



중고차 플랫폼을 위한 머신 러닝

머신 러닝을 통한 중고차 가격 예측, 차종 분류, 차량 파손 검출

SOCAR x LIKE LIONS x **SS501**

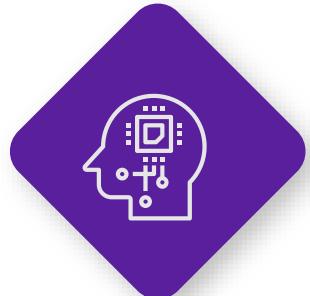


중고차 플랫폼을 위한 머신 러닝

NO	목차
01	소개
02	차종 분류
03	번호판 인식
04	중고차 가격 예측
05	Scratch Semantic Segmentation
06	찌그러짐 인식
07	웹/앱 개발

— 중고차 플랫폼을 위한 머신 러닝

IDEA



최종 목표

차량 파츠(부품)에 따른 차량 파손을 정확히 검출 또는 분할하여
감가상각이 적용된 중고차 가격을 정확히 예측.



현 시점에서 구현 불가



중고차 플랫폼을 구축한다고 가정



중고차 플랫폼을 위한 머신 러닝 기능 구현



중고차 가격 예측, 차종 분류, 번호판 인식,
파손 검출

우리팀은요

SS501



Socar & Student 5 0 likelions

쏘카 해커톤 학생 다섯명이 한마리의
멋쟁이 사자처럼!

술은 고수



김가영

번호판 인식, 웹 개발

철없는 팀장



박웅규

차종 분류, 파손 인식

개강中인데도…



김수정

중고차 가격 예측, 웹 개발

열정 폭발



백인욱

차종 분류, 웹/앱 개발

어디계신가요?



박준언

파손 인식 자료 조사

중고차

시장을 알아보자

01

중고차 시장 규모 ?

₩ 39조

02

2025년에는 ?

₩ 50조

03

당사자간 거래?

55%

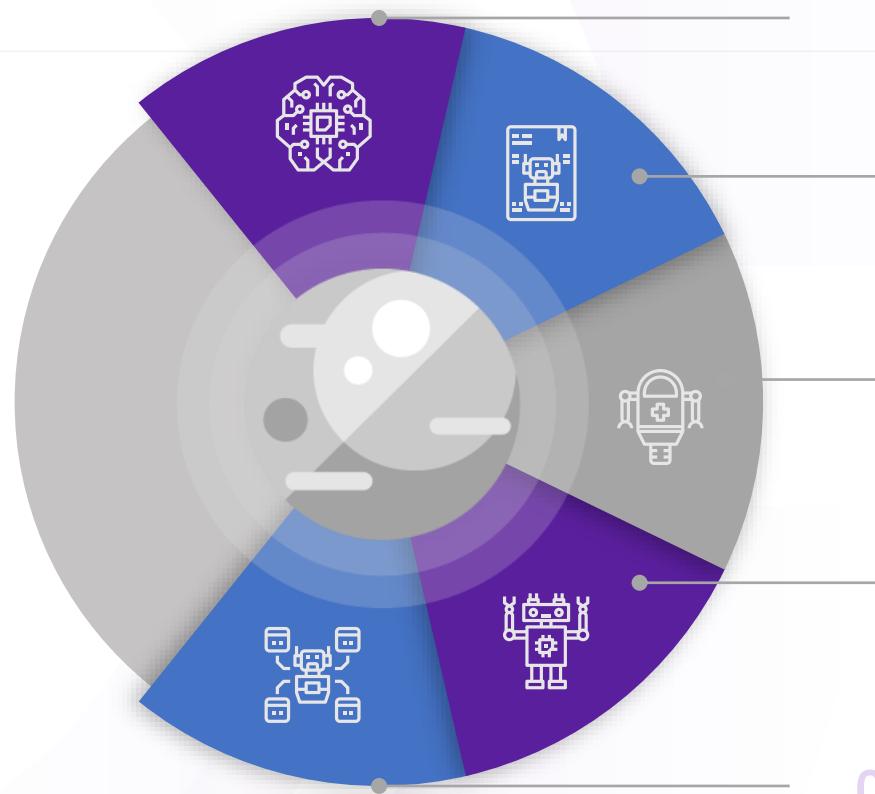
04

반도체 수급난 ?

호황

중고차 플랫폼을 위한 머신 러닝

SOCAR x LIKELIONS x SS501



01 차종 분류

02 번호판 인식

03 중고차 가격 예측

04 Scratch Segmentation

05 Dent Detection

차종 분류



내 차의 세부 차종은?



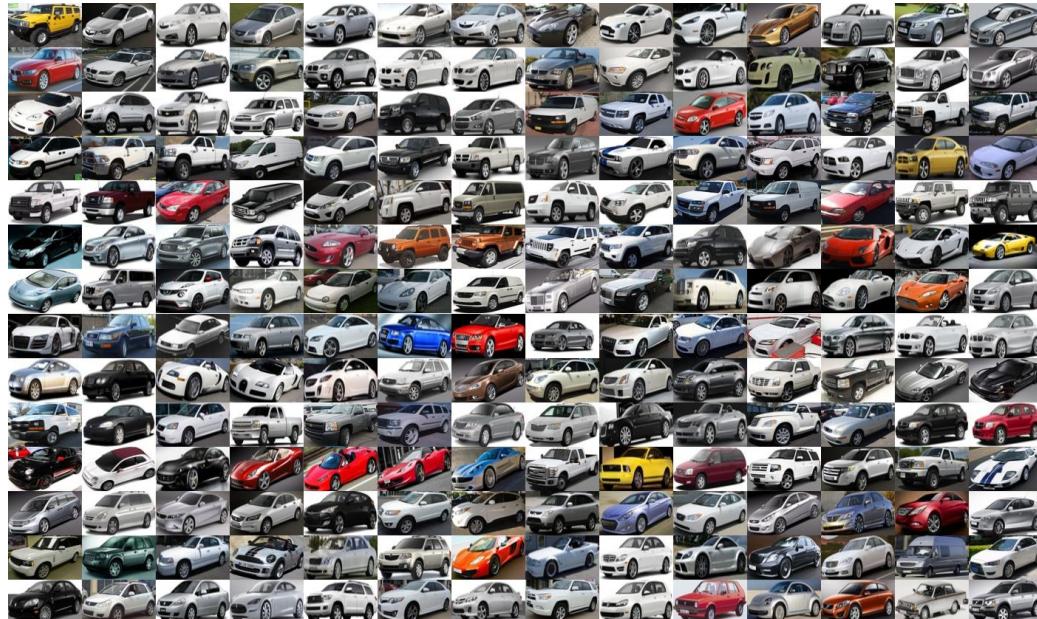
이 차는 무슨 차야?



이 차는 가격이 얼마지?
중고는 있나?



차종 분류



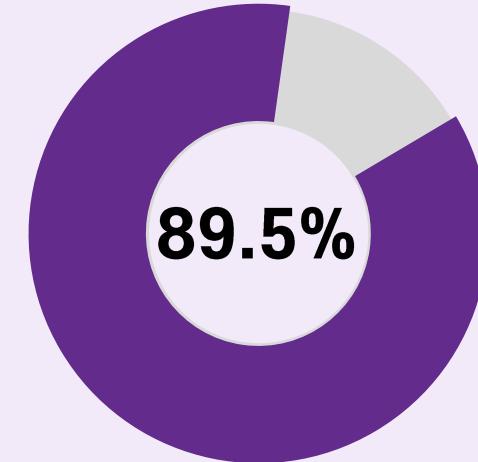
Dataset

Stanford Car Dataset

- Class : 196
- Training : 8144
- Testing : 8041

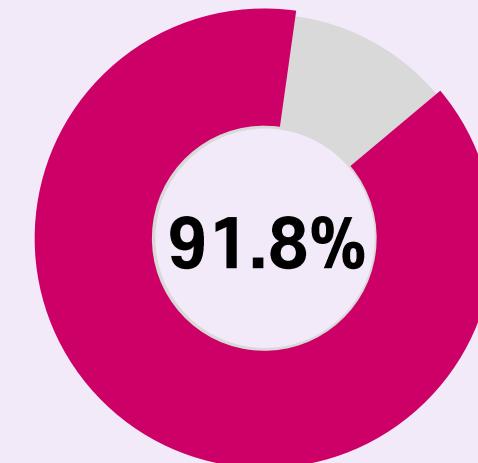
학습 공통 사항

- Epoch : 40
- Scheduler : ReduceLROnPlateau



89.5%

Resnet-152
Adam
Cross Entropy Loss



91.8%

Efficientnet-b3 with Mish
RAdam
Label Smoothing Loss()

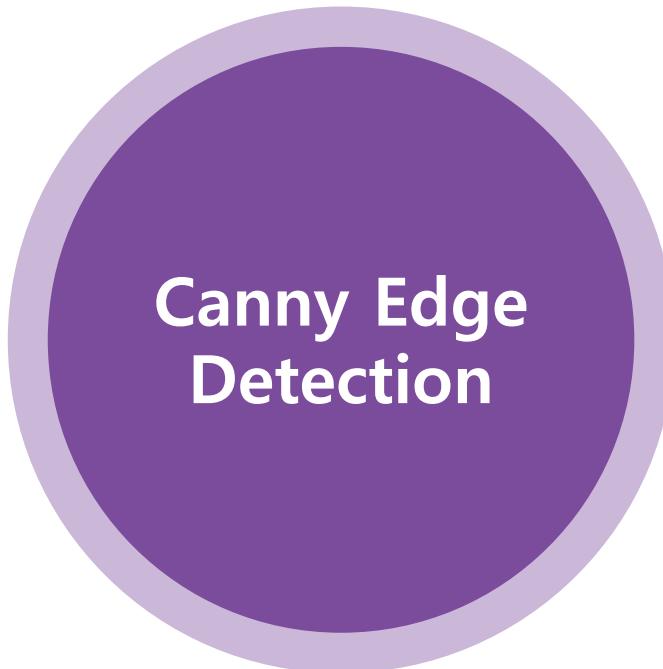
중고차 플랫폼을 위한 머신 러닝

번호판 인식

- 번호판도 개인 정보
- 차량 번호를 통해 차량 정보 조회 가능



번호판 위치 및 글자 인식



적은 에러율!

Detection을 통해 모든 Edge를 인식하며,
인식된 Edge는 모두 실제 Edge이다.

좋은 Localization!

Detection을 통하여 검출된 Edge와
실제 Edge의 중심 간 거리가 최소가 된다.

하나의 Edge!

실제 Edge 하나에 대해서 한 Edge만 검출해야 한다.

번호판 위치 및 글자 인식



Tesseract / Pytesseract

Basic Setting : [앞 숫자][문자][뒤 숫자] ex) 01가2345



Test Result with 1000 data

[O][X][O] [앞 숫자와 뒤 숫자]를 맞게 검출한 경우가 [433]개
43.3%

[O][X][X] [앞 숫자]만 맞게 검출한 경우가 [305]개 30.5%

[O][O][O] 모든 부분을 맞게 검출한 경우가 [174]개 17.4%

[O][O][X] [앞 숫자와 문자]를 맞게 검출한 경우가 [45]개 4.5%

[X][X][O] [뒤 숫자]만 맞게 검출한 경우가 [36]개 3.6%

[X][X][X] 세 부분 모두 틀리게 검출한 개수가 [7]개 0.7%

[X][O][X] [문자]만 맞게 검출한 경우가 [0]개 0%

[X][O][O] [문자와 뒤 숫자]를 맞게 검출한 경우가 [0]개 0%

번호판 위치 및 글자 인식



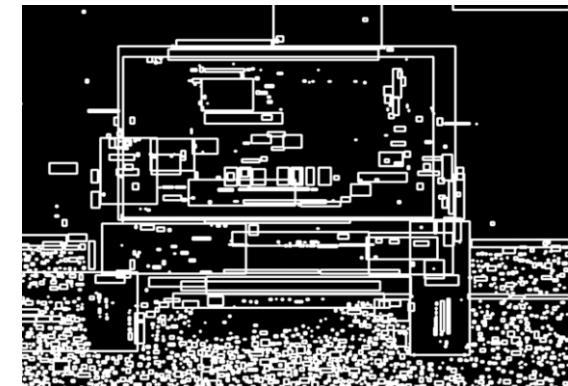
이미지 입력



Grayscale로 변환

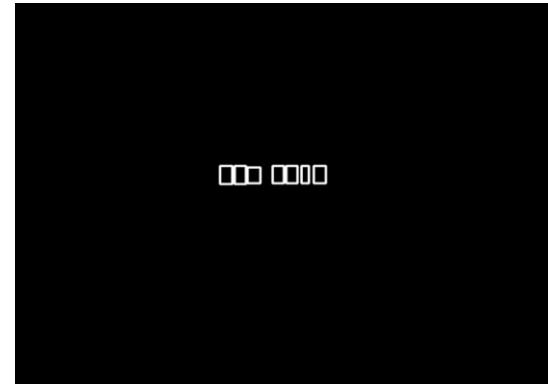


Canny Edge



Labeling

번호판 위치 및 글자 인식

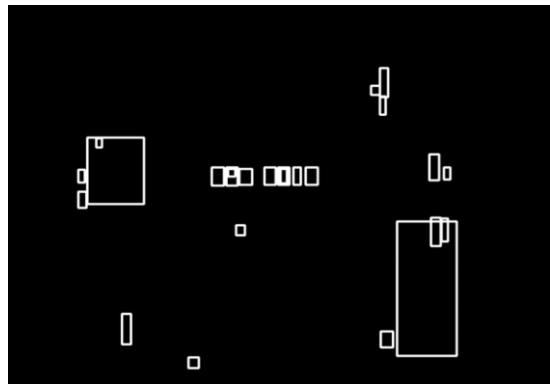


후보 줄이기, OCR 실행

OCR 결과 - 有 모음

번호판 인식

번호판 정보 제거

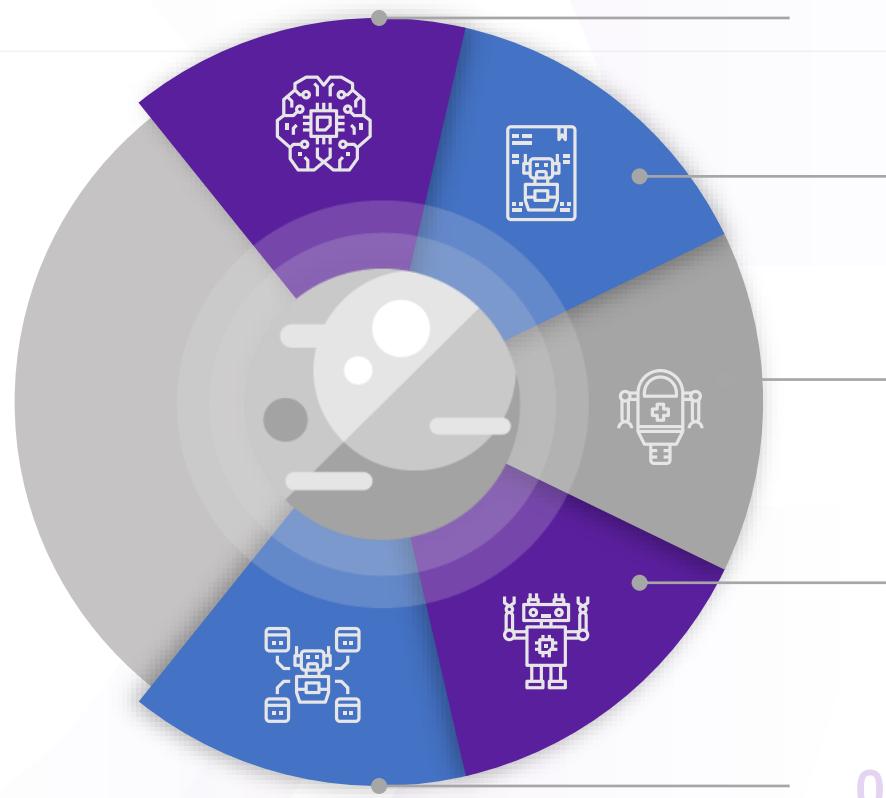
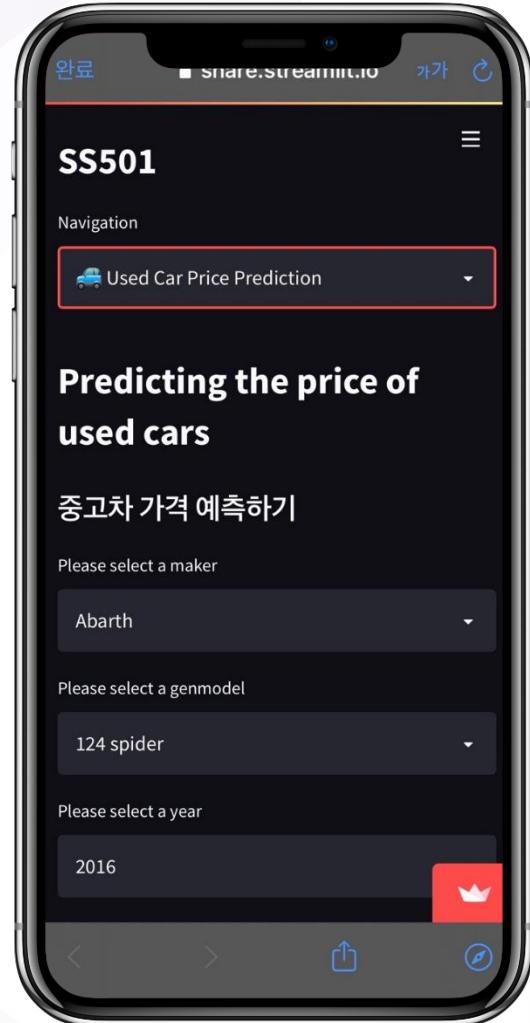


·49구 2013·



중고차 플랫폼을 위한 머신 러닝

SOCAR x LIKELIONS x SS501



01 차종 분류

02 번호판 인식

03 중고차 가격 예측

04 Scratch Segmentation

05 Dent Detection

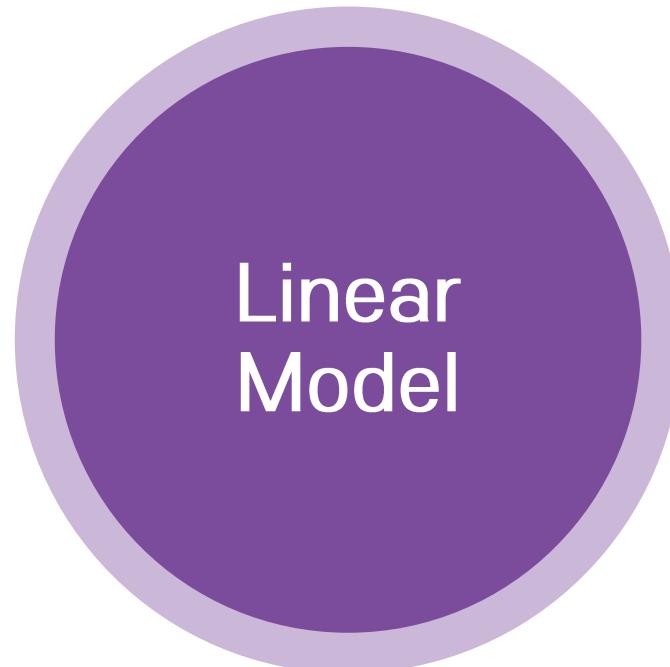
중고차 플랫폼을 위한 머신러닝

SOCAR x LIKELIONS x SS501



중고차 가격 예측

중고차 가격 예측



Linear
Model

간단하지만 효과적인!

관련있는 데이터들을 가장 잘 설명할 수 있는
하나의 Line을 그어 target값을 예측

직관적인!

예측값을 하나의 Line으로 그리기 때문에 직관적임

중고차 가격 예측

DVM-CAR Dataset



- **Basic table:** car attributes such as model name, model ID and brand name.
- **Sales table:** ten years car sales data in UK/GB.
- **Price table:** entry-level (i.e. the cheapest trim price) new car prices across years.
- **Trim table:** trim attributes like the selling price (trim level), engine type and engine size.
- **Ad table:** more than 0.25 million used car advertisements.
- **Image table:** car images attributes like colour and viewpoint.

사용 Feature

DVM Car Dataset을 활용
Genmodel_ID : Genmodel의 ID
Maker : 차의 브랜드, 메이커
Genmodel : 차의 모델명
Trim : 옵션을 포함한 차의 모델명
Year : 판매된 연도
Price : 그 연도에 판매된 가격
Gas_emission : 가스 배출
Fuel_type : 연료의 타입
Engine_size : 엔진 사이즈
Entry_price : 제일 싼 판매 가격



특징 추출 및 가공

중고차 가격 예측

02. Train & Test the Linear Model



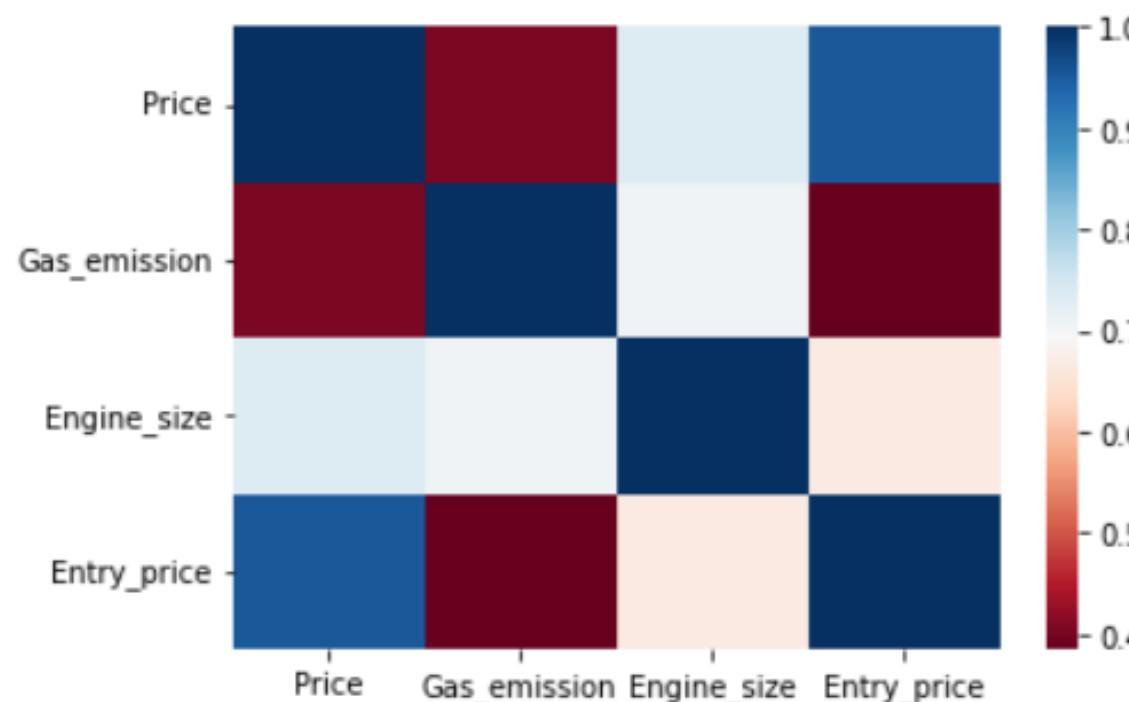
- Train : 70% 거래 데이터**
LinearRegression() : 선형회귀 사용

- Test Result : 30% 거래 데이터**
MSE : 25812749.262, RMSE : 5080.6249
R_Square : 0.907

중고차 가격 예측

03. Simulation

	Price	Gas_emission	Engine_size	Entry_price
Price	1.000000	0.402581	0.735678	0.952523
Gas_emission	0.402581	1.000000	0.706309	0.385485
Engine_size	0.735678	0.706309	1.000000	0.664360
Entry_price	0.952523	0.385485	0.664360	1.000000



중고차 가격 예측

Predicting the price of used cars

중고차 가격 예측하기

Please select a maker!

Abarth

Please select in a genmodel!

124 spider

Please select the year!

2016

predict : 37256

STEP 1
Select maker

STEP 2
Select genmodel

STEP 3
Select year

Finish
Check result

중고차 가격 예측

결과 예

Please select a maker

Dacia

Please select a genmodel

Duster

Please select a year

2012

Prediction in Pound: 13570 £

Price in Pound: 13320 £

Prediction in Won: 2209 만원

Price in Won: 2168 만원

Difference in Won: 40 만원

Please select a maker

Bentley

Please select a genmodel

Arnage

Please select a year

1998

Prediction in Pound: 170241 £

Price in Pound: 145000 £

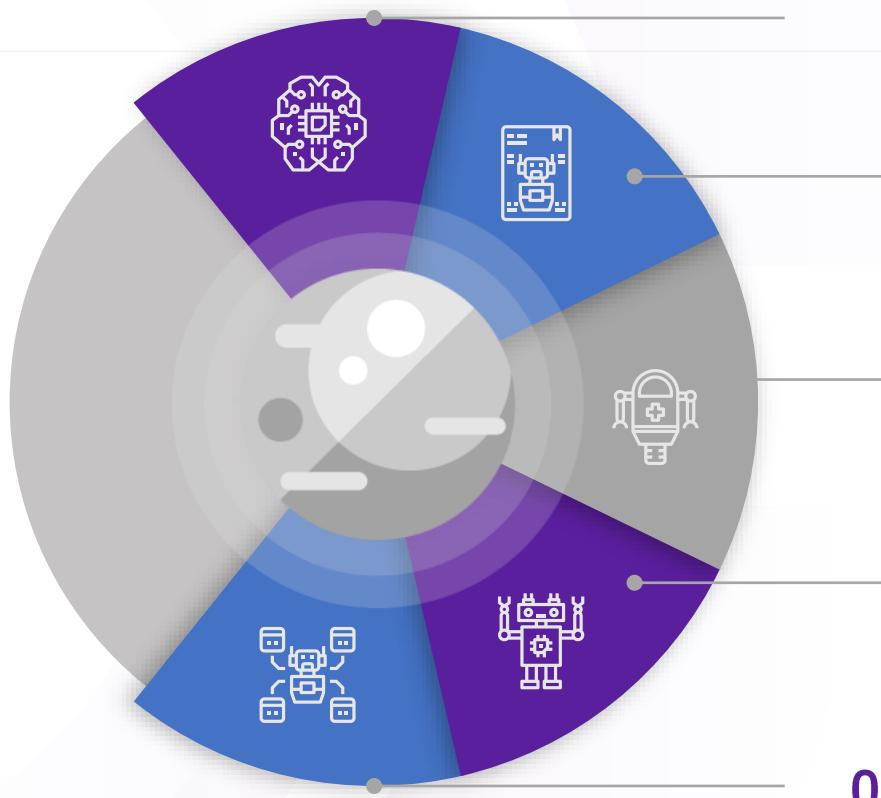
Prediction in Won: 27718 만원

Price in Won: 23609 만원

Difference in Won: 4109 만원

중고차 플랫폼을 위한 머신 러닝

SOCAR x LIKELIONS x SS501



01 차종 분류

02 번호판 인식

03 중고차 가격 예측

04 Scratch Segmentation

05 Dent Detection

중고차 플랫폼을 위한
머신 러닝

Scratch Semantic Segmentation



Scratch Semantic Segmentation

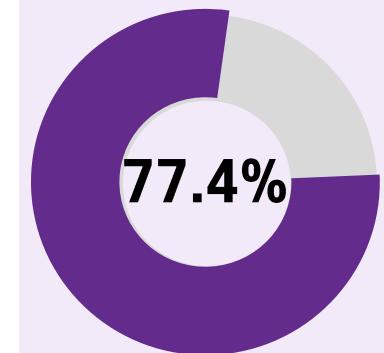
Dataset

쏘카 제공 데이터셋(image/mask)

- Training : 2132
- validation : 267
- Testing : 267

학습 공통 사항

- 사용 모델 : Efficient-B7
- 손실함수 : Focal Loss + High Pass Filter Constraint Loss
- 최적화 방법 : Adam
- Epoch : 15
- Scheduler : StepLR
- 후 처리 적용
 - CRF – 성능 떨어짐
 - Threshold – 작은 성능 향상

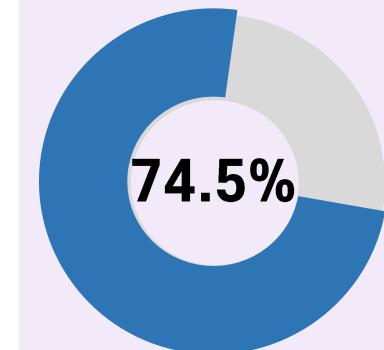


Non Post Processing

miou: 0.774468

iou_back : 0.981806

iou_scratch : 0.567130

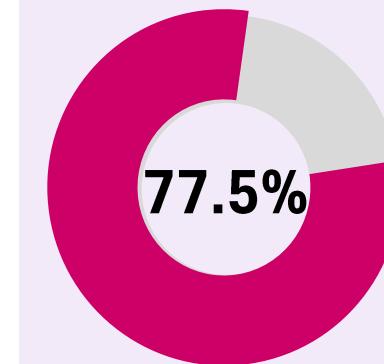


CRF 적용

miou: 0.745233

iou_back : 0.980299

iou_scratch : 0.510166



Thresholding

miou: 0.774886

iou_back : 0.981813

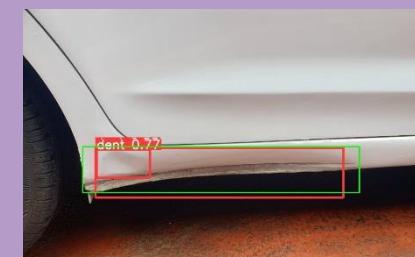
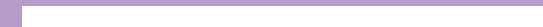
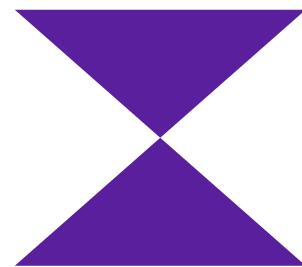
iou_scratch : 0.567958

중고차 플랫폼을 위한 머신 러닝

SOCAR x LIKELIONS x SS501

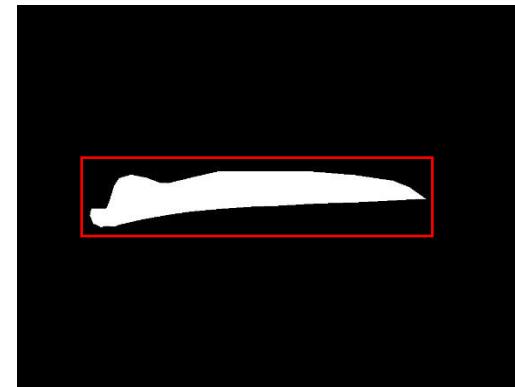


피그리짐 인식



찌그레짐 인식

SOCAR x LIKELIONS x SS501

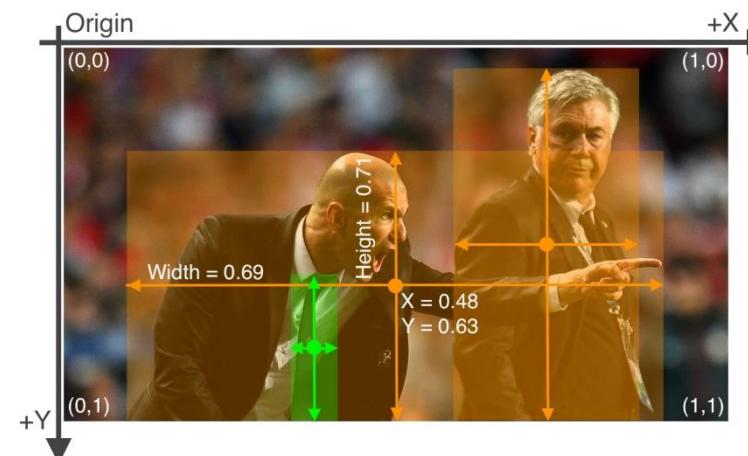
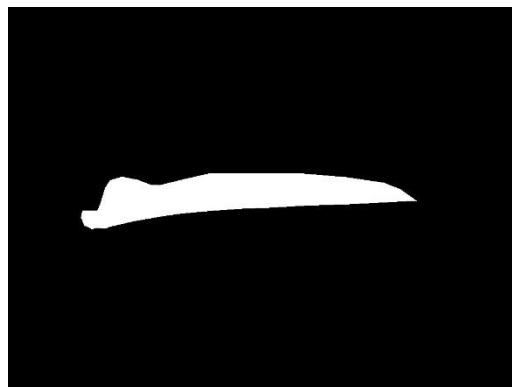


이미지 입력

Labeling

Label box 정보 저장

Yolov5 학습



[class x_center y_center width height]



찌그러짐 인식

Dataset

쏘카 제공 데이터셋(image/mask)

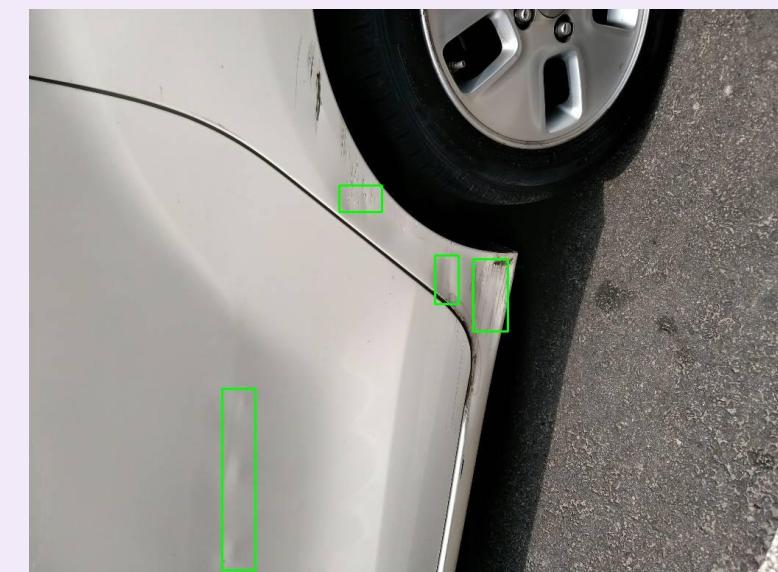
- Training : 2132
- validation : 267
- Testing : 267

학습 공통 사항

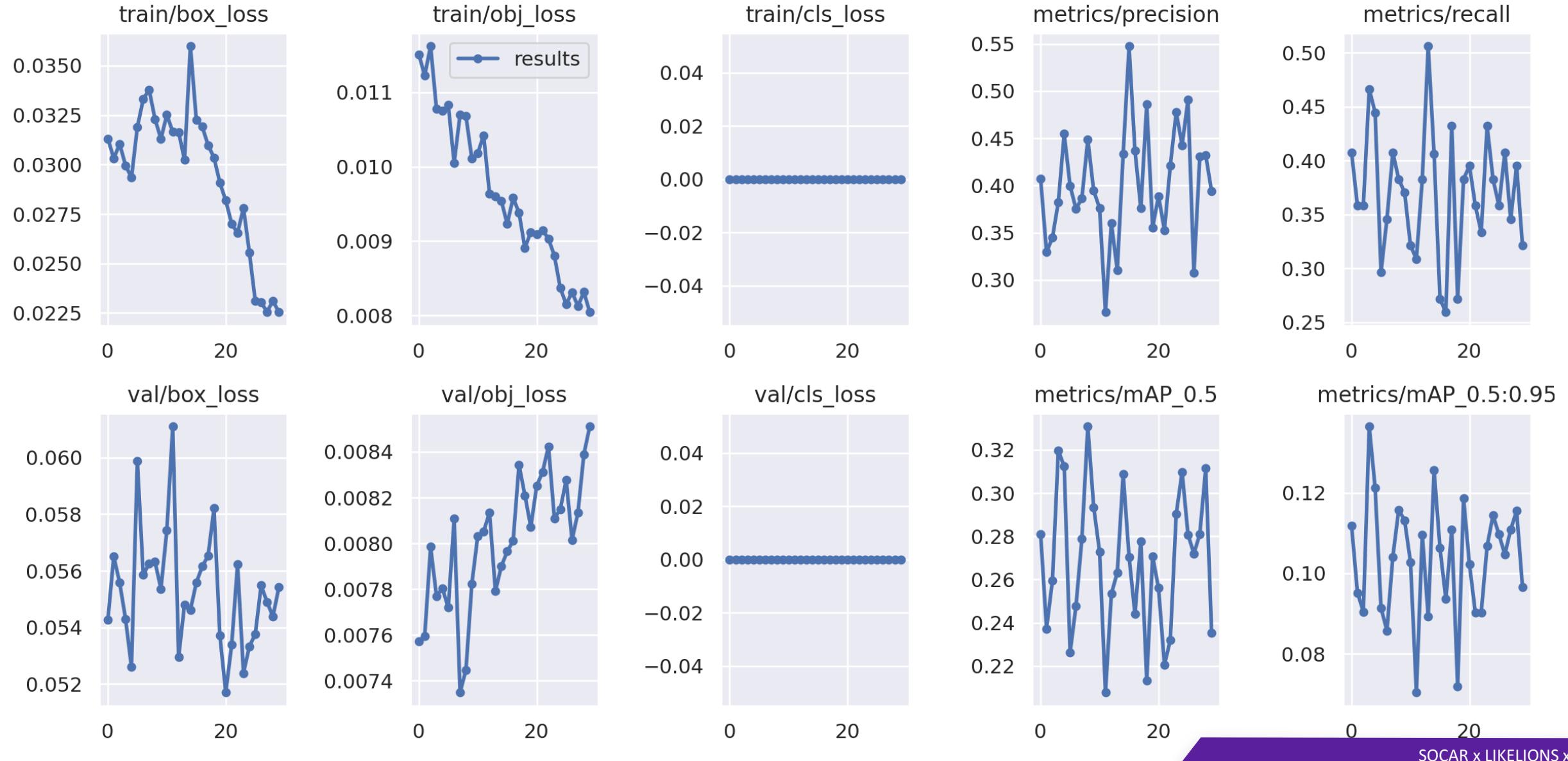
- 사용 모델 : Yolov5-Ultralytics
- 손실함수 : BCEWithLogitsLoss
 - + bounding box regression loss
- 최적화 방법 : SGD
- Epoch : 30 + 30
- Scheduler : OneCycleLR

실험 결과 정리

- True Positive : 36개
- False Positive 17개
- True Negative : 51개



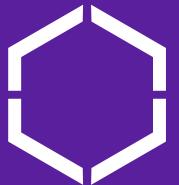
피그리짐 인식



중고차 플랫폼을 위한 머신 러닝

웹/앱 개발

- 웹으로 머신 러닝 모델 배포
- 안드로이드 어플 개발



웹/앱 개발

모델 구현



차종 분류



파손 인식



중고차 가격 예측



번호판 인식

웹 개발



Flask Over Streamlit

1. Data Science 프로젝트에 최적화 되어있다.
2. Flask 보다 더 빠르고 간단하게 웹/앱을 구현 할 수 있다.
3. 배포시 Heroku 대신 Streamlit Cloud 를 사용할 수 있다.

앱 빌드 및 배포



Figma



Bravo

1. Figma로 앱 구현

2. Bravo Studio로 빌드 후 Google Play Store 배포

웹/앱 개발

■ Web View [LINK]



■ App Start Screen [안드로이드 배포]



진행 상황 및 목표



Progress 01

- 주제 선정
- 업무 분담

Progress 02

- 머신 러닝 진행
- 진행 상황 공유

Progress 03

- 성능 향상 중
- 웹/앱 개발 중

Progress 04

- 서비스 시작
- 데이터 축적
- 성능 향상

Progress 05

- 중고차 정확도 향상
- 서비스 질 향상

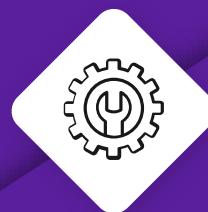
감사합니다

SOCAR x LIKELIONS x SS501



홈페이지

Slack 공지사항 / 갤러리 / 링크 참조



안드로이드

Slack 공지사항 / 갤러리 / 링크 참조



Github

해커톤 종료 후 공개.