


수리수리 차수리

에이콘 아카데미 3팀 카수리남



A photograph of a person wearing a white lab coat and gloves, working on a car tire. The person is using a tool to adjust the tire. The background is a blue wall. The image is overlaid with a semi-transparent white banner.

팀원 소개

팀원 소개

역할	훈련생	담당 업무
팀장	김영 *	프로젝트 총 관리 데이터 정규화 서버 구축
팀원	배연 *	학습 모델 연구 모델 생성 모델 테스트
	권혁 *	데이터 가공 데이터 연동 관리
	엄익 *	서비스 시스템 설계 모델 생성 데이터 전처리
	<u>인정</u> *	프론트엔드 설계 백엔드 설계 학습 데이터 수집



CONTENTS 목차

01

프로젝트 개요

- 배경
- 목적
- 기대효과
- 일정
- 사용기술

02

데이터 수집 및 분석

- 수집
- 전처리
- 분석
- 평가

03

서비스 설계 및 구현

- 설계
- 구현
- 테스트

04

프로젝트 시연

05

질의응답



01 프로젝트 개요

- 1.배경
- 2.목적
- 3.기대효과
- 4.일정
- 5.사용기술



배경

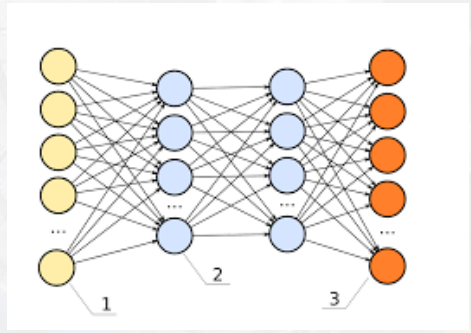
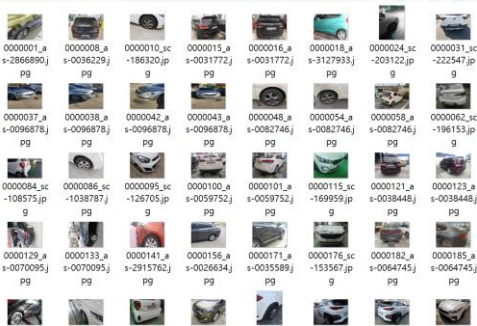
교통 사고 발생 시, 대부분의 운전자들은 전문지식의 부재로 즉각적이고 합리적인 자동차 수리 비용을 알기 힘들

객관적인 사고 차량 수리 비용을 정확하고 빠르게 추정하여 제공하는 서비스가 필요한 사람이 많을 것으로 판단



목적

다양한 차종의 사고 이미지 데이터를 딥러닝 기법을 활용하여
각 부품의 손상 부위, 손상 종류를 딥러닝 학습,
사고 차량의 수리비 견적을 객관적으로 예상하는 서비스 제공



기대 효과

- 1. 신속한 결정 지원** - 차량 사고 발생 시 손상된 위치 및 정도를 현장에서 즉각 객관적이고 보편적이게 예상 수리비를 산출하여 운전자에게 빠르게 전달하여 수리 및 보험 청구에 대해 대비
- 2. 비용 절감 기여** - 운전자 및 보험사의 수리비용 추정 및 보험 청구에 걸리는 시간을 최소화함으로써 타임 코스트 절감
- 3. 정확도 및 일관성 향상** - 딥러닝 알고리즘을 통해 정확한 분석 제공, 각 전문가마다 주관적 견해가 있기 때문에 그러한 오류를 최소화하고 운전자와 보험사에 더욱 신뢰도 있는 정보 제공 기대

활용 기술

Framework

django

IDE



Visual Studio Code

Web



Cloud Server



DB



Amazon
RDS

Deep Learning



TensorFlow



python™

Group Tool



Notion



Google Drive



02 데이터 수집 및 분석

1. 수집

2. 가공

3. 분석

4. 평가



수집

aihub.or.kr

AI 허브 : 한국지능정보사회진흥원에서 운영하는 개발 및 활용을 위한 데이터 무료 지원 사이트

차량 파손 데이터 이용 (용량 47 GB)

차량 파손 이미지 632,694장과 사고 자동차 견적 정보 125,007건이 기록된 엑셀 파일



#차량 파손

#차량 손상 유형

#차량 손상 부위

차량 파손 이미지 데이터

분야 교통물류 유형 이미지

갱신년월 : 2023-05 구축년도 : 2021 조회수 : 10,109 다운로드 : 561 용량 : 47.04 GB

다운로드

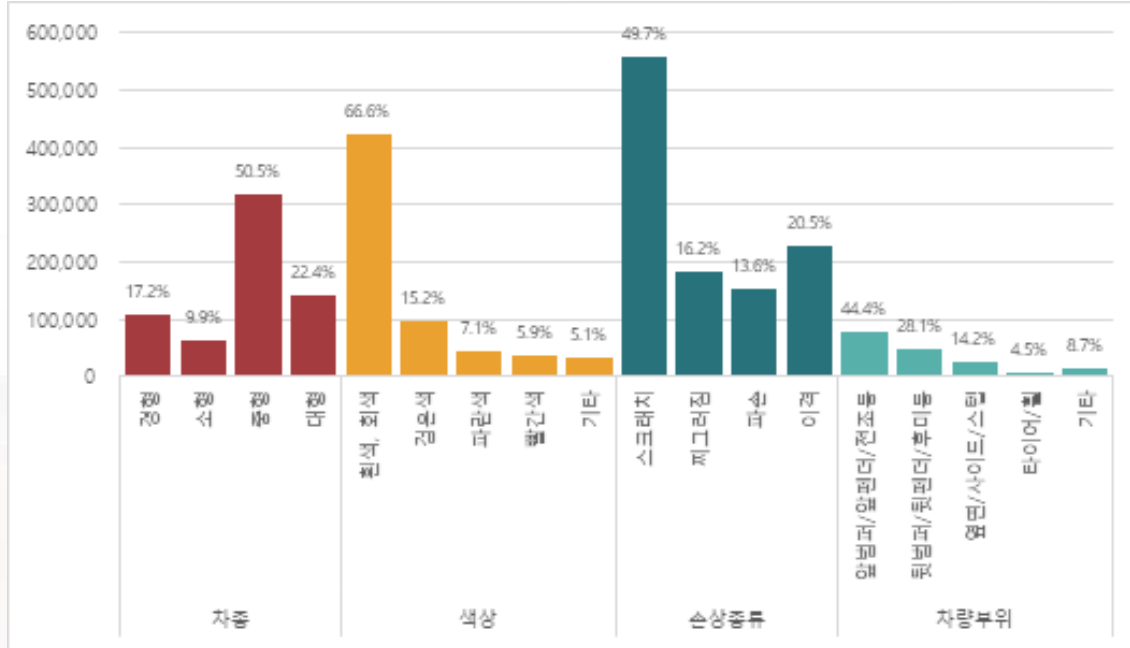
↓ 샘플 데이터

관심데이터 등록

👍 55

탐색 자료 분석 (EDA) 결과

각 항목에 따른 데이터 분포



차종

1. 중형
2. 소형
3. 대형
4. 기타

차량부위

1. 흰색/회색
2. 검은색
3. 파란색
4. 빨간색
5. 기타

손상종류

1. 스크래치
2. 찌그러짐
3. 파손
4. 이격

차량부위

1. 앞범퍼/앞펜더/전조등
2. 뒷범퍼/뒷펜더/후미등
3. 옆면/사이드/스텝
4. 타이어/휠

앞범퍼/앞펜더/전조등	77,830	44.40%
뒷범퍼/뒷펜더/후미등	49,275	28.10%
옆면/사이드/스텝	24,917	14.20%
타이어/휠	7,960	4.50%
기타	15,274	8.70%
합계	175,256	100.00%

파손 데이터 정보

```
{
  "info": {
    "name": "external",
    "date_created": "03/08/2022"
  },
  "images": {
    "id": 1,
    "width": 800,
    "height": 600,
    "file_name": "0000001_as-2866890.jpg"
  },
  "annotations": {
    {
      "id": 2,
      "image_id": 1,
      "category_id": "as-2866890",
      "segmentation": {
        {
          {
            234,
            441
          },
          {
            231,
            419
          },
          {
            255,
            401
          },
          {
            281,
            390
          },
          {
            287,
            389
          }
        }
      }
    }
  }
}
```

"damage": "Scratched", 파손 유형

```
"part": null,
"year": 2019,
"color": "Others",
"level": null,
"repair": [
  "Front fender(L):sheet_metal"
]
```

"part": "Front fender", 파손 부위

```
"year": 2019,
"color": "Others",
"level": 4,
"repair": "Front fender(L):sheet_metal"
```



각 파손 이미지마다 달려있는 JSON파일에서 파손 유형, 부위 데이터 추출

파손 유형별 분류

```
for annotation in annotations:
```

파손 여부

```
    if annotation['damage'] is not None:
        damage.append(annotation['damage'])

    elif annotation['part'] is not None:
        part = annotation['part']
```

```
    image_folder = img_save_dir + part
    imagename = train_dir + file_name
    damage = set(damage)
```

파손 종류

```
    if 'Scatched' in damage and len(damage) == 1:
        image_folder += '_s'
        saveimg(imagename, image_folder)
    else:
        image_folder += '_c'
        saveimg(imagename, image_folder)
```

Front bumper_c	Rear door(L)_s
Front bumper_s	Rear fender(R)_c
Front Wheel(R)_c	Side mirror(L)_c
Rear bumper_c	Side mirror(R)_c
Rocker panel(R)_c	Trunk lid_c
Bonnet_s	Trunk lid_s
Front door(L)_c	Head lights(L)_c
Front door(R)_s	Rear fender(L)_c
Front fender(L)_c	Rear Wheel(L)_s
Front fender(R)_c	Head lights(R)_c
Front Wheel(R)_s	Rear door(R)_c
Rear bumper_s	Front Wheel(L)_c
	Rear Wheel(R)_c

Bonnet	2817	4.27%
Bumper	38120	57.77%
Door	5682	8.61%
Head Light	4534	6.87%
Lamp	1667	2.53%
Rocker Panel	1759	2.67%
Side Mirror	3357	5.09%
Trunk Lid	2665	4.04%
Wheel	5380	8.15%
Sum	65981	100.00%

탐색 자료 분석 (EDA) 결과

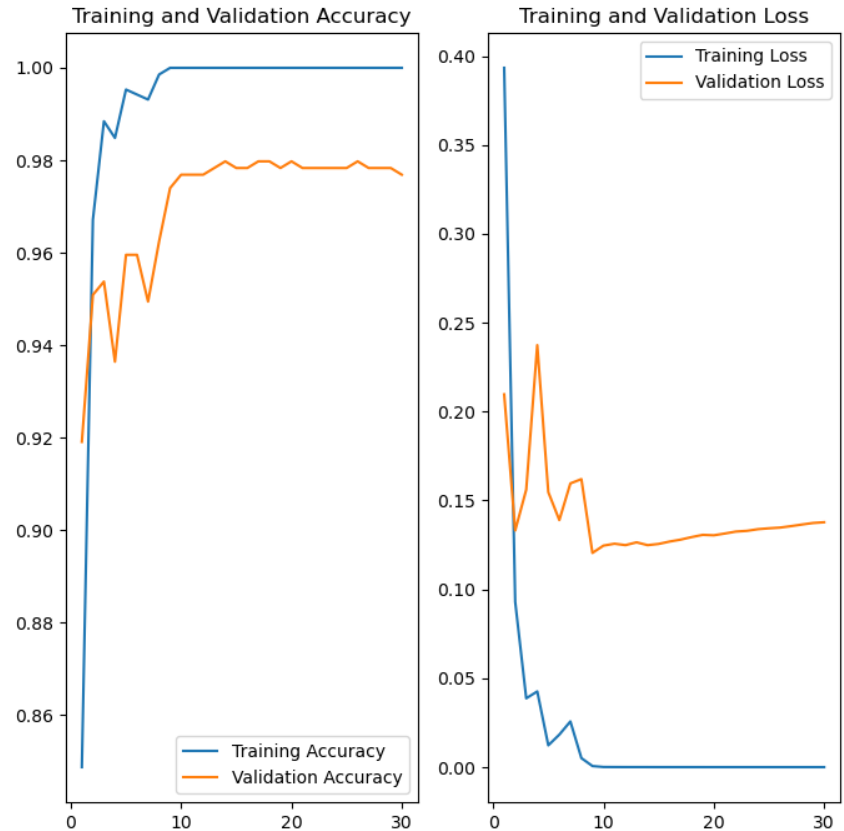
파손 유형별 분류된 이미지 중
최초 특징값이 가장 뚜렷한 부위 3가지 선택

1. Front Bumper (앞범퍼)
2. Side Mirror (사이드 미러)
3. Wheel (바퀴)

각 부위에서 학습용 샘플 이미지 100개 추출

기본 CNN 학습 진행

=> 약 98% 정확도



탐색 자료 분석 (EDA) 결과

파손 종류 레이블 추가
6개 클래스 학습 진행

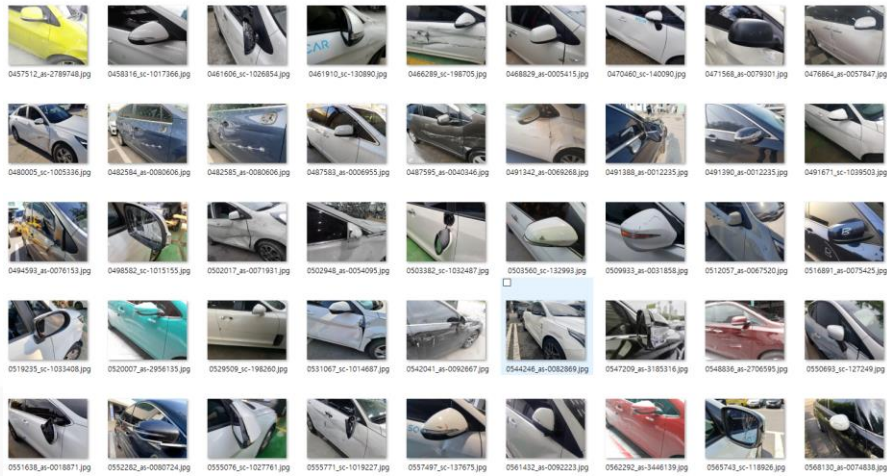
CNN, ResNet50, MobileNetV3 모델 학습 후 비교

=> Validation Loss, Accuracy 수치가 불안정

과적합 발생

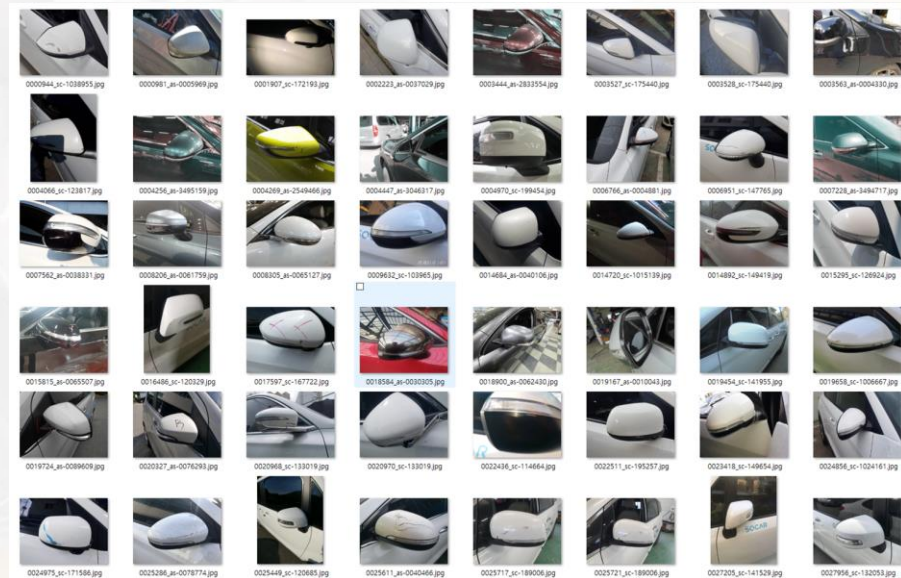


데이터 전처리



원본 데이터 :
초점이 맞지 않건, 너무 멀리서 찍히는등,
특징을 잡기 어려운 사진 포함

데이터 전처리 이후 :
특징을 잡기 쉬운 사진만 남겨 학습에 용의하게 준비함



모델링

1. 데이터셋 분할

```
train_ds = tf.keras.utils.image_dataset_from_directory(
    data_dir,
    validation_split=0.2,
    subset="training",
    seed=123,
    image_size=(img_height, img_width),
    batch_size=batch_size)

val_ds = tf.keras.utils.image_dataset_from_directory(
    data_dir,
    validation_split=0.2,
    subset="validation",
    seed=123,
    image_size=(img_height, img_width),
    batch_size=batch_size)
```

2. 그레이 스케일 및 엣지

```
# 그레이 스케일 및 에지 감지 함수
def preprocess_image(img):
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    gray = cv2.cvtColor(gray, cv2.COLOR_GRAY2RGB)
    edges = cv2.Canny(gray.astype(np.uint8), 80, 150)
    edges = cv2.cvtColor(edges, cv2.COLOR_GRAY2RGB)
    return edges.astype(np.float32)
```

3. 데이터 증강

데이터 증강 생성기

```
data_augmentation = ImageDataGenerator(
    preprocessing_function=preprocess_image,
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)
```

4. 데이터 표준화

데이터 표준화

```
normalization_layer = layers.Rescaling(1./255)

normalized_ds = train_ds.map(lambda x, y: (normalization_layer(x), y))
image_batch, labels_batch = next(iter(normalized_ds))
first_image = image_batch[0]
# Notice the pixel values are now in `[0,1]`.
print(np.min(first_image), np.max(first_image))
```

모델링 데이터 분석 및 결과 (CNN)

모델 생성

```
num_classes = len(class_names)

model = Sequential([
    layers.Rescaling(1./255, input_shape=(img_height, img_width, 3)),
    layers.Conv2D(16, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(32, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(64, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Dropout(0.2),
    layers.Flatten(),
    layers.Dense(128, activation='relu'),
    layers.Dense(64, activation='relu'),
    layers.Dense(num_classes)
])
```

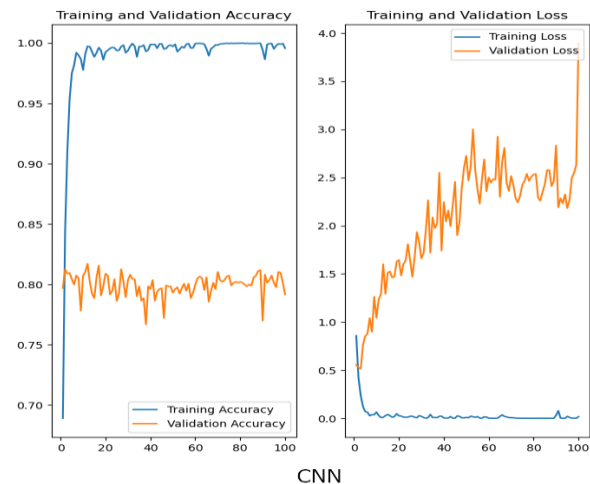
모델 컴파일

```
model.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])

epochs=100

history = model.fit(
    train_ds,
    validation_data=val_ds,
    epochs=epochs.)

model.save('cnnModel_4part_0620_.h5')
```



좌 : Accuracy
우 : Loss

모델링 데이터 분석 결과 (ResNet50)

모델 생성

```
num_classes = len(class_names)
```

```
model = Sequential([  
    layers.Rescaling(1. / 255, input_shape=(img_height, img_width, 3)),  
    ResNet50(include_top=False, weights='imagenet', pooling='avg'),  
    layers.Dense(num_classes)  
])
```

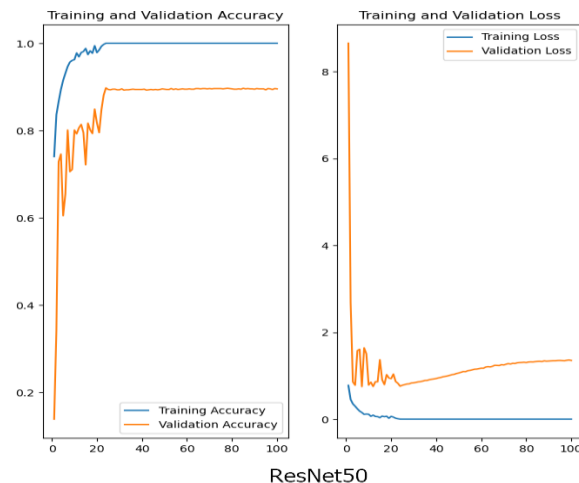
모델 컴파일

```
model.compile(optimizer='adam',  
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),  
              metrics=['accuracy'])
```

```
epochs = 100
```

```
history = model.fit(  
    train_ds,  
    validation_data=val_ds,  
    epochs=epochs,  
    callbacks=[chk])
```

```
model.save('resnet50_gray_edge2_0619.h5')
```



좌 : Accuracy
우 : Loss

모델링 데이터 분석 결과 (MobileNetV3)

모델 생성

```
num_classes = len(class_names)

model = tf.keras.applications.MobileNetV3Small(
    input_shape=(img_height, img_width, 3),
    alpha=1.0,
    minimalistic=False,
    include_top=True,
    weights="imagenet",
    input_tensor=None,
    classes=1000,
    pooling=None,
    dropout_rate=0.2,
    include_preprocessing=True,
)
```

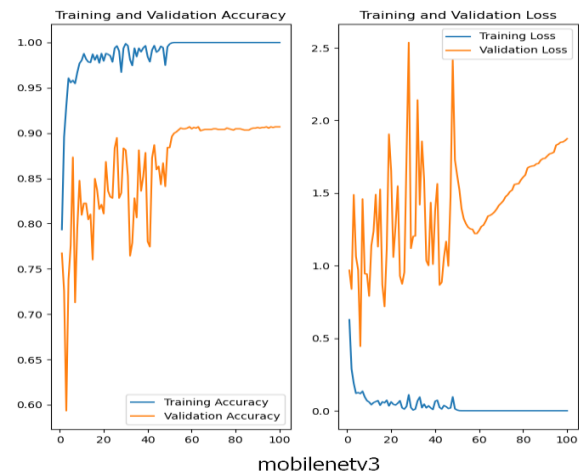
모델 컴파일

```
model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=0.001),
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])

epochs = 100

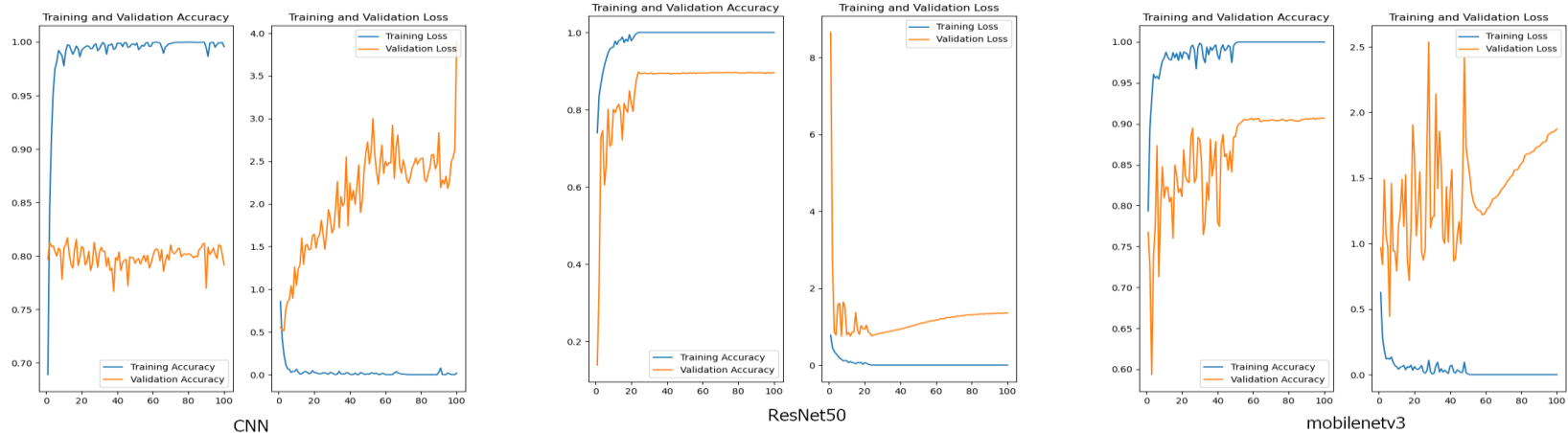
history = model.fit(
    train_ds,
    validation_data=val_ds,
    epochs=epochs,
)

model.save('mobilenetV3_4part_gray_edge_0620.h5')
```



좌 : Accuracy
우 : Loss

데이터 분석 결과 비교 (CNN vs ResNet50 vs MobileNetV3)



비교 결과 : 기본 CNN, Resnet50, MobileNetV3 small 세 가지 모델의 학습 그래프 비교를 진행함

처음에는 모델 모두 전형적인 과적합 그래프를 보임.

판별이 어려워보이는 이미지를 제거하고, 학습 단계에서 gray scale을 입히고, opencv의 canny함수를 이용해 외곽선을 검출해 학습시킨 결과 epoch 3,40 이후로 안정된 곡선을 보임.

여러 모델 중 높은 val acc를 보인 MobileNetV3 small을 선택해서 학습을 진행함

03 서비스 설계 및 구현

1.설계

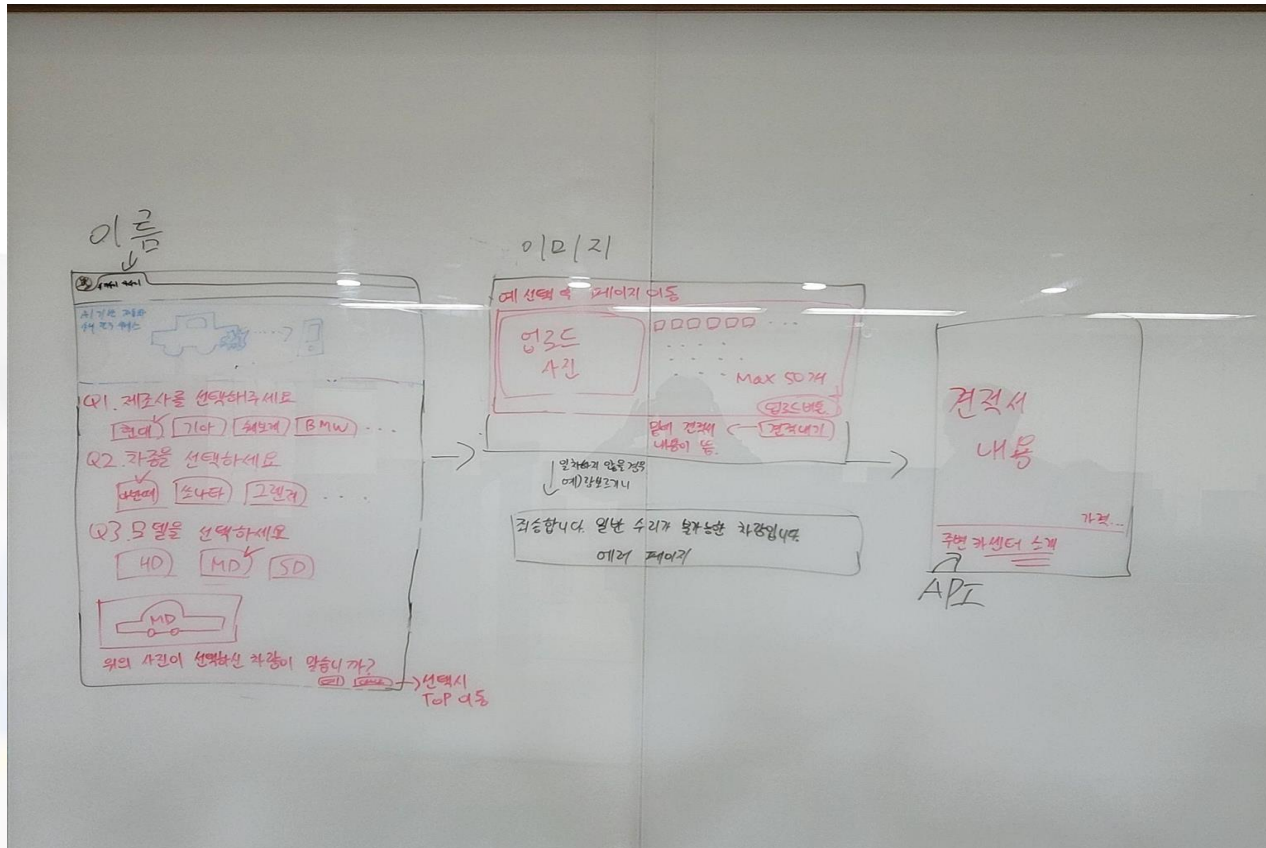
2.구현

3.테스트



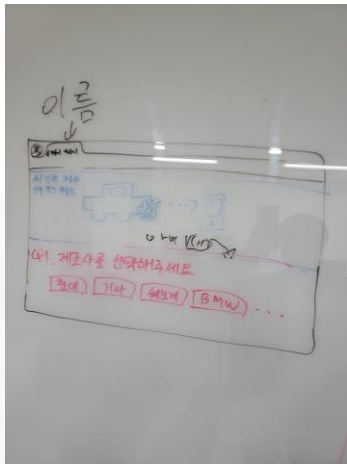
서비스 설계 (스토리보드)

서비스 이용자의 시점에서 사용시 화면 구성과 기능 구조 설립



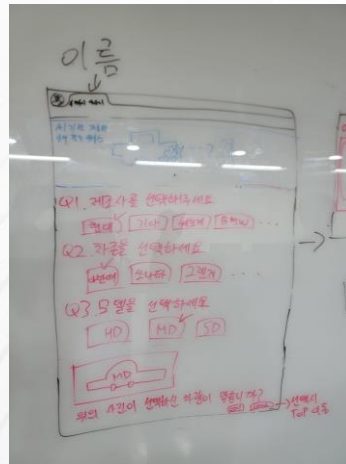
서비스 설계 (스토리보드)

서비스 이용자의 시점에서 사용시 화면 구성과 기능 구조 설립



첫화면

제조사 선택해 주세요
[현대], [기아], [쌍용]...



차종 선택
사진 업로드

제조사 선택...
차종 선택...
모델 선택...

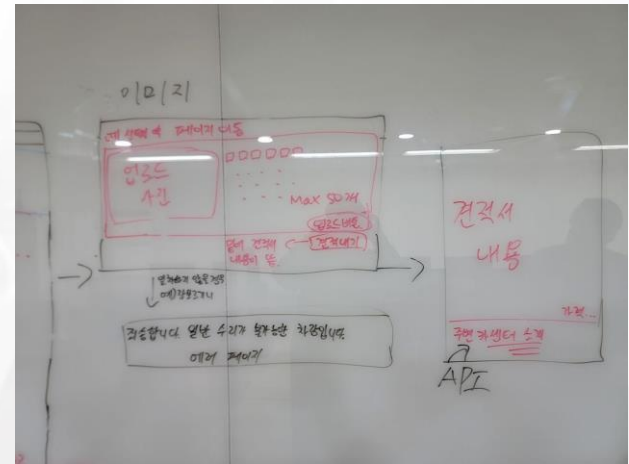


사진 분석
견적서 제출

사진 확인

에러

견적서

서비스 설계 (DB)

자동차 제조사와 모델로 분류하여 DB 작성

자동차 종류 정리 ☆ 📄

파일 수정 보기 삽입 서식 데이터 도구 확장 프로그램 도움말

100% W % 123 기본값... - 10 + B I U A

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	maker1	model	detail	maker2	model	detail	maker3	model	detail	maker4	model	detail	maker5	model	detail
2	기아	K3	K3	르노삼성	SM7	All New SM7	쌍용	렉스턴	G4 렉스턴	한국GM	스파크	더넥스트스파크	현대	EQ900	EQ900
3	기아	K3	K3(18)	르노삼성	SM5	New SM5 플래티넘	쌍용	렉스턴	뉴렉스턴	한국GM	말리부	말리부	현대	G70	G70(17)
4	기아	K3	K3(4도어)(16)	르노삼성	QM3	QM3	쌍용	렉스턴	렉스턴	한국GM	스파크	스파크(마티즈크리에이티브)	현대	G80	G80(16)
5	기아	K5	K5	르노삼성	QM5	QM5	쌍용	렉스턴	렉티플코란도	한국GM	아베오	아베오(세단)	현대	G80	G80(20)-RG3
6	기아	K5	K5(15)	르노삼성	QM5	QM5(11)	쌍용	코란도	뷰티풀코란도	한국GM	아베오	아베오(해치백)	현대	G70	GV70(20)-JK1
7	기아	K5	K5(18)	르노삼성	QM5	QM5(내오)(14)	쌍용	제어맨W	제어맨W	한국GM	알파온	알파온	현대	G70	GV70(21)
8	기아	K5	K5(19)-DL3	르노삼성	QM6	QM6	쌍용	코란도	코란도 C	한국GM	말리부	올뉴말리부	현대	G80	GV80(20)
9	기아	K7	K7	르노삼성	SM3	SM3 (09)	쌍용	코란도	코란도 투리스모	한국GM	말리부	올뉴말리부(2017)	현대	G80	GV80(20)-JX1
10	기아	K9(21)-GL3	K9(21)-GL3	르노삼성	SM3	SM3 (2.0)	쌍용	코란도	코란도스포즈	한국GM	크루즈	올뉴크루즈(17)	현대	그랜저	HG그랜저
11	기아	K9	K9	르노삼성	SM5	SM5 (2.5)	쌍용	티볼리	티볼리	한국GM	올인도	올인도	현대	그랜저	HG그랜저
12	기아	K9	K9(15)	르노삼성	SM5	SM5 Nova				한국GM	윈스름	윈스름	현대	i30	i30
13	기아	K9	K9(18)	르노삼성	SM5	SM5 TCE(13)				한국GM	윈스름	윈스름엑스	현대	i30	i30 (2012)
14	기아	카니발	그랜드카니발	르노삼성	SM5	SM5(10)				한국GM	임팔라	임팔라	현대	i30	i30(2017)
15	기아	모닝	뉴 모닝	르노삼성	SM6	SM6				한국GM	임팔라	임팔라(15)-10TH	현대	i40	i40
16	기아	쏘렌토	뉴쏘렌토R(13)	르노삼성	SM7	SM7 Nova				한국GM	라세티	크루즈(라세티 프리미어)	현대	i40	i40 살롱
17	기아	카니발	뉴카니발9인승(06)	르노삼성	SM3	XM3(20)-LJL				한국GM	트렉스	트렉스	현대	i40	i40 살롱(15)
18	기아	카렌스	뉴카렌스(06)	르노삼성	SM3	뉴 SM3							현대	i40	i40(15)
19	기아	프라이드	뉴프라이드4DR(05)	르노삼성	SM5	뉴 SM5							현대	쏘나타	LF 쏘나타
20	기아	프라이드	뉴프라이드5DR(05)	르노삼성	SM3	뉴SM3(2012)							현대	쏘나타	LF 쏘나타 뉴라이즈
21	기아	니로	니로	르노삼성	클리오	클리오							현대	쏘나타	YF 쏘나타
22	기아	니로	니로EV(19)										현대	쏘나타	YF쏘나타 하이브리드
23	기아	레이	레이										현대	스타렉스	그랜드스타렉스 3번
24	기아	레이	레이(18)										현대	스타렉스	그랜드스타렉스 3번(15)
25	기아	모닝	모닝										현대	스타렉스	그랜드스타렉스 5번
26	기아	모하비	모하비										현대	스타렉스	그랜드스타렉스 5번(15)
27	기아	모하비	모하비(16)										현대	스타렉스	그랜드스타렉스 웨건
28	기아	셀토스	셀토스(19)										현대	스타렉스	그랜드스타렉스 웨건(15)
29	기아	스토닉	스토닉										현대	스타렉스	그랜드스타렉스3번(18)
30	기아	스팅어	스팅어(17)										현대	스타렉스	그랜드스타렉스5번(18)
31	기아	스포티지	스포티지(16)										현대	스타렉스	그랜드스타렉스웨건(18)
32	기아	쏘렌토	쏘렌토R										현대	그랜저	그랜저(19)-IG
33	기아	쏘울	쏘울										현대	그랜저	그랜저(IG)(17)
34	기아	쏘울	쏘울(16)										현대	그랜저	그랜저(IG)(20)
35	기아	K7	올 뉴 K7										현대	그랜저	그랜저(IG)(17)
36	기아	모닝	올 뉴 모닝(11)										현대	엑센트	뉴엑센트(4Dr)
37	기아	쏘울	올 뉴 쏘울										현대	엑센트	뉴엑센트(5Dr)

시트1 | 탐색

서비스 설계 (DB)

수리 정보 :

- 제작사
- 차 종류
- 수리 비용
- 수리 부품

filtered_data_repair(수리정보 없음) ☆ 田

파일 수정 보기 삽입 서식 데이터 드루 확장 프로그램 도움말

100% W % 123 기본값 10 B I U A

K141 Front bumper:repair,coating

	A	B	C	D	E	F	G	H	I	J	K
1	category_id	cnt	maker	model	detail	model_detail	cost_wages	cost_repairparts	cost_total	part	repair
2	as-0000034	21	기아	K5	K5(18)	가솔린 2.0 인텔리전트	282728	0	282728		Front bumper:repair,coating
3	as-0000037	23	기아	쏘렌토	솔뉴쏘렌토	R2.0 디젤 디럭스	287178	0	287178	Front bumper	Front bumper:repair,coating
4	as-0000208	122	기아	카니발	솔뉴카니발(14)	노블레스(11인승)	588110	0	588110		Bumper:repair;Front bumper:repair,coating
5	as-0000291	171	현대	G80	G80(16)	3.3 스포즈	621294	0	621294		Front bumper:repair,coating
6	as-0000320	190	현대	아반떼	아반떼AD(16)	1.6 GDI 프리미엄	288716	0	288716		Front bumper:repair,coating
7	as-0000347	203	현대	그랜저	그랜저IG(17)	가솔린 3.0 엑스	298016	0	298016		Front bumper:repair,coating
8	as-0000348	204	기아	K5	K5(15)	가솔린 2.0 프레스티지	310916	0	310916		Front bumper:repair,coating
9	as-0000634	354	기아	모닝	솔뉴 모닝(11)	락셔리	190258	0	190258		Front bumper:coating
10	as-0000664	371	현대	아반떼	아반떼AD(16)	1.6 GDI 프리미엄	280246	0	280246		Front bumper:repair,coating
11	as-0000716	396	현대	스타렉스	그랜드스타렉스	5인승(CVX)	338096	0	338096		Front bumper:repair,coating
12	as-0000720	399	한국GM	말리부	솔뉴말리부(2011)	1.6 터보 LTZ	315966	0	315966	Front bumper	Front bumper:repair,coating
13	as-0000721	400	현대	그랜저	그랜저IG(17)	가솔린 2.4 프리미엄 스포셜	313136	0	313136		Front bumper:repair,coating
14	as-0000731	404	현대	쏘나타	LF 쏘나타	2.0 CVVL PREMIUM	287616	0	287616		Front bumper:repair,coating
15	as-0000782	433	현대	그랜저	HG그랜저	HG 240 (15)	285098	0	285098		Front bumper:coating
16	as-0000806	448	현대	투싼	솔뉴투싼	R2.0 로먼	300506	11400	317456		Front bumper:repair,coating
17	as-0000980	525	기아	K3	K3(4도어)(16)	프레스티지(가솔린)	200546	0	200546		Front bumper:repair,coating
18	as-0001093	618	기아	모닝	솔뉴모닝(15)	락셔리	240316	0	240316		Front bumper:repair,coating
19	as-0001182	668	현대	EQ900	EQ900	3.8 GDI 프레스티지	502884	0	502884		Front bumper:repair,coating
20	as-0001187	672	기아	스포트지	스포트지(16)	노블레스 2.0	317218	0	317218		Front bumper:coating
21	as-0001282	722	현대	그랜저	HG그랜저	HG 300	318138	0	318138		Front bumper:coating
22	as-0001288	726	기아	쏘렌토	솔뉴쏘렌토	R2.0 디젤 프레스티지	286237	0	286237		Front bumper:repair;Bumper:coating
23	as-0001310	746	기아	모닝	솔뉴모닝(17)	레이디	155398	0	155398		Front bumper:coating
24	as-0001334	758	기아	쏘렌토	솔뉴쏘렌토	R2.0 디젤 노블레스	238339	0	238339	Front bumper	Bumper:coating,exchange
25	as-0001389	781	기아	K3	K3(18)	노블레스	314569	0	314569		Bumper:coating
26	as-0001372	783	현대	쏘나타	LF 쏘나타 뉴라C	2.0 로먼	372582	0	372582		Front bumper:repair,coating
27	as-0001376	786	기아	쏘렌토	솔뉴쏘렌토	R2.0 디젤 노블레스	261179	0	261179		Bumper:coating
28	as-0001405	800	현대	아반떼	아반떼AD(16)	1.6 GDI 프리미엄	420079	0	420079		Bumper:coating
29	as-0001416	807	현대	아반떼	아반떼AD(16)	1.6 GDI 프리미엄	309469	0	309469		Bumper:coating,exchange
30	as-0001417	808	기아	K3	K3(18)	프레스티지	225029	0	225029		Bumper:coating,exchange
31	as-0001440	824	현대	싼타페	싼타페DM	디젤 2.2 2WD	488126	0	488126		Bumper:repair;Front bumper:repair,coating
32	as-0001478	845	현대	아반떼	아반떼AD(16)	1.6 GDI 로먼	224799	0	224799	Rear bumper	Bumper:coating,exchange
33	as-0001488	857	기아	스포트지	스포트지(16)	노블레스 2.0	389756	0	389756	Front bumper	Front bumper:repair,coating
34	as-0001580	908	기아	모닝	솔뉴모닝(17)	터보프레스티지	326823	0	326823		Front bumper:repair,coating
35	as-0001592	917	기아	쏘렌토	솔뉴쏘렌토	R2.0 디젤 프레스티지	301937	0	301937	Front bumper	Front bumper:repair;Bumper:coating
36	as-0001595	919	기아	쏘렌토	솔뉴쏘렌토(18)	2.2 디젤 노블레스 스포셜	383788	0	383788		Front bumper:repair,coating
37	as-0001611	932	기아	K3	K3(18)	노블레스	300649	0	300649	Front bumper	Bumper:coating,exchange
38	as-0001630	948	기아	K3	K3(18)	노블레스	374009	0	374009		Bumper:coating
39	as-0001638	952	현대	쏘나타	LF 쏘나타 뉴라C	2.0 로먼	402846	0	402846		Front bumper:repair,coating
40	as-0001642	954	현대	쏘나타	LF 쏘나타	2.0 CVVL PREMIUM	483177	0	483177		Front bumper:repair;Bumper:coating,exchange
41	as-0001676	978	기아	모닝	솔뉴모닝(15)	락셔리	247998	0	247998		Front bumper:coating
42	as-0001693	984	기아	K3	K3(18)	프레스티지	831127	0	831127		Front bumper:repair;Bumper:coating
43	as-0001703	992	쌍용	티볼리	티볼리	LX (가솔린 고급형)	748886	0	748886	Bonnet	Front bumper:repair,coating
44	as-0001714	999	현대	그랜저	그랜저IG(17)	가솔린 2.4 프리미엄	404736	0	404736		Front bumper:repair,coating
45	as-0001716	1001	기아	니르	니르EV(19)	노블레스	358819	0	358819		Bumper:coating,exchange
46	as-0001733	1014	현대	아반떼	아반떼AD(16)	1.6 GDI 로먼	392049	0	392049		Bumper:coating,exchange
47	as-0001751	1027	기아	K7	K7	VG 270 엑스	332158	0	332158		Front bumper:coating
48	as-0001773	1041	현대	싼타페	싼타페TM(18)	디젤 2.2 인스퍼레이션	295729	0	295729		Bumper:coating
49	as-0001794	1054	기아	K5	K5(15)	가솔린 2.0 노블레스	420729	0	420729		Front bumper:repair,coating
50	as-0001798	1056	기아	카니발	솔뉴카니발(14)	2.2디젤노블레스(9인승)	412376	0	412376		Front bumper:repair,coating

서비스 설계 (DB)

수리 정보 :

- 제작사
 - 차 종류
 - 수리 비용
 - 수리 부품
- +
- 수리내용
1. 교환
 2. 수리

filtered_data_repair(통합) ☆ 📄 🗑

파일 수정 보기 삽입 서식 데이터 도구 확장 프로그램 도움말

100% W % .00 123 기본값 10 + B I Z A 📏 📐 📑 📄 📅 📆 📇 📈 📉 📊 📋 📌 📍 📎 📏 📐 📑 📄 📅 📆 📇 📈 📉 📊 📋 📌 📍 📎

J917 📄 📑

1	A	B	C	D	E	F	G	H	I	J
	maker	model	detail	category_id	cnt	cost_total	repair	exchange	repaired	
449	447	기아	K7	K7	as-0058078	13544	345176	Front bumper:repair,coating		Front bumper:repair,coating
450	448	기아	K7	K7	as-0058651	12006	304016	Front bumper:repair,coating		Front bumper:repair,coating
451	449	기아	K7	K7	as-0054790	11528	350941	Front bumper:repair,coating		Front bumper:repair,coating
452	450	기아	K7	K7	as-0054030	11073	325468	Front bumper:repair,coating		Front bumper:repair,coating
453	451	기아	K7	K7	as-0053785	10939	317028	Front bumper:repair,coating		Front bumper:repair,coating
454	452	기아	K7	K7	as-0051378	9575	230748	Front bumper:coating		Front bumper:coating
455	453	기아	K7	K7	as-0048987	8082	383271	Front bumper:repair,coating		Front bumper:repair,coating
456	454	기아	K7	K7	as-0040121	2896	344201	Front bumper:repair,coating		Front bumper:repair,coating
457	455	기아	K7	K7	as-0037738	1415	248328	Front bumper:repair,coating		Front bumper:repair,coating
458	456	기아	K8	K8(21)-GL3	as-0005392	3170	384229	Bumper:coating,exchange	Bumper:coating,exchange	Front bumper:repair,coating
459	457	기아	K8	K8(21)-GL3	as-0082109	38208	806912	Front bumper:repair:Bumper:coating,exchange	Front bumper:repair:Bumper:coating,exchange	Front bumper:repair,coating
460	458	기아	K8	K8(21)-GL3	as-0092076	53395	725532	Front bumper:repair:Bumper:coating		Front bumper:repair:Bumper:coating
461	459	기아	K8	K8(21)-GL3	as-0092076	33395	725532	Front bumper:repair:Bumper:coating		Front bumper:repair:Bumper:coating
462	460	기아	K8	K8(21)-GL3	as-0082109	18208	806912	Front bumper:repair:Bumper:coating,exchange	Front bumper:repair:Bumper:coating,exchange	Front bumper:repair,coating
463	461	기아	K9	K9	as-0003523	2044	325718	Front bumper:repair,coating		Front bumper:repair,coating
464	462	기아	K9	K9	as-0022250	11815	276268	Front bumper:repair,coating		Front bumper:repair,coating
465	463	기아	K9	K9	as-0029767	15849	386278	Front bumper:repair,coating		Front bumper:repair,coating
466	464	기아	K9	K9	as-0029885	15928	491278	Front bumper:repair,coating		Front bumper:repair,coating
467	465	기아	K9	K9	as-0058201	33824	401956	Front bumper:repair,coating		Front bumper:repair,coating
468	466	기아	K9	K9	as-0081046	47711	540132	Front bumper:repair,coating		Front bumper:repair,coating
469	467	기아	K9	K9	as-0081046	27711	540132	Front bumper:repair,coating		Front bumper:repair,coating
470	468	기아	K9	K9	as-0058201	13824	401956	Front bumper:repair,coating		Front bumper:repair,coating
471	469	기아	K9	K9(15)	as-0005954	3448	406898	Front bumper:repair,coating		Front bumper:repair,coating
472	470	기아	K9	K9(15)	as-0033039	18039	323146	Front bumper:repair,coating		Front bumper:repair,coating
473	471	기아	K9	K9(15)	as-0038329	20504	387907	Front bumper:repair,coating		Front bumper:repair,coating
474	472	기아	K9	K9(15)	as-0059982	34850	398456	Front bumper:repair,coating		Front bumper:repair,coating
475	473	기아	K9	K9(15)	as-0083428	37032	419156	Front bumper:repair,coating		Front bumper:repair,coating
476	474	기아	K9	K9(15)	as-0083428	17032	419156	Front bumper:repair,coating		Front bumper:repair,coating
477	475	기아	K9	K9(15)	as-0059982	14850	398456	Front bumper:repair,coating		Front bumper:repair,coating
478	476	기아	K9	K9(15)	as-0038329	504	387907	Front bumper:repair,coating		Front bumper:repair,coating
479	477	기아	K9	K9(18)	as-0003265	1918	810299	Bumper:coating		Bumper:coating
480	478	기아	K9	K9(18)	as-0005348	3142	355169	Bumper:coating,exchange	Bumper:coating,exchange	Bumper:coating
481	479	기아	K9	K9(18)	as-0013344	6926	700599	Bumper:coating,repair		Bumper:coating,repair
482	480	기아	K9	K9(18)	as-0030112	16092	374559	Bumper:coating,exchange	Bumper:coating,exchange	Bumper:coating,repair
483	481	기아	K9	K9(18)	as-0030113	16093	486899	Bumper:coating,repair		Bumper:coating,repair
484	482	기아	K9	K9(18)	as-0034336	16006	321979	Bumper:coating,repair		Bumper:coating,repair
485	483	기아	K9	K9(18)	as-0034481	19122	248819	Bumper:coating,exchange	Bumper:coating,exchange	Bumper:coating,repair
486	484	기아	K9	K9(18)	as-0034833	19396	624729	Bumper:coating,repair		Bumper:coating,repair
487	485	기아	K9	K9(18)	as-0035149	19648	602208	Bumper:coating,repair		Bumper:coating,repair
488	486	기아	K9	K9(18)	as-0035175	19669	384249	Bumper:coating,exchange	Bumper:coating,exchange	Bumper:coating,repair
489	487	기아	K9	K9(18)	as-0035480	19904	785469	Bumper:coating,exchange	Bumper:coating,exchange	Bumper:coating,repair
490	488	기아	K9	K9(18)	as-0035557	10998	482387	Front bumper:repair:Bumper:coating		Front bumper:repair:Bumper:coating
491	489	기아	K9	K9(18)	as-0036818	20823	558817	Front bumper:repair:Bumper:coating		Front bumper:repair:Bumper:coating
492	490	기아	K9	K9(18)	as-0037121	21030	457819	Bumper:coating,repair		Bumper:coating,repair
493	491	기아	K9	K9(18)	as-0059968	34839	630847	Front bumper:repair:Bumper:coating		Front bumper:repair:Bumper:coating
494	492	기아	K9	K9(18)	as-0080621	35279	599587	Front bumper:repair:Bumper:coating		Front bumper:repair:Bumper:coating
495	493	기아	K9	K9(18)	as-0081439	35834	555349	Bumper:coating,exchange	Bumper:coating,exchange	Front bumper:repair:Bumper:coating
496	494	기아	K9	K9(18)	as-0081772	39019	738949	Bumper:coating,exchange	Bumper:coating,exchange	Front bumper:repair:Bumper:coating
497	495	기아	K9	K9(18)	as-0082468	39430	386176	Bumper:coating,exchange	Bumper:coating,exchange	Front bumper:repair:Bumper:coating

서비스 설계 (DB)

DB 접근 코드

CSV 파일 경로

CSV 파일을 데이터프레임으로 읽어오기

데이터베이스 연결

maker, model, detail 테이블에서 데이터 읽어오기

model 테이블에서 데이터 읽어오기

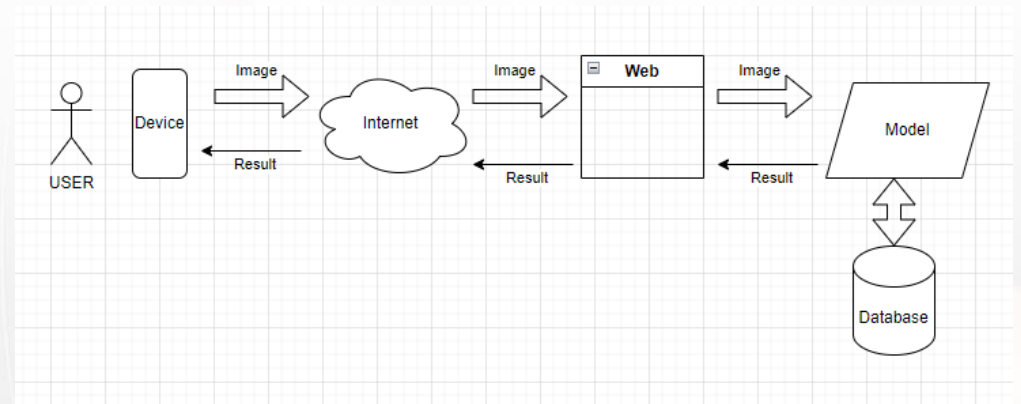
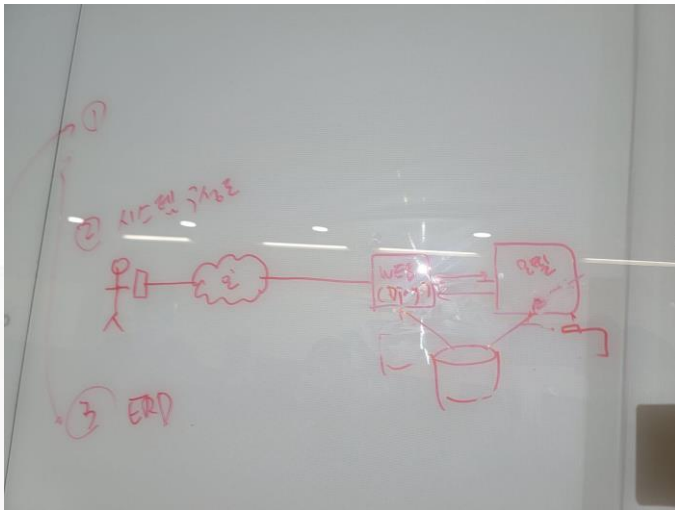
db에 저장

```
carinfor_repair_cost_db ☆
파일 수정 보기 삽입 런타임 도구 도움말 모든 변경사항이 저장됨
+ 코드 + 텍스트

1 import mysql.connector
2 import pandas as pd
3
4 # CSV 파일 경로
5 csv_file_path = 'data.csv'
6
7 # CSV 파일을 데이터프레임으로 읽어오기
8 df = pd.read_csv(r'filtered_data_repair(통합).csv', encoding='utf-8')
9 rp = pd.DataFrame(columns=['maker_num', 'model_num', 'detail_num', 'oost', 'exohange', 'repair'])
10
11 print(df[:10])
12 # 데이터베이스 연결
13 oonn = mysql.connector.connect(
14     host='127.0.0.1',
15     port = 3306,
16     user='root',
17     password='123',
18     database='oer'
19 )
20 cursor = oonn.cursor()
21
22
23
24 # maker 테이블에서 데이터 읽어오기
25 cursor.execute("SELECT maker_no, maker_name FROM maker")
26 rows = cursor.fetchall()
27 maker_table = pd.DataFrame(rows, columns=['maker_no', 'maker_name'])
28
29 # 데이터프레임(df)에서 maker_name과 일치하는 maker_no로 치환
30 rp['maker_num'] = df['maker'].map(maker_table.set_index('maker_name')['maker_no'])
31
32
33 # model 테이블에서 데이터 읽어오기
34 cursor.execute("SELECT model_no, model_name FROM model")
35 rows = cursor.fetchall()
36 model_table = pd.DataFrame(rows, columns=['model_no', 'model_name'])
37
38 # 데이터프레임(df)에서 model_name과 일치하는 model_num으로 치환
39 rp['model_num'] = df['model'].map(model_table.set_index('model_name')['model_no'])
40
41
42 # detail 테이블에서 데이터 읽어오기
43 cursor.execute("SELECT detail_no, detail_name FROM detail")
44 rows = cursor.fetchall()
45 detail_table = pd.DataFrame(rows, columns=['detail_no', 'detail_name'])
46
47 # 데이터프레임(df)에서 model_name과 일치하는 model_num으로 치환
48 rp['detail_num'] = df['detail'].map(detail_table.set_index('detail_name')['detail_no'])
49 # 'rp' 데이터프레임에 'oost', 'exohange', 'repair' 컬럼 추가
50 rp['oost'] = df['oost_total']
51 rp['exohange'] = df['exohange']
52 rp['repair'] = df['repair']
53
54 rp.to_csv('test(수리견적).csv', encoding='utf-8-sig')
55
```

서비스 설계 (시스템 구성도)

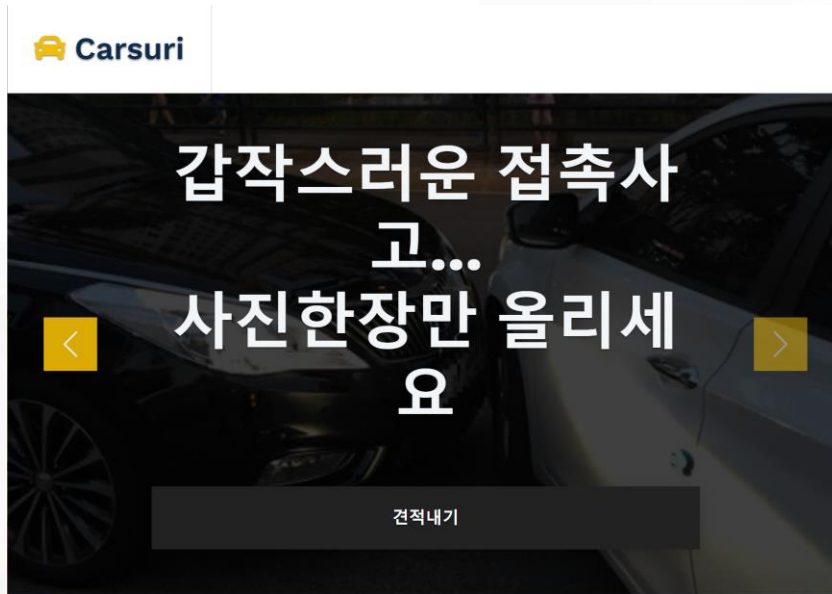
시스템 구성도 내용



디바이스로 인터넷에 연결
웹에 연결
데이터 베이스 바탕으로 모델 연동
웹에 출력된 내용을 다시 디바이스로 전송

서비스 구현

메인 페이지 구현

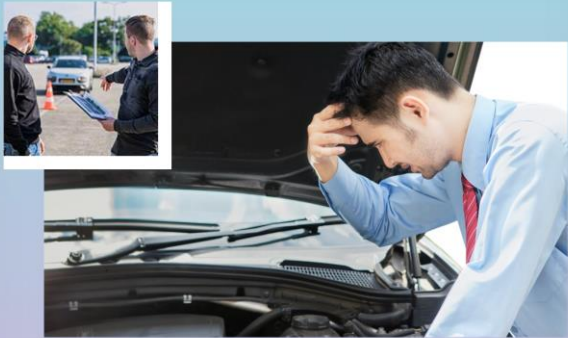


Carsuri

갑작스러운 접촉사
고...
사진한장만 올리세
요

견적내기

The banner features a dark background with a car's front end. The text is in white and yellow. There are yellow arrow icons on the left and right sides of the main text. A dark button with white text is at the bottom center.



소개

차량이 긁히거나 파손이 있을 경우 사진 한 장으로 간단하게 예상 견적서를 확인해보세요!

- ✓ 보험에 가입을 안하셨나요?
- ✓ 본인의 실수로 사고를 내셨는데 보험 처리 시 보험료 인상금액이 걱정되시나요?
- ✓ 차에 흠집이 생겼는데 수리비가 얼마인지 궁금하신가요?
- ✓ 자연재해로 인해 본인의 돈으로 임시적으로 수리하셔야 하나요?

The block contains an image of a man in a blue shirt looking at a car's engine. Above it is a smaller inset image of two men talking. Below the image is a purple gradient box with white text and a list of four questions with checkmarks.

서비스 구현

서비스 구현

The screenshot displays a web application interface for a car rental service, divided into several sections:

- Top Section:** A teal button labeled "카를 클릭하기" (Click Car).
- Brand Selection:** A row of six yellow buttons for car brands: "아반떼(2019~2021)", "아반떼(2019~2021)", "아반떼(2019)", "아반떼(2019)", "아반떼(2019)", and "올뉴아반떼".
- Manufacturer Selection:** A teal header "제조사를 선택해주세요!" (Please select a manufacturer!) with five teal buttons: "기아", "현대", "쌍용", "현대", and "현대".
- Model Selection:** A teal header "차종을 선택해주세요!" (Please select a model!) with a grid of teal buttons for various car models such as "i3000", "670", "680", "그랜저", "i30", "i40", "쏘나타", "스태렉스", "엑센트", "엑스트로스", "레뉴", "베리크루즈", "벨로스터", "스태리아", "윈다비", "아반떼", "엑셀랑", "아이오닉", "에쿠스", "동원", "코나", and "엘리세이드".
- Image Upload:** A dark-themed file explorer window is open, showing a list of image files. The files include "Wheel_c_val_11.jpg", "Side_mirror_c_val_34.jpg", "Rear_bumper_c_val_503.jpg", "Rear_bumper_c_val_3.jpg", "Front_bumper_c_val_54.jpg", and "Front_bumper_c_val_933.jpg". The "Front_bumper_c_val_933.jpg" file is selected.
- Final Selection:** Three yellow buttons for car models: "아반떼(MD100)", "아반떼(MD113)", and "올뉴아반떼".
- Bottom Section:** A teal button labeled "사진 첨부하기" (Attach photo).

서비스 구현

오류 메시지 구현

localhost 내용:

이미지는 최대 6개 까지 업로드가 가능합니다.

확인

localhost 내용:

이미지 파일만 업로드가 가능합니다.

확인

서비스 구현

최종 확인 및 추천 구현

사진 첨부하기

위의 차량이 선택한 차량이 맞습니까?

예
 아니요

요청하신 차량 부위에 대한 견적 내용입니다.

- 1 견적으로 인한 가격이 실제 견적과 다를 수 있습니다.
- 2 본인이 선택한 차종이랑 다른 차종의 사진일 경우 가격이 다를 수 있습니다.
- 3 차량이 아닌 다른 사진을 올리셨을 경우 견적이 나올 수도 있습니다.



Part: 앞 범퍼
Repair: 교환
cost: 396,000



Part: 앞 범퍼
Repair: 수리
cost: 411,000



Part: 뒷 범퍼
Repair: 교환
cost: 371,000



Part: 뒷 범퍼
Repair: 수리
cost: 391,000



Part: 사이드 미러
Repair: 교환
cost: 77,000



Part: 휠
Repair: 교환
cost: 403,000

주변 카센터 추천



(주)천일오토모빌 02-2175-8180

에이큰 아카데미 강남 3동, 수리남

04 프로젝트 시연



실제 서비스 시연

<http://mycarsuri.shop>

프로젝트 진행 시 발생 문제, 문제에 대한 접근방식 결정

이미지 분류의 한계

1. 객체 검출 모델의 구조에 대한 이해 부족
2. 수집한 데이터의 레이블(coco 데이터셋 형식) 구조 파악에 시간을 과도하게 소모
3. 60만장 규모의 이미지 딥러닝을 위한 장비를 갖추고 있지 않음 (추후 1050ti, 32G RAM 대여로 다소 개선)
4. 딥러닝 하이퍼파라미터에 대한 이해도 부족
5. 모델의 레이어 구성에 대한 이해도 부족

따라서 현실적인 조건들을 고려하여 이미지 분류로 프로젝트 진행 결정

이미지 분류 방식 채택

1. 차량 부위별 파손 종류를 각각 별개의 클래스로 학습시키는 분류방식
2. 차량 부위를 판별한 후 파손 종류를 판별하는 이진분류 방식

난이도와 일정의 문제로 전자를 선택

05
질의응답



감사합니다

