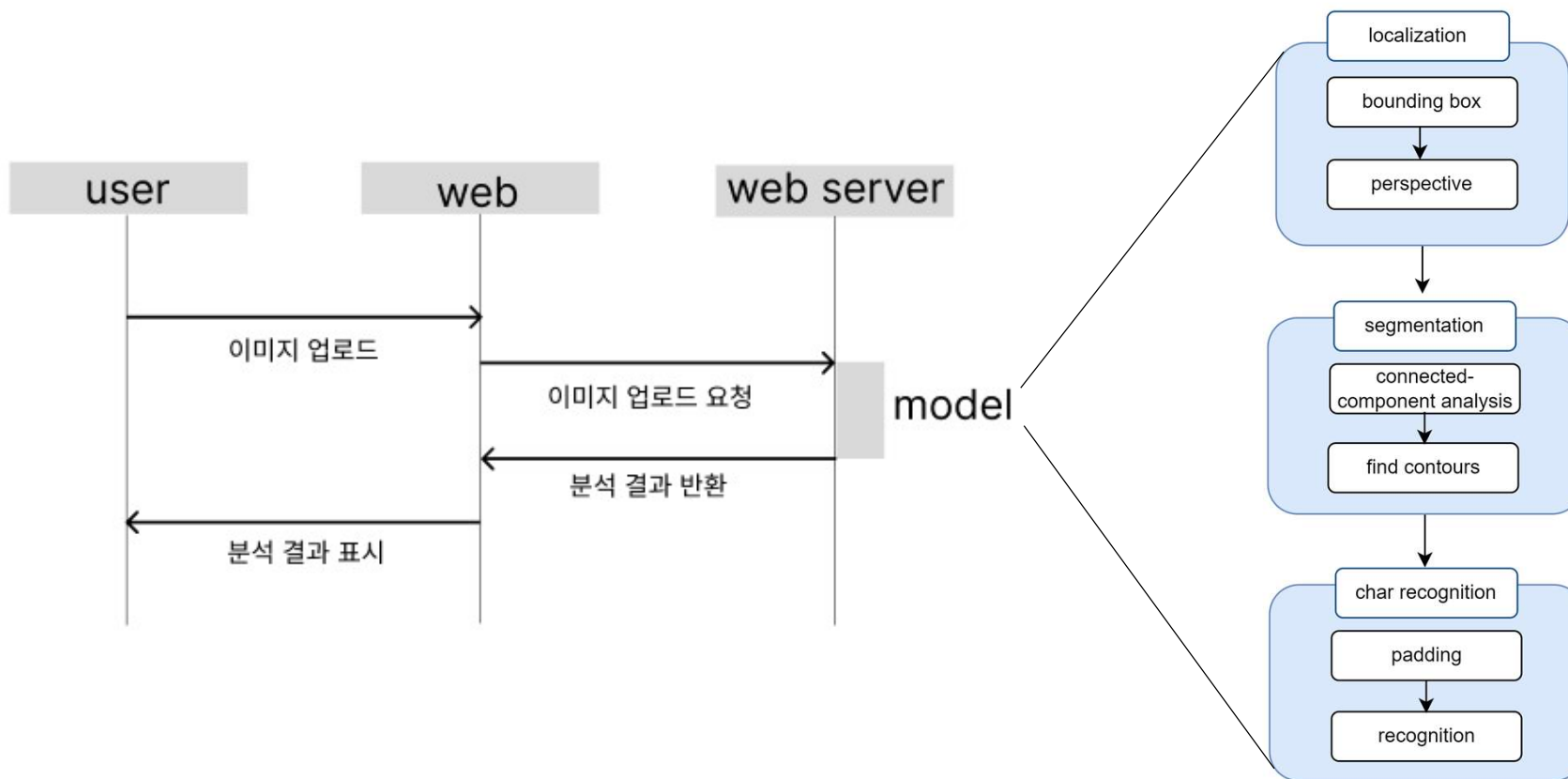
ABC-123

인식결과

49노 3847

49노 3847

시스템 구성도



2

Web

Web page 구축 필요성

사용자
접근성

웹 브라우저를 통해 별도의 설치과정 없이 모델을 활용할 수 있는 기회를 제공
별도의 소프트웨어 설치 과정이 필요하지 않기 때문에 모델 사용 장벽을 낮춤

시각화,
실시간 서비스
제공

유저가 입력 데이터를 요청시 모델은 처리 후에 즉시 결과를 반환
모델의 결과를 실시간, 시각적으로 확인 가능

flask 프레임워크 선정 이유

flask

1. 코드 단순, API 서버 만들기 편리
2. RESTful API 제공하기 때문에 HTTP 메소드를 활용한 데이터 처리, 교환 가능

백엔드 로직

1. 경량화되어 있어서 작은 규모의 웹페이지 구축에 적합
2. 다양한 확장 모듈과 플러그인이 있어 웹 애플리케이션 구축에 용이

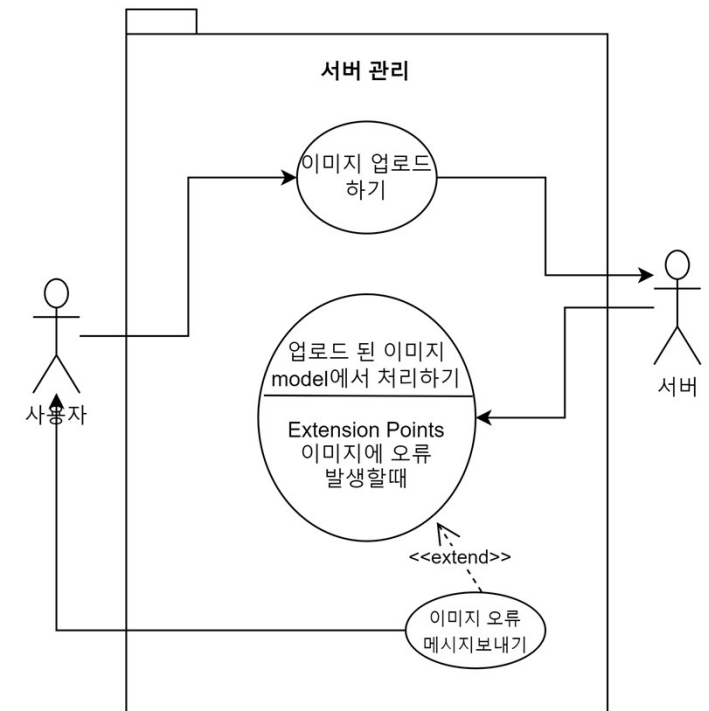
통합 개발 환경

1. Python 기반이기에 python 모델과 compatible하다
2. 백엔드와 프론트엔드를 모두 파이썬으로 개발할 수 있어 복잡성이 적다

사용자 요구사항 분석

Use – case 명세서

이름	사용자 웹 페이지 이용하기		
개요	사용자가 이미지를 전달하면 서버로 이동		
관련 액터	주 액터	사용자	
	보조 액터	서버	
우선 순위	개발의 우선순위	중요도	(상)
		난이도	(중)
선행 조건	사용자가 이미지를 전달할 수 있게 만들어야 한다		
후행 조건	사용자는 이미지 결과를 확인할 수 있어야 한다		
시나리오	기본 시나리오	<ol style="list-style-type: none"> 1. 사용자가 이미지를 첨부한다 2. 전달하기 버튼을 누른다 3. DO 4. 이미지가 서버로 업로드 5. 모델을 통해 이미지 처리된 후 결과 전달 6. WHILE (이미지 전달 시) 7. 메인 화면을 보여준다 	
	대안 시나리오	<p>대안 시나리오- A1. 등록 기본 시나리오 5에서 분기 (조건) 이미지에 오류 발생 시</p> <ol style="list-style-type: none"> 1. 이미지에 오류 발생 메시지를 전달한다 2. 어디서 오류가 발생했는지 메시지 전달한다 3. 기본 시나리오 7로 돌아간다 	



웹 페이지 구축계획서

UI 기획

<메인 화면>

- 번호판 이미지 업로드 기능 제공
- 드래그앤 드롭 등의 기능 제공

<결과 화면>

- 사용자 가독성을 높이기

와이어프레임 작성

<메인 화면>

- 화면 상단에 웹 페이지의 제목과 로고 이미지가 표시
- 중앙에는 번호판 이미지 업로드 버튼이 위치

<결과 화면>

- 번호판 인식 결과 즉시 결과 페이지에 표시
- 인식된 번호판의 문자에 bounding box와 텍스트가 빨간색 글씨로 표시

시각적 디자인 작업

<아이콘 및 버튼 디자인>

- 이미지 업로드 기능 옆에 카메라 아이콘을 넣어 이미지 삽입 버튼을 강조
- 이미지 업로드 기능에는 '파일 업로드를 위해 클릭해주세요!' 텍스트 삽입

<이미지 그래픽 요소>

- 자동차 모형 이미지를 블러 처리 후 삽입해서 웹페이지의 로고로 사용



프로토타입 제작

CSS

웹페이지의 스타일과 레이아웃을 정의한다.
색상, 폰트, 텍스트 스타일, 크기 및 위치 등의 시각화 요소를 설정한다.

```
.submit-button {
  display: block;
  margin: 0 auto;
  padding: 10px 20px;
  background-color: #4CAF50;
  color: white;
  border: none;
  text-align: center;
  text-decoration: none;
  font-size: 16px;
  margin-top: 50px;
  cursor: pointer;
  position: relative;
  z-index: 1000;
}

/* 차량 번호판 인식 */
.license-recognition {
  position: absolute;
  width: 250px;
  height: 57px;
  left: 50%;
  top: 20%;
  transform: translate(-50%, -20%);
  font-family: 'S-Core Dream';
  font-style: normal;
  font-weight: 700;
  font-size: 32px;
  line-height: 57px;
  display: flex;
  align-items: flex-end;
  color: #000000;
  text-shadow: 0px 4px 4px rgba(0, 0, 0, 0.25);
}
```

제출버튼과 제목의 css 코드

HTML

① 파일 업로드 기능:

‘파일 업로드를 위해 클릭해주세요!’라는 텍스트가 표시된 레이블을 선택할 경우 이미지 업로드 기능을 제공한다.

```
<div class="upload-section">
  <label for="file">파일 업로드를 위해 클릭해주세요!</label>
  <input type="file" name="file" id="file" />
</div>
```

파일 업로드 기능 html 코드

② 제출 기능:

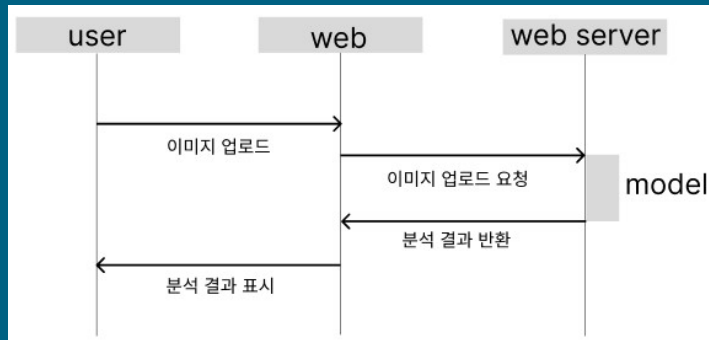
파일 업로드를 완료하고 서버에 이미지를 전송하기 위한 기능을 제공한다

```
<input type="submit" class="submit-button" value="제출" />
```

제출 기능 html 코드

웹 페이지 구축

(1) 웹 시스템 레이아웃



(2) 이미지 업로드 기능

flask에서 url 경로 'uploader'에 대한 GET과 POST 요청을 처리한다. 사용자가 이미지 파일을 업로드하는 상황이 POST 요청을 보낸 경우이다. POST 요청을 보내야 프로그램이 실행된다. 이미지 저장 경로를 지정하고, 이미지를 해당 경로에 저장한다

```

# 파일 업로드 처리
@app.route('/uploader', methods=['GET', 'POST'])
def uploader_file():
    if request.method == 'POST':
        f = request.files['file']
        # 이미지 저장경로 지정
        filename = secure_filename(f.filename)
        upload_path = os.path.join(app.config['UPLOAD_FOLDER'], filename)
        f.save(upload_path)
        image_url = url_for('static', filename=f'file_upload/{filename}') # 이미지 URL 생성
  
```

(3) 결과 이미지 반환 및 결과화면 html 렌더링

지정한 경로에 결과 이미지를 저장한다. 모델 수행 결과를 저장하고 결과 페이지 템플릿을 렌더링해서 유저에게 응답을 반환한다.

```

# 결과 이미지 파일 경로
result_filename = 'result.jpg'
result_path = os.path.join(app.config['UPLOAD_FOLDER'], result_filename)
# 결과 이미지를 저장합니다.
im_pil.save(result_path, "JPEG")
result_text = ''.join(final_char)
print(result_text)
print('결과 이미지 저장 완료')
if os.path.exists(result_path):
    os.chmod(result_path, 0o644)
    print("result.jpg 파일이 생성됨")
    # return redirect('/show_result')
    # return render_template('result.html')
    # 결과 이미지 파일 URL
    filename = 'static/result.jpg'
  
```

(4) 파일 업로드 및 제출 기능

파일 업로드를 유도하기 위한 텍스트 라벨을 구현했고 이 라벨을 클릭하면 'input' 요소가 선택된다. input의 type 속성을 file로 설정했기 때문에 파일 업로드가 가능한 파일 선택 창이 표시된다.

```

<label for="file">파일 업로드를 위해 클릭해주세요!</label>
<input type="file" name="file" id="file" />
</div>
<input type="submit" class="submit-button" value="제출" />
  
```

자세한 내용은 아래 링크 참조

Github: https://github.com/qwertyfired/car_plate/blob/main/web.py

Model 구성 요소

localization

segmentation

**Character
recognition**

3

localization

Localization이란?

물체나 특정한 지점의 위치를 찾는 기술

위치의 정확도는 **bounding box**를 사용

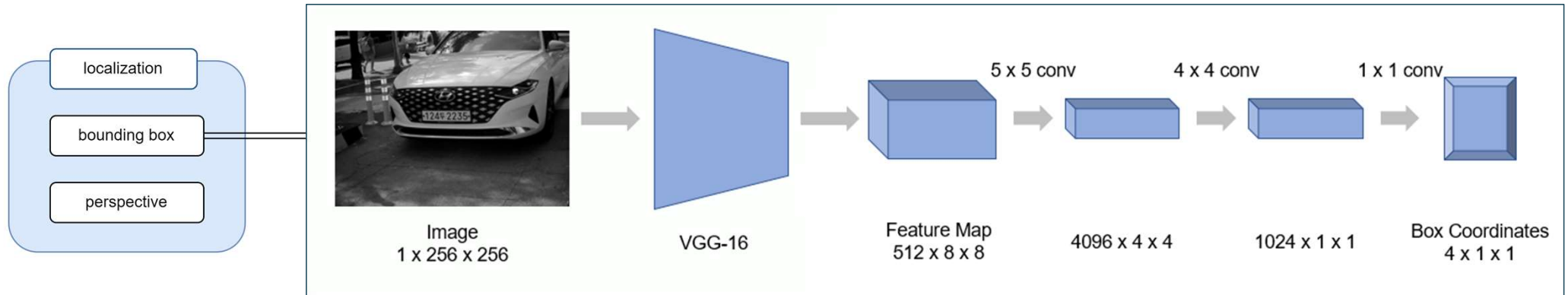
bounding box: 객체를 둘러싼 직사각형 영역(위치, 크기)

회귀 모델(Regression)을 이용
좌표 x, y, w, h 를 구한다

달성 목표



Bounding box



- 입력값: 가로 256, 세로 256 이미지
- 출력값: 좌표 x, y, w, h 4개 값

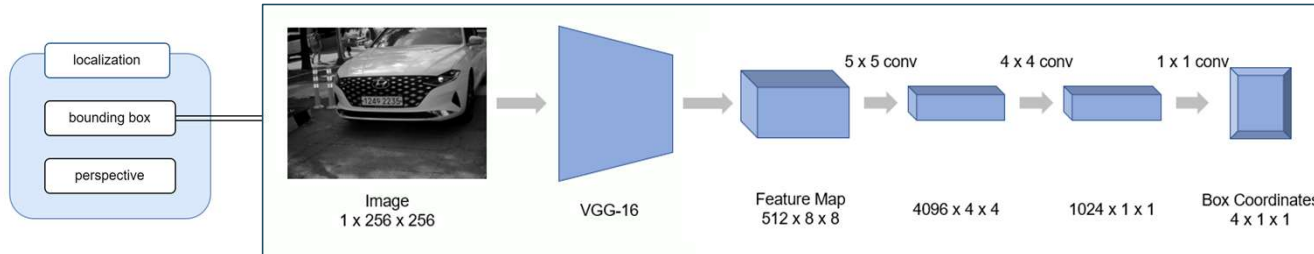
참조 사항

VGG-16: 16개 계층으로 구성된 convolution 신경망

자세한 내용: https://www.mathworks.com/help/deeplearning/ref/vgg16_ko_KR.html

Feature map: input 이미지가 convolution layer를 지났을 때의 결과값

Bounding box



학습 방법

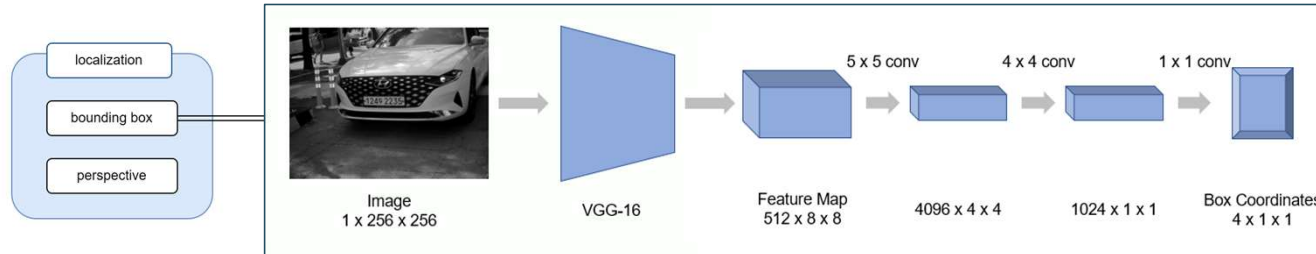
1. 이미지가 neural network를 지나 출력값을 도출 한다. 출력값(x, y, w, h)
2. 이미지의 Labeled 값(x, y, w, h)과 비교한다.
3. 이미지의 출력값과 labeled 값과의 차이값을 neural network에 학습한다.
4. 1,2,3과정을 100번 반복한다. (epoch: 100)

이미지 labeled 파일

```
"version": "5.0.1",
"flags": {},
"shapes": [
  {
    "label": "LP",
    "points": [
      [
        933.3846153846152, (x,y)
        569.9230769230769
      ],
      [
        1253.3846153846152, (w,h)
        669.9230769230769
      ]
    ],
    "group_id": null,
    "shape_type": "rectangle",
    "flags": {}
  }
]
```

github https://github.com/qwertyfired/car_plate/blob/main/localization.py

Bounding box



결과



input image



output image



Bounding box 부분 cropping(자르기)

문제점 발생



이미지에 왜곡
(perspective) 발생



Perspective란?

초점 거리, 카메라의 위치 등에 따라 물체의 모양, 크기, 거리 등이 이미지에 투영될 때의 왜곡

참조 문서

변환행렬: https://en.wikipedia.org/wiki/Transformation_matrix

wrapPerspective: https://docs.opencv.org/4.x/da/d54/group__imgproc__transform.html

github https://github.com/qwertyfired/car_plate/blob/main/perspective.py

문제 해결 방법

1. 그 이미지를 binary image(값이 255 혹은 0)로 변경
2. openCV의 connected-component 함수를 이용
(값이 연결된 것들을 graph로 인식하여 pixel값이 255로 구성된 그래프 배열을 얻는다)
3. 그래프 배열 중에 가장 큰 값을 plate라고 추정
4. 각 꼭지점을 찾아 OpenCV warpPerspective 함수를 이용하여 perspective 왜곡 문제 해결

4

Segmentation

Segmentation이란?

이미지를 픽셀 단위로 분할하여 각 픽셀을 서로 다른 객체 또는 배경으로 할당하는 작업

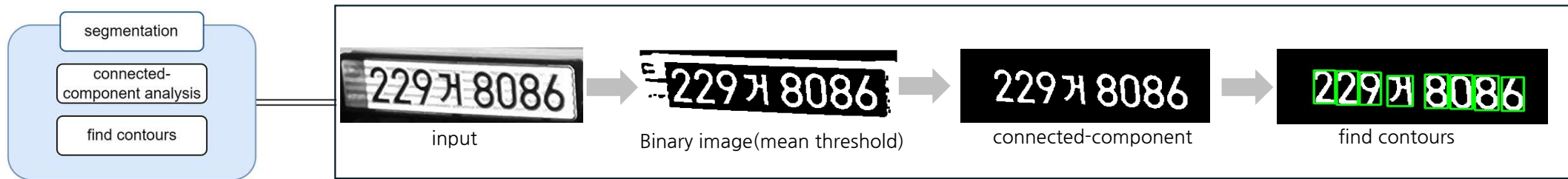
이미지 내에서 객체의 경계를 정확하게 식별하고 객체의 영역을 분리하여 추출

Plate 문제의 경우, 값이 연결된 것들을 graph로 인식하여 그래프를 하나의 객체로 인식

달성 목표



Segmentation



방법

1. Image의 pixel 평균값을 구한다
2. 평균값 보다 큰 pixel은 0, 작은 pixel은 255로 변경한다
3. OpenCV의 connected-component 함수를 이용해 graph 배열을 구한다[1].
4. 각 graph에 대해서 OpenCV의 contour 함수를 이용하여 segmentation을 한다

github https://github.com/qwertyfired/car_plate/blob/main/segmentation.py

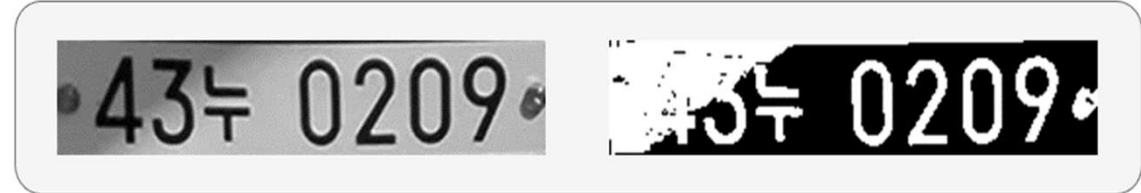
Segmentation



문제점 발생



1. 모음과 자음이 분리된 경우

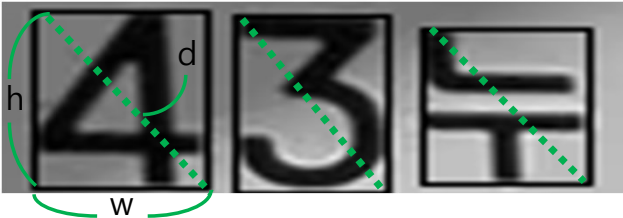


2. 한 쪽에 그림자가 있는 경우



문제 해결 방법

1. 모음과 자음이 분리된 경우



1. 각 객체의 width, height, diagonal 길이를 구해 상대적으로 짧은 것들을 모음, 자음으로 인식

모음과 자음 2개를 합쳐 하나의 객체로 인식

2. 모음 'ㅏ', 'ㅑ'의 경우 자음과 x좌표가 겹치기 때문에 겹치는 부분이 80% 이상이면 두 개의 객체를 합쳐서 하나의 객체로 인식

Segmentation



문제 해결 방법

2. 한 쪽에 그림자가 있는 경우



1. plate의 pixel 평균값을 구한다
2. 평균의 70% 이상에 해당하는 값들을 255로 변경하여 문자 부분의 검은 색 부분만 강조

5

Char recognition

Char recognition이란?

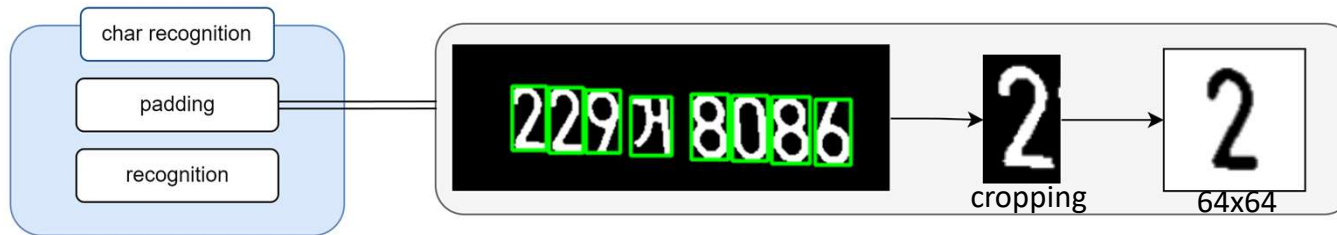
문자 이미지를 입력으로 받아 해당 문자를
예측하고 분류하는 작업

학습에는 대량의 문자 데이터셋이 필요
데이터셋에는 알파벳, 숫자, 기호 등으로 구성
된 다양한 문자 이미지가 포함

분류(classification) 문제
출력값: 0~44개의 정수로 표현

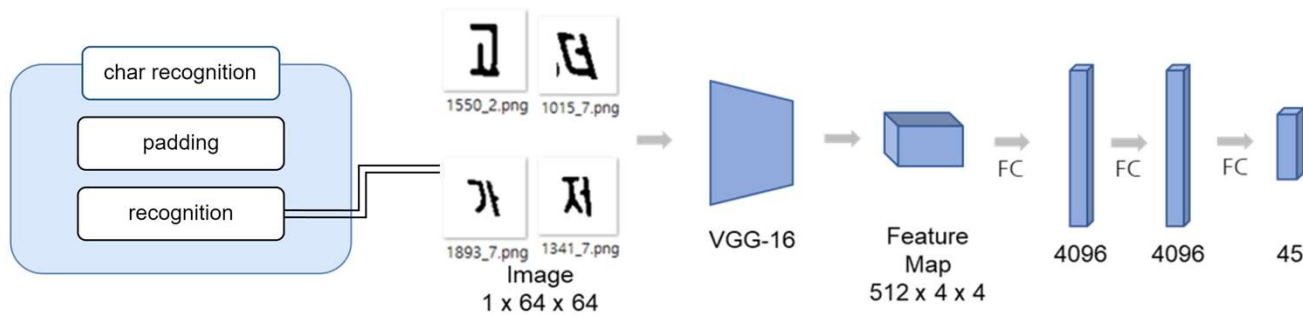
달성 목표





Padding 방법

1. Segmentation된 문자들을 crop한다
2. Recognition model의 input size에 맞게 padding(의미 없는 값을 추가)한다
crop한 이미지의 w, h 크기를 구한 뒤, OpenCVdml copyMakeBorder함수를 사용하여 64x64 image를 생성한다



학습 방법

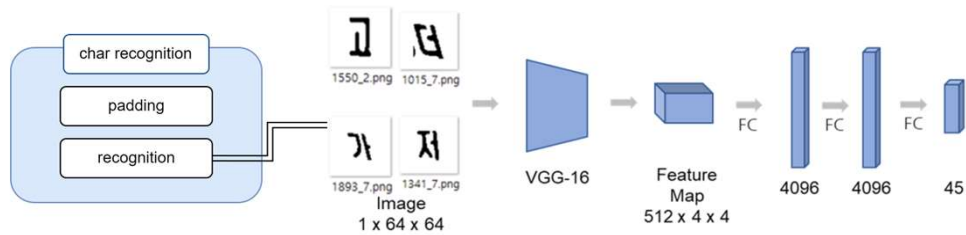
1. 이미지가 neural network를 지나 출력값을 도출 한다.
출력값: 0 ~ 44
2. 이미지의 Labeled 값(0~44)과 비교한다.
3. 이미지의 출력값과 labeled 값과의 차이값을 neural network에 학습한다. 차이값은 Cross-entropy를 사용한다
4. 1,2,3과정을 100번 반복한다. (epoch: 100)

Data 구성(총 45개)

1. 숫자: 0 ~ 9
2. 한글: [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,하,허,호]

3. SCdream, KBIZ한마음 font로 데이터 생성

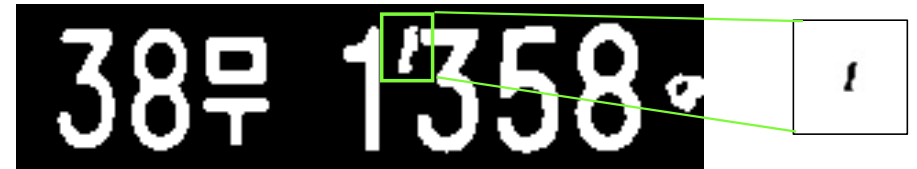
Labelling: 숫자 0 ~ 9, 한글 10 ~ 44



결과



문제점 발생(미완)



1) 노이즈를 문자로 인식하는 경우 발생

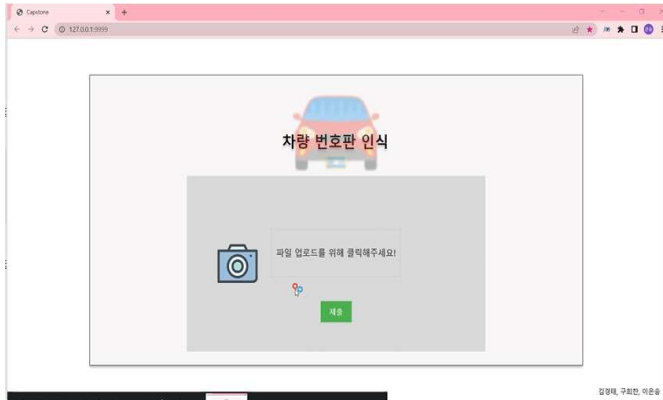


2) 버스, 택시, 개정 전 plate의 정확도 하락

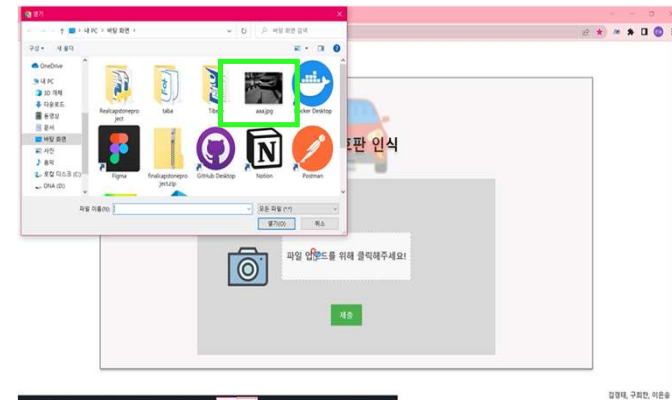
6

구축 결과

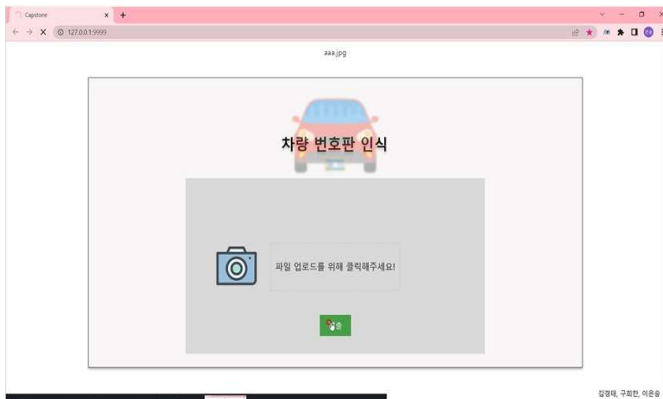
서비스 과정



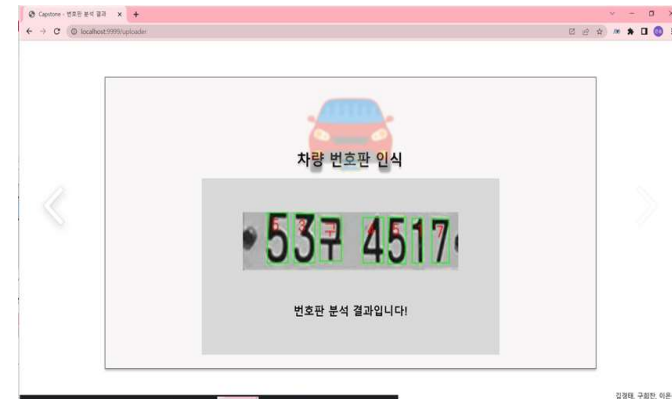
1) 메인 화면 구축



2) 이미지 업로드 기능

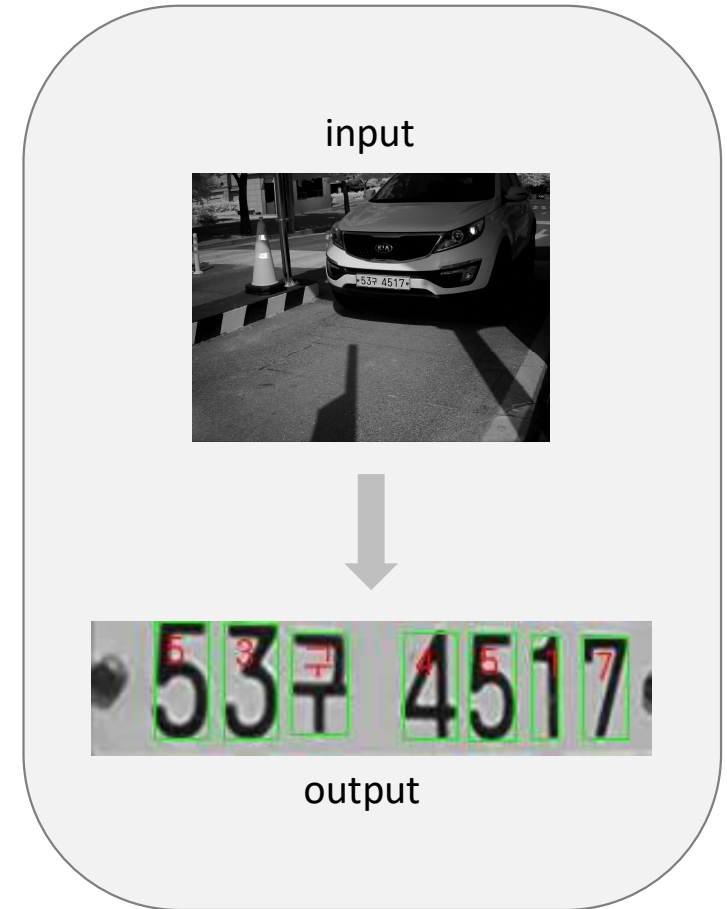


3) 이미지 제출



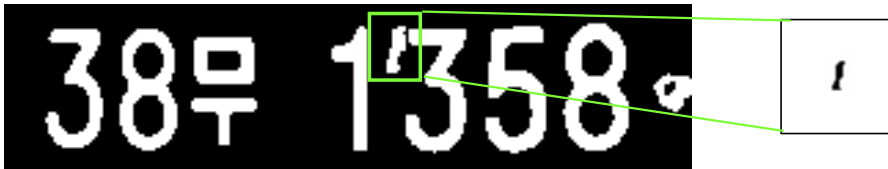
4) 결과 화면

PHASE	Data	Error	Accuracy(%)
LP Localization	3746	108	97.11
Char Recognition	21878	49	99.71
Performance speed	3 ~4 second		



분류	항목	3월				4월				5월				6월		
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3
web 개발	사용자 요구사항 분석															
	웹 페이지 구축 계획서															
	prototype 제작															
	웹 페이지 구축															
	웹 페이지 수정															
localization 개발	데이터 셋 분석															
	localization model 개발															
	성능 평가 및 수정															
segmentation 개발	contour 작업															
	오류 수정(모음, 자음)															
	오류 수정(그림자)															
recognition 개발	문자 데이터 생성															
	recognition model 개발															
	성능 평가 및 수정															
오픈	서비스 통합테스트															
	보고서 작성															

Recognition 부분



1) 노이즈를 문자로 인식하는 경우 발생



2) 버스, 택시, 개정 전 plate의 정확도 하락

1) 노이즈를 문자로 인식하지 않는 모델 구현

-> Sigmoid 함수를 이용하여 해당 문자의 확률이 기준보다 낮을 경우 노이즈로 판단한다 (캡스톤2에 구현할 예정)

2) 특수한 plate 전용 모델을 구현하기

[1] Characters Segmentation and Recognition for Vehicle License Plate - Minh Thang Dang

[2] Rediscovering Semantic Segmentation | by Farhan Ahmad | 100 Shades of Machine Learning | Medium

[3] OpenCV: Automatic License/Number Plate Recognition (ANPR) with Python - PyImageSearch

[4] Real-Time License Plate Recognition using Raspberry Pi and Python

[5] SSD : Single Shot Detector for object detection using MultiBox | by Renu Khandelwal | Towards Data Science

[6] From Masks to Rotating Bounding Boxes using OpenCV | Kaggle